

Daniel Trabold Pascal Welke Nico Piatkowski
(Eds.)

**Proceedings of the LWDA 2020 Workshops:
KDML, FGWM, FGWI-BIA, and FGDB**

Online at <http://LWDA2020.org>
September 9 – 11, 2020

Published at <http://ceur-ws.org>

Preface

LWDA 2020 is a joint conference of four special interest groups of the German Computer Science Society (GI), addressing research in the areas of knowledge discovery and machine learning, information retrieval, database systems, and knowledge management. The German acronym LWDA stands for “Lernen, Wissen, Daten, Analysen” (Learning, Knowledge, Data, Analytics). Following the tradition of the last years, LWDA 2020 provides a joint forum for experienced and young researchers, to bring insights to recent trends, technologies and applications and to promote interaction among the special interest groups. The following special interest groups participate at LWDA 2020:

- KDML (Knowledge Discovery, Data Mining and Machine Learning)
- FGWM (Knowledge Management)
- FGWI-BIA (Business Intelligence and Analytics)
- FGDB (Database Systems)

LWDA 2020 was planned to be held at the University of Bonn. Due to Covid-19 the organizers decided to host the conference virtually in the interest of public health. The papers of the individual workshops have been peer reviewed and selected by independent program committees from the respective domains. The program consists of several joint sessions which include contributions of interest for all conference participants. In addition, there are parallel sessions for individual workshops focussing on more specific topics. A poster session and short pitch videos gives all presenters the opportunity to discuss their work in a broader context. Recent trends in the corresponding research areas are highlighted by three distinguished keynote speakers:

- Geoff Webb, Monash University
- Kristian Kersting, Technical University of Darmstadt
- Thomas Gärtner, Technical University of Vienna

The organizers would like to thank the workshop chairs and their program committees for their excellent work as well as the keynote speakers for their contribution to the success of LWDA 2020. We hope that LWDA 2020 will be an inspiring event for all participants with lots of scientific exchange and discussions.

September 2020

Daniel Trabold
Pascal Welke
Nico Piatkowski
(Editors, LWDA'20)

Organization

LWDA 2020 is hosted and organized jointly within the competence center ML2R by Fraunhofer IAIS and the research group “Machine Learning and Artificial Intelligence” at the University of Bonn.

LWDA 2020 General Chair

Daniel Trabold	Fraunhofer IAIS
Stefan Wrobel	Fraunhofer IAIS & University of Bonn

KDML 2020 Workshop Chair

Pascal Welke	University of Bonn
Nico Piatkowski	Fraunhofer IAIS

FGWM 2020 Workshop Chair

Lisa Grumbach	University of Trier
Pascal Reuss	University of Hildesheim

FGWI-BIA 2020 Workshop Chair

Henning Baars	University of Stuttgart
---------------	-------------------------

FGDB 2020 Workshop Chair

Wolf-Tilo Balke	TU Braunschweig
Stefan Conrad	Heinrich Heine University Düsseldorf

Program Committee

KDML 2020 Programme Committee

Maram Akila	Fraunhofer IAIS
Klaus-Dieter Althoff	DFKI / University of Hildesheim
Martin Atzmueller	Tilburg University
Christian Bauckhage	Fraunhofer IAIS
Rainer Gemulla	University of Mannheim
Goran Glavaš	University of Mannheim
Stephan Günnemann	Technical University of Munich
Marwan Hassani	Eindhoven University of Technology
Sibylle Hess	Data Mining Group, TU Eindhoven
Andreas Hotho	University of Wuerzburg
Sebastian Houben	Fraunhofer IAIS
Marius Kloft	Technical University of Kaiserslautern
Christian Kühnert	Fraunhofer IOSB
Florian Lemmerich	RWTH Aachen University
Thomas Liebig	Materna SE
Michael Mock	Fraunhofer IAIS
Petar Ristoski	IBM Research-Almaden
Ute Schmid	University of Bamberg
Thomas Seidl	Ludwig-Maximilians-University (LMU) Munich
Stefan Wrobel	Fraunhofer IAIS & University of Bonn

FGWM 2020 Programme Committee

Klaus-Dieter Althoff	DFKI / University of Hildesheim
Kerstin Bach	Norwegian University of Science and Technology
Joachim Baumeister	denkbare GmbH and University of Würzburg
Ralph Bergmann	University of Trier
Andrea Kohlhase	University of Applied Sciences Neu-Ulm
Michael Kohlhase	Friedrich-Alexander-University Erlangen-Nürnberg
Michael Ley	University of Rostock
Mirjam Minor	Goethe University Frankfurt
Ulrich Reimer	University of Applied Sciences St. Gallen
Bodo Rieger	University of Osnabrück
Christian Severin Sauer	University of Hildesheim

FGWI-BIA 2020 Programme Committee

Carsten Felden	Technical University Bergakademie Freiberg
Ralf Finger	Information Works
Sebastian Olbrich	EBS University

FGDB 2020 Programme Committee

Klemens Böhm	Karlsruhe Institute of Technology
Ralf Krestel	University of Passau
Stephan Mennicke	Technical University of Dresden
Sebastian Michel	Technical University of Kaiserslautern
Felix Naumann	Hasso Plattner Institute, University of Potsdam
Gunter Saake	University of Magdeburg
Kai-Uwe Sattler	Technical University of Ilmenau

Table of Contents

Keynote Talks

Time Series Classification at Scale <i>Geoffrey I. Webb</i>	2
On Hybrid and Systems AI <i>Kristian Kersting</i>	3
Interactive Machine Learning with Structured Data <i>Thomas Gärtner</i>	4

KDML Workshop

Solving Abstract Reasoning Tasks with Grammatical Evolution <i>Raphael Fischer, Matthias Jakobs, Sascha Mücke and Katharina Morik</i>	6
Grace - Limiting the Number of Grid Cells for Clustering High-Dimensional Data <i>Anna Beer, Daniyal Kazempour, Julian Busch, Alexander Tekles and Thomas Seidl</i>	11
Segmenting and Clustering Noisy Arguments <i>Lorik Dumani, Christin Katharina Kreutz, Manuel Biertz, Alex Witry and Ralf Schenkel</i>	23
Combining Universal Adversarial Perturbations <i>Beat Tödli and Maurus Kühne</i>	35
Phantom Embeddings: Using Embeddings Space for Model Regularization in Deep Neural Networks <i>Mofassir Ul Islam Arif, Mohsan Jameel, Josif Grabocka and Lars Schmidt-Thieme</i>	47
A hierarchical multi-level product classification workbench for retail <i>Maximilian Harth, Christian Schorr and Rolf Krieger</i>	59
Comparison of knowledge based feature vector extraction and geometrical parameters of Photovoltaic I-V Curves <i>Cem Basoglu, Grit Behrens, Konrad Mertes and Matthias Diehl</i>	70
Using Probabilistic Soft Logic to Improve Information Extraction in the Legal Domain <i>Birgit Kirsch, Sven Giesselbach, Timothée Schmude, Malte Völkening, Frauke Rostalski and Stefan Rüping</i>	76

Fusing Multi-label Classification and Semantic Tagging <i>Jörg Kindermann and Katharina Beckh</i>	88
Native sentiment analysis tools vs. translation services - Comparing GerVADER and VADER <i>Karsten Tymann, Louis Steinkamp, Oxana Zhurakovskaya and Carsten Gips</i>	100
EmoDex - An emotion detection tool composed of established techniques <i>Oxana Zhurakovskaya, Louis Steinkamp, Karsten Tymann and Carsten Gips</i>	105

FGWM Workshop

Construction of a Corpus for the Evaluation of Textual Case-based Reasoning Architectures <i>Andreas Korger and Joachim Baumeister</i>	118
Visualizing the behaviour of CBR agents in a FPS scenario <i>Jobst-Julius Bartels, Sebastian Viefhaus, Philipp Yasrebi-Soppa, Pascal Reuss and Klaus-Dieter Althoff</i>	130
Development and Implementation of a Case-Based Reasoning Approach to Speed-Up Deep Reinforcement Learning through Case-Injection for AI Gameplay <i>Marcel Heinz, Jakob Michael Schoenborn and Klaus-Dieter Althoff</i>	142
Towards case-based reasoning in real-time strategy environments with SEASALT <i>Jakob Michael Schoenborn and Klaus-Dieter Althoff</i>	154
Process Mining for Case Acquisition in Oncology: A Systematic Literature Review <i>Joscha Grüger, Ralph Bergmann, Yavuz Kazik and Martin Kuhn</i>	162
Student Graduation Projects in the Context of Framework for AI-Based Support of Early Conceptual Phases in Architecture <i>Viktor Eisenstadt, Klaus-Dieter Althoff and Christoph Langenhan</i>	174
A Concept for the Automated Reconfiguration of Quadcopters <i>Kaja Balzereit, Marta Fullen and Oliver Niggemann</i>	180
INWEND: Using CBR to automate legal assessment in the context of the EU General Data Protection Regulation <i>Clarissa Dietrich, Sebastian Schriml, Ralph Bergmann and Benjamin Raue</i>	192

FGWI-BIA Workshop

- Ein agiles Vorgehensmodell zur Einführung von Predictive Analytics in Unternehmen 203
Jule Aßmann, Joachim Sauer and Michael Schulz
- Comparing Brand Perception Through Exploratory Sentiment Analysis in Social Media 218
Mario Cichonczyk and Carsten Gips
- Substitution der Akteur-Beteiligung durch KI und BI am Beispiel eines Logistik-Projekts in den Neuss-Düsseldorfer Häfen 234
Claus Brell, Ralf Kuron and Wilhelm Müller

FGDB Workshop

- Discovery of Ontologies from Implicit User Knowledge 241
David Haller and Richard Lenz
- Sense Tree: Discovery of New Word Senses with Graph-based Scoring 246
Jan Ehmüller, Lasse Kohlmeyer, Holly McKee, Daniel Paeschke, Tim Repke, Ralf Krestel and Felix Naumann
- Schema Evolution and Reproducibility of Long-term Hydrographic Data Sets at the IOW 258
Tanja Auge, Erik Manthey, Susanne Jürgensmann, Susanne Feistel and Andreas Heuer
- Future Fetch – Towards a ticket-based data access from secondary storage in database systems 270
Demian E. Vöhringer and Klaus Meyer-Wegener
- Modeling Interdependent Preferences over Incomplete Knowledge Graph Query Answers 279
Till Affeldt, Stephan Mennicke and Wolf-Tilo Balke
- Towards Evolutionary, Domain-Specific Query Classification Based on Policy Rules 291
Peter K. Schwab and Klaus Meyer-Wegener

Keynote Talks

Time Series Classification at Scale

Geoffrey I. Webb

Faculty of Information Technology, Monash University, VIC 3800, Australia
Geoff.Webb@monash.edu

Abstract. Time series classification is a fundamental data science problem, providing understanding of dynamic processes as they evolve over time. The recent introduction of ensemble techniques has revolutionised this field, greatly increasing accuracy, but at a cost of increasing already burdensome computational overheads. I present new time series classification technologies that achieve the same accuracy as recent state-of-the-art developments, but with many orders of magnitude greater efficiency and scalability. These make time series classification feasible at hitherto unattainable scale.

Copyright © 2020 by the paper's authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

On Hybrid and Systems AI

Kristian Kersting

Computer Science Department, TU Darmstadt, 64289 Darmstadt, Germany
kersting@cs.tu-darmstadt.de

Abstract. Our minds make inferences that appear to go far beyond standard machine learning. Whereas people can learn richer representations and use them for a wider range of learning tasks, machine learning algorithms have been mainly employed in a stand-alone context, constructing a single function from a table of training examples. In this talk, I shall touch upon a view on AI and machine learning, called Systems AI, that can help capturing these human learning aspects by combining different AI and ML models using high-level programming. Since inference remains intractable, existing approaches leverage deep learning for inference. Instead of “just going down the neural road,” I shall argue to also use probabilistic circuits, a deep but tractable architecture for probability distributions. This hybrid approach can speed up inference as I shall illustrate for unsupervised science understanding, database queries and automating density estimation.

Copyright © 2020 by the paper’s authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

Interactive Machine Learning with Structured Data

Thomas Gärtner

Faculty of Informatics, TU Wien, 1040 Vienna, Austria
thomas.gaertner@tuwien.ac.at

Abstract. In this talk I'll give an overview of our contributions to what I call interactive machine learning. Often, interaction in Computer Science is interpreted as the interaction of humans with the computer but I intend a broader meaning of the interaction of machine learning algorithms with the real world, including but not restricted to humans. Interactions with humans span a broad range where they can be intentional and guided by the human or they can be guided by the computer such that the human is oblivious of the fact that he is being guided. Another example of an interaction with the real world is the use of machine learning algorithms in cyclic discovery processes such as drug design. Important properties of interactive machine learning algorithms include efficiency, effectiveness, responsiveness, and robustness. In the talk I will show how these can be achieved in a variety of interactive contexts.

KDML Workshop

Solving Abstract Reasoning Tasks with Grammatical Evolution

Raphael Fischer, Matthias Jakobs, Sascha Mücke, and Katharina Morik

TU Dortmund, AI Group, Dortmund, Germany
<http://www-ai.cs.tu-dortmund.de>

Abstract. The Abstraction and Reasoning Corpus (ARC) comprising image-based logical reasoning tasks is intended to serve as a benchmark for measuring intelligence. Solving these tasks is very difficult for off-the-shelf ML methods due to their diversity and low amount of training data. We here present our approach, which solves tasks via grammatical evolution on a domain-specific language for image transformations. With this approach, we successfully participated in an online challenge, scoring among the top 4% out of 900 participants.

Keywords: Machine Learning · Reasoning · Grammatical Evolution

1 Introduction

Despite AI research’s fast advancements, the question of how to rigorously define and measure intelligence is still open [5,3]. It is taken up by the recently published *Abstraction and Reasoning Corpus* (ARC) [1] and its corresponding *Kaggle* challenge¹. It features hundreds of image-based logic tasks (some examples given in Figure 1), which are expected to be solved by reasoning AIs without any human aid. Finding solutions requires learning the inherent logic of a task from very few examples, which is easy for humans but proves to be very hard for machines. Learning logic from few examples has already been explored (e.g. for text data [2]), however ARC’s image logic space is much larger.

We here present our approach to solve abstract reasoning tasks based on a *domain-specific language* (DSL), whose expressions are generated via *grammatical evolution* (GE). With our method, we were able to reach the 28th place out of over 900 participating teams in the ARC challenge²).

2 Problem Statement

The ARC challenge requires participants to develop a model that is able to solve tasks D_i from the set $\mathcal{D} = \{D_1, \dots, D_M\}$. Each D_i comprises an image-based

Copyright © 2020 by the paper’s authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

¹ <https://www.kaggle.com/c/abstraction-and-reasoning-challenge/>

² *ls8-arc* team on the official ARC challenge leaderboard.

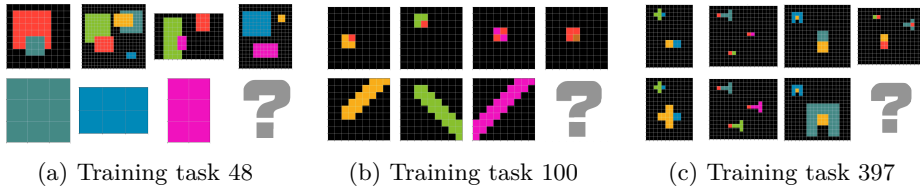


Fig. 1: Exemplary ARC tasks: (a) Crop to the smallest unicolor rectangle; (b) Draw lines over the image in directions indicated by pixels colored in red; (c) Match pattern, upscale and/or recolor if necessary.

reasoning task with m_i training pairs $\{(x^k, y^k)\}$ and n_i test pairs $\{(\hat{x}^\ell, \hat{y}^\ell)\}$, where $m_i > 0$ is typically around 3 and $n_i > 0$ is mostly 1. The images feature up to 10 discrete colors $C = \{c_1, \dots, c_{10}\}$, and image sizes range between 1 and 30 pixels per dimension. The implied function f that maps inputs to the expected output images is vastly different for every task, often based on abstract features (e.g. connected shapes) or pattern continuation (see Figure 1). For each task D_i , the method should derive f_i from $\{(x^k, y^k)\} \in D_i$ which correctly maps all input images to the expected output: $\forall(\hat{x}^\ell, \hat{y}^\ell) \in D_i : f_i(\hat{x}^\ell) = \hat{y}^\ell$.

The challenge participants have access to $M = 400$ training tasks, which show some logic concepts and come with labels \hat{y}^ℓ . The final scoring however is based on an undisclosed test set, whose task are only seen by the model.

3 Reasoning Approach

For our approach, we assume that f can be broken down into a sequence of basic image transformations. We developed a custom *domain-specific language* (DSL) specifying such sequences, whose space of expressions is explored using an *evolutionary algorithm* (EA). We first explain our language in more detail and then show how we use an EA to create ARC task solvers from our DSL.

Domain-Specific Language In a first step, we manually implemented solvers for approximately 20 random training tasks and identified reoccurring image operations, which became the basis of our DSL.

Let $\mathcal{X} = \bigcup_{u,v \geq 1} C^{u \times v}$ denote the set of rectangular images with colors in C , and \mathcal{X}^* ordered lists of images, which we call *layers*. Our DSL is a context-free grammar in *Backus Naur Form* (BNF) [6]; the non-terminal symbols represent function sets whose members (i) modify images ($T = \mathcal{X}^{\mathcal{X}}$), (ii) decompose an image into layers ($S = (\mathcal{X}^*)^{\mathcal{X}}$), (iii) combine layers into a single image ($J = \mathcal{X}^{\mathcal{X}^*}$) or (iv) modify a layer object ($L = (\mathcal{X}^*)^{\mathcal{X}^*}$). The terminal productions of these symbols are either concrete functions of the corresponding type or more complex function compositions. Figure 2 depicts a visualization of our DSL function types.

Our atomic functions comprise basic operations such as translation, rotation and cropping, as well as layer-specific operations like extracting a layer, sorting

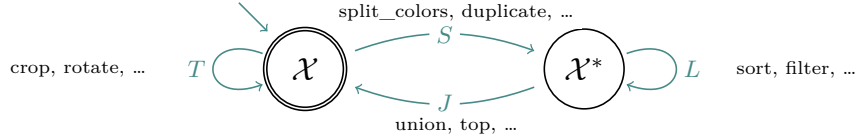


Fig. 2: Function types of custom reasoning DSL, displayed as state automaton, with some example functions for each type.

by different criteria, splitting images into layers, and merging them together. For additional flexibility, we also allow higher-order functions like *map*, which lifts the *T*-type to an *L*-type function by applying it to each layer. The grammar structure ensures that all possible expressions have an overall *T*-type logic, i.e. they transform a single input image into an output image. This allows to find solvers for some tasks, an example is given in Figure 3.

Grammatical Evolution We use Grammatical Evolution (GE) [7] to generate expressions in our DSL, and ultimately find solvers for a given task. We use the standard modulo-based mapping from codons to syntax trees of our DSL [6] to obtain image functions \tilde{f} . Uniform mutation and 1-point crossover are used to produce offspring [4]. To prevent running out of codons, we limit the maximum tree depth by preemptively excluding rules at every tree node. We choose the next generation’s parents via tournament selection on the combined parent and offspring population. For this we assess the loss value of each function, given by the distance of its outputs $\tilde{f}(x)$ to the ground-truth output images y , averaged over all m_i training pairs,

$$\mathcal{L}(f|D_i) = \frac{1}{m_i} \sum_{k=1}^{m_i} d_{\text{img}}(f(x^k), y^k). \quad (1)$$

Here we define d_{img} as (i) the proportion of correctly colored pixels (if images have equal size) or (ii) the Euclidean distance between the color histograms:

$$d_{\text{img}}(x, y) = \begin{cases} \sum_i \mathbf{1}_{\{x_i \neq y_i\}} / (u_x v_x) & \text{if } u_x = u_y \text{ and } v_x = v_y \\ 1 + \|\phi(x) - \phi(y)\|_2 & \text{else} \end{cases} \quad (2)$$

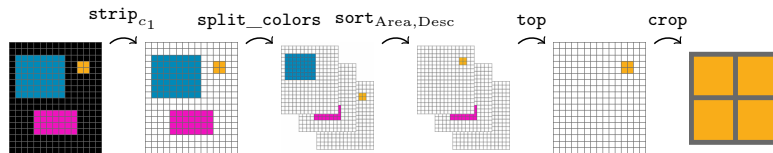


Fig. 3: Solver for task displayed in Figure 1a. It first strips away black (c_1) pixels, then splits the image into layers according to the remaining colors, and finally crops to the layer with smallest surface area.

where u, v is the image width and height, $\mathbf{1}_{\{P\}} = 1$ if $P \equiv \top$ else 0, and $\phi(x) = \sum_{x_i \in x} (\mathbf{1}_{\{x_i=c\}})_{c \in C}^\top$ as the (unnormalized) color histogram of x . \tilde{f} is an optimal solution if (and only if) all predictions are equal to the expected output, i.e. $\mathcal{L}(f|D_i) = 0$, thus we can also use Equation 1 as an early-stopping criterion.

4 Experimental Results

Following the evaluation procedure of the challenge, our accuracy is the proportion of correctly solved tasks: $\text{ACC} = M^{-1} \sum_{i=1}^M \prod_{\ell=1}^{n_i} \mathbf{1}_{\{f_i(\hat{x}^\ell) = \hat{y}^\ell\}}$. Here, f_i is the solution produced by our method after training on task D_i . A grid search was performed to obtain good hyperparameters for population size, mutation rate and mutation strength. As GE is randomized, we ran the experiments with 40 different seeds and discuss the averaged results with standard deviation.

We first evaluate our method on the 400 training tasks of ARC, from which we solved $7.68(\pm 0.61)\%$. The challenge leaderboard evaluation however is based on a secret data set of 100 tasks with slightly higher logic complexity. Here, we were able to correctly solve $\text{ACC} = 3\%$ of the tasks. Despite this seemingly low value, we scored among the top 30 of over 900 participants, which illustrates the challenge’s non-triviality. The small number of tasks makes it hard to assess the usefulness of atomic functions in the DSL. Moreover, there is no information about the overlap of required logic operations between training and test tasks.

Our results also made us question the effectiveness of GE considering the tremendous search space complexity. Small mutations to the current best individual, such as swapping out a single atomic function, can significantly change its behavior. As a result, the algorithm operates in a highly non-convex search space. We therefore compared our EA approach to simply generating random individuals from our DSL. This random search baseline solves an average of $6.17(\pm 0.13)\%$ of the training tasks, indicating that the EA is able to traverse the search space at least somewhat more efficiently. This is even more significant on the test tasks, where random-search is not able to solve even a single task.

5 Conclusion

We showed that our DSL+GE approach is viable for solving reasoning tasks. By designing a language for image-based logic and learning corresponding expressions from the training instances, we were able to score well in the ARC challenge. Ensuring a high expressiveness of the DSL, while at the same time limiting the complexity to allow tractable function evolution, appears to be the key problem. Finding the optimal set of functional atoms for the given domain may thus be subject to further research.

The ARC challenge’s difficulty illustrates the long way still to go until ML methods reach abstract reasoning capabilities comparable to the human mind’s.

Acknowledgment This research has been funded by the Federal Ministry of Education and Research of Germany as part of the competence center for machine learning ML2R (01|18038A)

References

1. Chollet, F.: The measure of intelligence. arXiv preprint arXiv:1911.01547 (2019)
2. Gulwani, S.: Automating string processing in spreadsheets using input-output examples. *ACM Sigplan Notices* **46**(1), 317–330 (2011)
3. Hernández-Orallo, J., Martínez-Plumed, F., Schmid, U., Siebers, M., Dowe, D.L.: Computer models solving intelligence test problems: Progress and implications. *Artificial Intelligence* **230**, 74–107 (2016)
4. Koza, J.R.: Genetic programming as a means for programming computers by natural selection. *Statistics and Computing* **4**, 87–112 (1994)
5. Legg, S., Hutter, M.: A collection of definitions of intelligence. *Advances in Artificial General Intelligence: Concepts, Architectures and Algorithms* **157** (07 2007)
6. O’Neill, M., Ryan, C.: Grammatical evolution. *IEEE Transactions on Evolutionary Computation* **5**(4), 349–358 (2001)
7. Ryan, C., Collins, J.J., Neill, M.O.: Grammatical evolution: Evolving programs for an arbitrary language. In: *European Conference on Genetic Programming*. pp. 83–96. Springer (1998)

Grace – Limiting the Number of Grid Cells for Clustering High-Dimensional Data

Anna Beer, Daniyal Kazempour, Julian Busch,
Alexander Tekles, and Thomas Seidl

Ludwig-Maximilians-Universität München, Munich, Germany
{beer,kazempour,busch,seidl}@dbs.ifi.lmu.de
alexander.tekles@campus.lmu.de

Abstract. Using grid-based clustering algorithms on high-dimensional data has the advantage of being able to summarize datapoints into cells, but usually produces an exponential number of grid cells. In this paper we introduce Grace (using a *Grid* which is *adaptive for clustering*), a clustering algorithm which limits the number of cells produced depending on the number of points in the dataset. A non-equidistant grid is constructed based on the distribution of points in one-dimensional projections of the data. A density threshold is automatically deduced from the data and used to detect dense cells, which are later combined to clusters. The adaptive grid structure makes an efficient but still accurate clustering of multidimensional data possible. Experiments with synthetic as well as real-world data sets of various size and dimensionality confirm these properties.

Keywords: Grid-based, Clustering, High-dimensional

1 Introduction

Clustering is one of the most important and well investigated unsupervised data mining tasks. Nevertheless, some problems related to the curse of dimensionality are still not solved. Grid based approaches suffer not only from the exponentially increasing number of cells in relation to the number of dimensions, but also from the incoherence between data and grid structure. As many real-world datasets have high dimensional feature spaces, being able to handle many dimensions is quite important for clustering algorithms.

Even though subspace clustering algorithms focus on high-dimensional data, they assume clusters to be in a low dimensional subspace of the data and are thus not suitable to find clusters lying in the full-dimensional space. Density based approaches on the other hand find clusters in full-dimensional space where all dimensions are equally important, but cannot handle high-dimensional data. Thus,

Copyright © 2020 by the paper's authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

efficient clustering of high-dimensional data without any a priori knowledge is still a huge challenge.

Hence we developed a grid-based clustering approach for high-dimensional data. It works fully automatically and finds clusters in full-dimensional data space without creating an exponential number of cells. The grid adapts itself to the data in regards of cell size as well as individual number of cells per dimension by using information from one-dimensional projections of the data. This leads not only to an adequate quality of the clustering results, but at the same time facilitates high efficiency of the algorithm.

Our main contributions are as follows:

- We develop Grace, a new grid-based clustering algorithm.
- By constructing the adaptive grid gradually, we are, to the best of our knowledge, the first ones to circumvent an exponential number of grid cells in relation to the number of dimensions.
- Grace is efficient and detects clusters of arbitrary shape in high-dimensional space.

The rest of the paper is structured as follows. Section 2 provides an overview of related work in the field of density-based and grid-based clustering. The algorithm itself is described in Section 3 and evaluated theoretically as well as empirically in Section 4. A brief conclusion is finally provided in Section 5.

2 Related Work

Since there exists a wealth of literature in the field of density-based as well as grid-based clustering, this section aims to provide an overview on some of the existing methods. We shall provide a brief elaboration on the core ideas behind each of the methods revealing the distinctive properties of our method in contrast to the competitors.

2.1 Density-based Clustering Approaches

Density-based clustering methods detect dense regions which are enclosed and separated by sparse regions and are thus suitable to find arbitrarily shaped clusters. Most density-based methods rely on local densities based on distances in the full-dimensional data space. The most common density-based approach is DBSCAN [7], which considers points with at least *minPts* points in their ε -range as core points. Core points are connected if their distance is lower than ε and form a cluster. Not-core points either lie in the ε -range of a core point and get assigned to the cluster of the core point, or are declared noise. DBSCAN is quite sensitive to ε and *minPts*, which are two parameters hard to guess for a user without detailed knowledge of the data. OPTICS [3] improves DBSCAN by introducing a reachability plot based on *minPts*, on which users can see the cluster structure and choose appropriate ε .

DENCLUE [11] uses local densities to compute an overall density function, the maxima of which constitute density attractors. Every object is connected to such a density attractor by means of a hill-climbing procedure. A threshold ξ gives the minimum density level for a density-attractor which allows to find noise. To accelerate the calculation of local densities, a simple grid with the same cell width 2σ in all dimensions is used, where σ is a user given parameter.

Other density-based clustering algorithms usually build on the approaches presented so far and aim to improve or extend them. HDBSCAN [5] for example extends DBSCAN to a hierarchical approach allowing different levels of density that can detect clusters of different density or nested clusters, overcoming the aforementioned issue of one global hyperparameter setting. DeLiClu [1] and SOPTICS [15] represent algorithms with purposes similar to OPTICS, but with improvements regarding their efficiency. Likewise, DENCLUE 2.0 [9] is a straightforward improvement of DENCLUE with reduced runtime complexity.

2.2 Grid-based and Subspace Clustering Approaches

Grid-based clustering methods generally partition the data into cells of different densities by dividing each dimension into several intervals. Those cells can then be connected into clusters without having to look at each data point again, which decreases the runtime. The results are often highly dependent on the structure of the constructed grid and most algorithms require users to set the defining parameters. STING [14] proposes a quite interesting hierarchical grid structure based on statistical information, but is neither used for clustering, nor does it deliver exact values, but rather approximations. Also, the distribution type of the data has to be known or ascertained by hypothesis tests.

Most grid-based clustering techniques produce subspace clusterings i.e. they detect subspaces of a high-dimensional data space which contain clusters of the given data. Grid-based approaches are well-suited for this task because they can easily exploit the monotonicity of the clustering criterion regarding dimensionality. This criterion implies that a k -dimensional cell is dense only if every $(k - 1)$ -dimensional projection of the cell is also dense, given a constant density threshold for the number of objects in a cell.

One of the first clustering algorithms to implement a grid-based subspace clustering was CLIQUE [2]. After constructing a grid with the same number of equidistant intervals in each dimension and identifying the dense cells, CLIQUE employs a bottom-up approach to find subspaces with dense regions by joining cells in k -dimensional spaces to candidate cells in $(k + 1)$ dimensions. If the number of objects in such a candidate cell exceeds a given density threshold, the corresponding $(k + 1)$ -dimensional space is considered a relevant subspace. After detecting the relevant subspaces, CLIQUE connects adjacent dense cells in their corresponding subspaces.

However, CLIQUE does not take the data distribution into account for generating the grid structure or for finding the dense regions. One subspace clustering approach that considers the data distribution beforehand is FIRES, which detects clusters on the basis of a greedy heuristics merging one-dimensional clusters

in order to find approximations of subspace clusters. This significantly reduces the runtime complexity of FIRES compared to CLIQUE.

While FIRES does not employ a grid structure at all, MAFIA [8] incorporates data distribution by using an *adaptive grid* in order to produce a better partitioning of the dimensions and reduce the number of grid cells. MAFIA determines the intervals in each dimension on the basis of one-dimensional histograms. Adjacent bins of a histogram are joined if they have approximately the same frequency. This yields larger intervals in the dimensions, each with roughly constant (one-dimensional) density. Nevertheless, MAFIA requires two parameters that may have a significant impact on the results and there is no guaranteed bound on the number of grid cells. On its adaptive grid, MAFIA proceeds like CLIQUE.

A general framework for clustering high-dimensional data on the basis of an adaptive grid is provided by OptiGrid [10] which recursively splits the data set by means of separating hyperplanes which should cut through low-density regions and separate high-density regions. The resulting cells already represent clusters, given a sufficiently high density. Though different approaches exist for selecting suitable hyperplanes [10, 6], these methods are not able to detect arbitrarily shaped clusters since generated cells already represent clusters and are not combined. Further, these methods require setting parameters whose impact on the result is difficult to assess a priori.

Further grid-based methods include SCHISM [17] which addresses the question of how to define and detect statistically interesting subspaces in high-dimensional data. As a measure for interestingness, the authors rely on the Chernoff-Hoeffding bound and use it for pruning. WaveCluster [18] relies on discrete wavelet transformation. The data is mapped to the frequency domain where clusters are then found by detecting dense regions. The method is insensitive to outliers and has a runtime complexity linear in the number of data objects. Among the most recent grid-based methods, ITGC [4] is an information theoretic approach regarding clustering as a data compression task. As such, neighboring grid cells are merged if it is beneficial with respect to compression costs.

3 Efficient Grid-based Clustering of Multi-dimensional Spaces

In this section we describe Grace in detail. In 3.1 we explain how the non-equidistant regular grid is generated dependent on the respective dataset. Section 3.2 shows how dense grid cells are combined to form clusters.

3.1 Generation of Adaptive Grids

The grid generation process is designed to limit the number of generated cells depending on the number of data points N and still allow for an accurate detection of clusters. For that we first estimate the density of each dimension

separately and then split the data space iteratively based on these estimations until the number of cells exceeds $N \cdot \log(N)$. To reduce runtime for calculating local changes in one-dimensional densities, we consider a histogram with $b = \max(50, \sqrt{N/d})$ equi-width bins for every dimension instead of all points separately. A maximum of 50 bins for each histogram has shown to produce appropriate grid structures for various data distributions and different numbers of points N . Higher N imply possibly more complex shaped clusters requiring a higher granularity of the histogram. A higher number of dimensions d in contrast results in a lower number of bins, since high dimensionality implies less expressiveness of distances and less bins allow for higher deviations.

Estimation of Local Changes in One-Dimensional Densities

Next, we compute for each bin in all one-dimensional histograms a *local change indicator* to express the local change of the one-dimensional density. To this end, we measure local changes in density as differences of frequencies between areas left and right to a particular bin and additionally set them in relation to the frequencies in their respective areas to distinguish random variations from relevant density shifts. The relevant neighborhood left and right of a particular bin is determined dynamically based on the frequency f covered by this area. The idea is to add less bins if the local density is already high, such that the separating hyperplanes will be lying closer together. Adjacent bins left and right are added iteratively until f exceeds a threshold t which is adjusted after each step. The threshold primarily depends on f and N such that a certain fraction of all objects needs to lie within the neighborhood range to stop expanding it. To avoid building large low-density cells, the neighborhood frequency is further weighted with the width of the current neighborhood range h , leading to a threshold

$$t = \frac{1}{h \cdot (1/b) \cdot (N/d)} \cdot N. \quad (1)$$

These weighted frequencies are used both for computing the differences between left and right areas as well as for determining the neighborhood area. As a

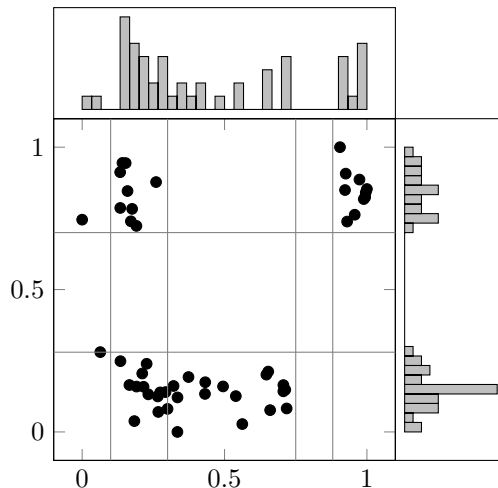


Fig. 1: Two-dimensional data with histograms as approximations of the one-dimensional data projections and edges chosen respectively

consequence, the density close to a bin has more impact on these computations than more distant bins. Figure 1 illustrates the generation of an adaptive grid based on the one-dimensional data projections as described so far.

Selection of Intervals After the one-dimensional histograms are computed and a change indicator is assigned to each bin, separating hyperplanes are selected iteratively until the number of cells generated by these planes is greater than or equal to $N \cdot \log(N)$. If d is high, not all dimensions are considered in order to limit the number of cells, i.e. some dimensions are not split by a separating plane. A dimension is only considered if at least two edges are chosen for this dimension. With only one edge in a dimension, i.e. two intervals, all data points would lie in the same interval or in adjacent intervals with respect to this dimension. In both cases, the corresponding dimension would have no informative content. This grid generation method yields at max $3 \cdot N \cdot \log(N)$ cells (see Theorem 1).

Theorem 1. *Given a d -dimensional data set with N objects. A regular grid in the d -dimensional space that is constructed by iteratively adding cutting hyperplanes, consists of at most $3N$ cells, if the grid generation process is stopped as soon as the number of cells is greater than or equal to N .*

Proof. Suppose h_{dim} edges are already chosen for dimension $dim \in 1, \dots, d$. If $h_k \leq 1$, dimension k is not yet considered due to the minimum of two edges in a dimension for it to be considered. Let $D_c \subseteq 1, \dots, d$ be the set of indices of the dimensions already considered. Dimension k is divided into $b_k = (h_k + 1)$ intervals. The number of cells c in the grid is computed by multiplying the number of intervals in all dimensions:

$$c = \prod_{dim \in D_c} b_{dim}$$

When adding a hyperplane separating dimension k to increase the number of cells from c to c' , three cases can occur. Before adding the separating plane, $c < N$ holds.

Case 1: $h_k > 1$

If $h_k > 1$, the number of intervals on the corresponding coordinate axis increases by one as well. This yields:

$$\begin{aligned} c' &= \prod_{dim \in D_c} b'_{dim} = (b_k + 1) \cdot \prod_{dim \in D_c \setminus k} b_{dim} \\ &= \left[b_k \cdot \prod_{dim \in D_c \setminus k} b_{dim} \right] + \left[1 \cdot \prod_{dim \in D_c \setminus k} b_{dim} \right] \\ &\leq c + c = 2 \cdot c < 2 \cdot N < 3 \cdot N \end{aligned}$$

Case 2: $h_k = 0$

If $h_k = 0$, the number of cells does not increase at all, as dimension k is not considered yet and will not after this iteration. For the relationship between c and c' it holds, that: $c' = c < N < 3N$

Case 3: $h_k = 1$

If $h_k = 1$, k is a new dimension to be considered, as $h'_k = h_k + 1 = 2$ and $b'_k = b_k + 1 = 3$ respectively. The number of cells therefore increases by the factor 3. $h'_k = 2 \Rightarrow k \in D_c$

$$c' = \prod_{dim \in D_c} b'_{dim} = b_k \cdot \prod_{dim \in D_c \setminus k} b_{dim} = 3 \cdot \prod_{dim \in D_c} b_{dim} = 3 \cdot c < 3 \cdot N$$

□

Every bin of the initial histogram represents a potential edge for splitting. In every iteration, the bin with the maximum local change indicator is chosen as the next edge. To choose edges closer to cluster borders, we make a small adjustment after selecting an edge: The edge is shifted in one direction as long as two successive bins have a frequency difference below 5% and we choose the direction to which the edge needs to be shifted less. After selecting an edge, we discard all bins within the neighborhood if the selected bin from the set of potential edges. This step avoids high granularity of the grid in areas of density changes. Algorithm 1 summarizes the grid creation.

Algorithm 1 CreateGrid

```

b ← max(50, √N/d)
for all dimensions do
    generate histogram with b bins
end for
for all dimensions do
    for all bins of histogram do
        determine neighborhood range
        compute local change indicator
    end for
end for
sort the bins of all histograms w.r.t. the local change indicator
moreEdgesNeeded ← true
selectedEdges ← {}
while moreEdgesNeeded do
    select the bin with highest local change indicator
    shift the bin if the adjacent bins have approximately similar frequencies
    add the edge to selectedEdges
    discard the bins within the neighborhood range of the selected bin from the set of
    potential edges
    if grid generated by selectedEdges contains more than N · log(N) cells then
        moreEdgesNeeded ← false
    end if
end while

```

3.2 Simple Connection of Dense Grid Cells to Clusters

Given the generated grid, the next steps involve detection of dense grid cells and subsequent combination of adjacent dense cells. For Grace, we apply wider notion of adjacency than existing works: Coordinates may differ at most by one in all dimensions – compared to a narrow notion, where the coordinates of adjacent bins may differ by one only in exactly one dimension.

A cell of volume V is considered dense, if it contains more than $minPts/V$ points for a $minPts$ given by the user. To avoid setting the density threshold too high for clusters with lower density, we first determine the most dense cells. To this end, we identify all cells containing more points than they would in expectation assuming a uniform distribution. In a second step, we discard these cells and detect the remaining dense cells with lower density using a new threshold.

Detection of dense cells is straightforward. After discretizing all data points to the grid, the number of data points in each grid cell is counted and compared to the threshold of the particular grid cell. Given an adequate grid structure, the number of dense cells p is usually much lower than N . Adjacent dense cells are now connected to form clusters by extracting connected components from the graph represented by the symmetric $p \times p$ adjacency matrix M with $M_{i,j}$ if and only if cells i and j differ by exactly one dimension. Adjacent cells can be found by sorting the dataset in every dimension and then iterating over the dimensions.

3.3 Connection of Diagonally Adjacent Grid Cells

So far, only adjacent cells in the narrower sense have been connected. To connect cells adjacent in the wider sense, i.e., diagonally adjacent cells, we add additional helper cells next to dense cells. Given a cell \tilde{a} , that is either a dense cell or a previously added helper cell with coordinates (a_1, a_2, \dots, a_d) and the order of dimensions considered for the current sorting of the coordinates $d_{i_1}, d_{i_2}, \dots, d_{i_d}$, a new helper cell (b_1, b_2, \dots, b_d) with $b_k = a_k \forall k \in \{i_1, \dots, i_{d-1}\}$ and $b_{i_d} = a_{i_d} + 1$ is now added to the set of helper cells if it has not yet been added before. New helper cells are not considered in the same iteration they were added, but in subsequent iterations they are treated just like the original dense cells. This ensures to find exactly all connections between originally adjacent dense cells in the wider sense.

Theorem 2. *Given a d -dimensional grid with a set P of dense cells and an initially empty set of helper cells H , both of whom are iterated d times. In every iteration i , a cell b with coordinates $b_k = a_k \forall k \in \{1, \dots, d\} \setminus i$ for each cell $a \in P \cup H$ that has no adjacent cell with the coordinates of b is added to the set of helper cells after the current iteration. With this procedure, two dense cells that are adjacent in the wider sense can be connected, either directly or indirectly with the help of other cells in $P \cup H$, by just applying the notion of adjacency in the narrow sense.*

Proof. Given two dense cells a and b with coordinates (a_1, a_2, \dots, a_d) and (b_1, b_2, \dots, b_d) , respectively.

Case 1: a and b are adjacent in the narrow sense

Adjacency in the narrow sense implies adjacency in the wider sense by definition, thus the two cells are also adjacent in the wider sense.

Case 2: $a_i \in \{b_i, b_i - 1\}, \forall i \in \{1, \dots, d\}$

Suppose the cells' coordinates differ in dimensions $\{j_1, j_2, \dots, j_m\} \subseteq \{1, 2, \dots, d\}$ with $j_s < j_t \forall s < t$. In iteration j_1 , the cell $a_{(1)}$ with coordinates $(a_1, \dots, a_{j_1} + 1, \dots, a_d)$ is either added as helper cell or already a dense cell. Since this cell differs by one from a in exactly one dimension, it is adjacent to a in the narrow sense. In iteration j_2 , the cell $a_{(2)}$ with coordinates $(a_1, \dots, a_{j_1} + 1, \dots, a_{j_2} + 1, \dots, a_d)$ is again either added as helper cell or already a dense cell. The new cell is now adjacent to the previously added cell $a_{(1)}$. This step is then repeated for all $j \in \{j_1, j_2, \dots, j_{m-1}\}$. Finally, the cell $a_{(m-1)}$ with coordinates $(a_1, \dots, a_{j_1} + 1, \dots, a_{j_2} + 1, \dots, a_{j_{m-1}} + 1, \dots, a_d)$ is either added as helper cell or already a dense cell. $a_{(m-1)}$ differs in exactly one dimension from b , so that $a_{(m-1)}$ and b are adjacent in the narrow sense and thus a and b .

Case 3: $a_i \in \{b_i, b_i + 1\}, \forall i \in \{1, \dots, d\}$

Switching a and b converts this case to the same as case 2.

Case 4: $a_i \in \{b_i, b_i - 1\}, \forall i \in I \subset \{1, \dots, d\}$ and $a_j \in \{b_j, b_j + 1\} \forall j \in J = \{j_1, j_2, \dots, j_m\} \subset \{1, \dots, d\} \setminus I$

In this case, two chains of adjacent cells can be constructed, each following the same idea as in case 2 or in case 3 respectively. The first chain starts from cell a , considering all dimensions with $a_i = b_i + 1$, which corresponds to case 2. The second chain starts from cell b , considering all dimension with $a_i = b_i - 1$, which corresponds to case 3. Finally, without loss of generality, the coordinates of the last cell \tilde{a} in the first chain are of the form $(\tilde{a}_1, \tilde{a}_2, \dots, \tilde{a}_d)$ with $\tilde{a}_i = \tilde{a}_i + 1 = b_i \forall i \in I$ and $\tilde{a}_k = b_k \forall k \in \{1, \dots, d\} \setminus I$. The coordinates of the last cell \tilde{b} in the second chain are of the form $(\tilde{b}_1, \tilde{b}_2, \dots, \tilde{b}_d)$ with $\tilde{b}_j = \tilde{b}_j + 1 = a_j \forall j \in \{j_1, \dots, j_{m-1}\}$, $\tilde{b}_{j_m} = a_{j_m} + 1$ and $\tilde{b}_k = a_k \forall k \in \{1, \dots, d\} \setminus J$. Now, these two last cells of both chains differ only in dimension j_m by one, so that they are connected in iteration j_m .

□

4 Evaluation

We investigate Grace from a theoretical as well as from an empirical point of view. In Section 4.1 we calculate the runtime complexity and in 4.2 we present and discuss several experiments based on synthetic as well as real world data sets and compare the results with DBSCAN and CLIQUE.

4.1 Complexity Analysis

The histograms for each dimension can be computed in time $O(N \cdot d)$. For each of the $b \cdot d$ histogram bins, the local change indicator can be computed in constant

time, leading to $O(b \cdot d)$. Having $b \leq \sqrt{N \cdot d}$, we get $O(\sqrt{N \cdot d} \cdot d) = O(\sqrt{N} \cdot d^2)$. Finding separating hyperplanes requires sorting the $b \cdot d$ local change indicators, which can be done in time $O(b \cdot d \cdot \log(b \cdot d)) = O(\sqrt{N \cdot d} \cdot d \cdot \log(\sqrt{N \cdot d} \cdot d)) = O(N \cdot d^2)$.

The dense cells are determined by counting for every point the occurrences of each coordinate combination, which is in $O(N \cdot d)$. Dense cells can be combined efficiently by sorting them in every dimension to identify adjacent grid cells in that dimension. Since the maximum number of dense cells is $N \cdot \log(N)$, the complexity of sorting the dense cells is $O(N \cdot \log(N) \cdot \log(N \cdot \log(N)) \cdot d) = O(N \cdot \log^2(N) \cdot d)$. Adjacent dense cells can be identified in each dimension by comparing consecutive cells in the sorted order with complexity $O(N \cdot d)$. Thus, all connections between dense grid cells can be detected in time $O(N \cdot \log^2(N) \cdot d)$. Connected components can be identified in time $O(N^2 \cdot \log^2(N))$. In total, the complexity is thus

$$O(N \cdot d + \sqrt{N} \cdot d^2 + N \cdot d^2 + N \cdot \log^2(N) \cdot d + N^2 \cdot \log^2(N)) = O(N^2 \cdot \log^2(N) \cdot d^2).$$

Note, that the number of dense cells p which is responsible for the $N^2 \cdot \log^2(N)$ part is usually far lower than N .

4.2 Empirical

The following experiments have been conducted on a Linux machine with a commodity hardware featuring a 2.0 GHz CPU with two cores and 3.6 GB RAM. As Grace uses elements from density-based as well as grid-based clustering, we compare our results to those DBSCAN and CLIQUE. Where Grace works fully automatically, DBSCAN and CLIQUE both need two parameters, for which those yielding the best results were chosen. For DBSCAN we used the scikit-learn Python library implementation and for CLIQUE the implementation from the data mining framework ELKI [16].

For a first visual interpretation, we show the effectivity of Grace working on two simple two-dimensional synthetic data sets containing density-based clusters and compare the results to those of DBSCAN. Figure 2 shows the clustering result for “TARGET” [19] with $N = 770$, consisting of four small clusters distributed at opposing corners as seen in Figure 2, one of them being detected as noise (bottom left cluster). Apart from the detected noise cluster, all other clusters are detected correctly by Grace. With a proper parameter setting, DBSCAN is capable of detecting all clusters, too. “CLUTO-T8-8K” contains 8000 two-dimensional objects [12]. Applied to this data set, the Grace found some, but not all clusters similar to DBSCAN.

Another synthetic data set with $N = 101,000$ and $d = 9$ containing three spherical clusters and 1,000 noise objects has been created to show the scalability of Grace. The clusters of this data set are found in mere 4.5 seconds, where, due to excessive memory consumption, neither DBSCAN nor CLIQUE could be applied to this data set with the machine used here. To compare at least CLIQUE with the proposed approach in high dimensions, another data set with

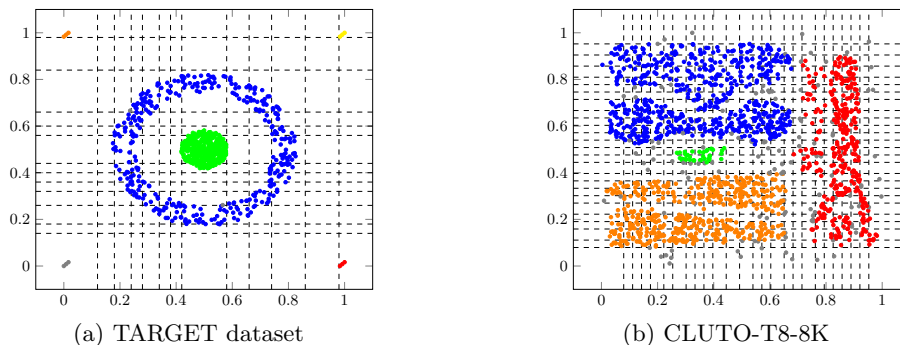


Fig. 2: Clustering results using an adaptive grid (represented by the dashed lines). Noise is colored gray.

$N = 100,000$ and $d = 8$, also containing three spherical clusters, is clustered by the two algorithms. They both detect all three clusters, where Grace was almost three times as fast as CLIQUE (1.7 vs 4.8 seconds).

Finally, a data set derived from the “VICON” data set containing physical action data measuring human activity [13] has been tested. The actions are measured by means of nine sensors on different body parts, each emitting three-dimensional spatial data. In summary, this yields a data set with 27 dimensions. For this experiment, the two actions punch and handshake have been merged into one data set with 5045 objects. Again, neither the DBSCAN implementation nor the CLIQUE implementation used can be applied to this data set on the machine used due to excessive memory consumption. Grace detected an accurate clustering within 232ms, with one cluster being detected 100% and the other cluster being split with 92% of it being grouped in one cluster.

5 Conclusion

Grace finds clusters in multidimensional data spaces where points build a cluster if they are close in all dimensions. It generates an adaptive grid structure that makes it possible to reduce the runtime complexity significantly for multidimensional data spaces compared to similar grid-based approaches. The experimental evaluation has shown that the algorithm outperforms DBSCAN and CLIQUE for large data sets and high dimensions. Grace works fully automatically and can be applied to datasets of various sizes, dimensionalities, and cluster densities. For clustering high-dimensional data with all dimensions being relevant for forming the clusters, it is an efficient alternative to established algorithms. It is moreover a possibility to get some first insights if no information about the data is available yet, since no expert knowledge about the data is needed beforehand due to the absence of any parameters. In future work noise should be handled separately and we are also investigating the suitability for anytime results. The

grid construction is promising for many other applications, and could, e.g., be applied in context of arbitrarily oriented correlation clusters.

Acknowledgments

This work has been partially funded by the German Federal Ministry of Education and Research (BMBF) under Grant No. 01IS18036A. The authors of this work take full responsibilities for its content.

References

1. Achtert, E., Böhm, C., Kröger, P.: Deliclu: Boosting robustness, completeness, usability, and efficiency of hierarchical clustering by a closest pair ranking. In: PAKDD (2006)
2. Agrawal, R., Gehrke, J., Gunopulos, D., Raghavan, P.: Automatic subspace clustering of high dimensional data for data mining applications. SIGMOD (1998)
3. Ankerst, M., Breunig, M.M., Kriegel, H.P., Sander, J.: Optics: Ordering points to identify the clustering structure. In: SIGMOD (1999)
4. Behzadi, S., Hinterhauser, H., Plant, C.: Itgc: Information-theoretic grid-based clustering. In: EDBT (2019)
5. Campello, R., Moulavi, D., Sander, J.: Density-based clustering based on hierarchical density estimates. In: Pei, J., Tseng, V., Cao, L., Motoda, H., Xu, G. (eds.) PAKDD (2013)
6. Chang, J.W., Jin, D.S.: A new cell-based clustering method for large, high-dimensional data in data mining applications. In: SAC (2002)
7. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD (1996)
8. Goil, S., Nagesh, H., Choudhary, A.: Mafia: Efficient and scalable subspace clustering for very large data sets. In: KDD (1999)
9. Hinneburg, A., Gabriel, H.H.: Denclue 2.0: Fast clustering based on kernel density estimation. In: IDA (2007)
10. Hinneburg, A., Keim, D.: Optimal grid-clustering: Towards breaking the curse of dimensionality in high-dimensional clustering. In: VLDB (1999)
11. Hinneburg, A., Keim, D.: An efficient approach to clustering in multimedia databases with noise. In: KDD (1998)
12. Karypis, G., Han, E.H., Kumar, V.: Chameleon: A hierarchical clustering algorithm using dynamic modeling. IEEE Computer (1999)
13. Lichman, M.: UCI machine learning repository (2013), <http://archive.ics.uci.edu/ml>
14. Muntz, R., Wang, W., Yang, J.: Sting: A statistical information grid approach to spatial data mining. In: VLDB (1997)
15. Schneider, J., Vlachos, M.: Scalable density-based clustering with quality guarantees using random projections. DMKD (2017)
16. Schubert, E., Zimek, A.: Elki: A large open-source library for data analysis-elki release 0.7. 5" heidelberg". arXiv preprint arXiv:1902.03616 (2019)
17. Sequeira, K., Zaki, M.: Schism: A new approach for interesting subspace mining. In: ICDM (2004)
18. Sheikholeslami, G., Chatterjee, S., Zhang, A.: Wavecluster: A multi-resolution clustering approach for very large spatial databases. In: VLDB (1998)
19. Ultsch, A.: Clustering with som, u*c. In: WSOM (2005)

Segmenting and Clustering Noisy Arguments

Lorik Dumani^(✉) , Christin Katharina Kreutz^(✉) , Manuel Biertz^(✉) , Alex Witry^(✉) , and Ralf Schenkel^(✉) 

Trier University, 54286 Trier, Germany
{dumani,kreutzch,biertz,s4alwitr,schenkel}@uni-trier.de

Abstract. Automated argument retrieval for queries is desirable, e.g., as it helps in decision making or convincing others of certain actions. An argument consists of a claim supported or attacked by at least one premise. The claim describes a controversial viewpoint that should not be accepted without evidence given by premises. Premises are composed of Elementary Discourse Units (EDUs) which are their smallest contextual components. Oftentimes argument search engines find similar claims to a query first before returning their premises. Due to heterogeneous data sources, premises often appear repeatedly in different syntactic forms. From an information retrieval perspective, it is essential to rank premises relevant for a query claim highly in a duplicate-free manner. The main challenge in clustering them is to avoid redundancies as premises frequently address various aspects, i.e., consist of multiple EDUs. So, two tasks can be defined: segmentation of premises in EDUs and clustering of similar EDUs.

In this paper we make two contributions: Our first contribution is the introduction of a noisy dataset with 480 premises for 30 queries crawled from debate portals which serves as a gold standard for the segmentation of premises into EDUs and the clustering of EDUs. Our second contribution consists of first baselines for the two mentioned tasks, for which we evaluated various methods. Our results show that an uncurated dataset is a major challenge and that clustering EDUs is only reasonable with premises as context information.

1 Introduction

Computational argumentation is an important building block in decision making applications. Retrieving supporting and opposing premises for controversial claims can help to make informed decisions on the topic or, when seen from a different viewpoint, to persuade others to take particular standpoints or even actions. In line with existing work in this field, we consider arguments that consist of a claim that is supported or attacked by at least one premise [24]. The claim is the central component of an argument, and it is usually controversial [23]. The premises increase or decrease the claim’s acceptance [11]. The stance of a premise indicates if it supports (pro) or attacks (con) the claim. Table 1 shows an example for an argument consisting of a claim supported or opposed by premises.

Copyright © 2020 by the paper’s authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

Table 1. Example of a claim c and its premises p_1 , p_2 and p_3 .

var.	type	stance	content
c	claim	-	<i>Aviation fuel should be taxed</i>
p_1	premise	pro	<i>Less CO₂ emissions lead to a clean environment</i>
p_2	premise	con	<i>Higher taxes would not change anything</i>
p_3	premise	pro	<i>It does not matter that the costs for aviation are already high as the environment can be protected by less CO₂ emissions</i>

In the NLP community researchers either address argument mining, i.e., the analysis of the structure of arguments in natural language texts (see the work of Cabrio and Villata [4] for an overview of recent contributions), or an information-seeking perspective, i.e., the identification of relevant premises associated with a predefined claim [19]. Due to the rapidly increasing need for argumentative queries, established search engines that only retrieve relevant documents will no longer be sufficient. Instead, argument search engines are required that can provide the best pro and con premises for a query claim. In fact, various argument search engines [27,22] have recently been developed. These systems usually work on claims and premises that were either mined from texts beforehand or extracted from dedicated argument websites such as idebate.org. Their workflow usually starts with finding *result claims* similar to the query claim. Then they locate the *result premises* belonging to these claims to present them as output.

However, these systems face a number of challenges since claims and premises are formulated in natural language. First, premises that are semantically (mostly) equivalent occur repeatedly in different textual representations since they appear in different sources, but should be retrieved only once to avoid duplicates. This requires the clustering of similar premises for result presentation. Second, discussions on debate portals, but also in natural language arguments are often not well-structured, such that a single supporting or attacking piece of text can address several aspects and thus should be represented as multiple premises. For example, a sentence supporting the viewpoint that aviation fuel should be taxed could address two aspects, the potential danger for the environment and the current low tax rate on aviation fuel. Directly using such sentences as formal premises, as seen in premise p_3 in Table 1, would make it impossible to retrieve a duplicate-free and complete list of premises.

This issue can be avoided by dividing the premises into their core aspects and clustering them instead of whole premises. In the literature, the smallest contextual components of a text are called *Elementary Discourse Units (EDUs)* [24]. Obtaining high quality EDUs [24] from text (discourse segmentation) is a crucial task preceding all efforts in parsing or representing discourses [21]. Thereby, it takes a pragmatic perspective, i.e., links between discourse segments are established not on semantic grounds but on the author’s (assumed) intention [17]. For the explorative purposes outlined here, only the concept of EDUs as smallest, non-overlapping units of intra-text-discourse – mostly clauses – is picked up [15].

In this paper we address the aforementioned limitations and deal with the segmentation of textual premises into EDUs and the clustering of EDUs based on their semantic similarity. Contrasting previous research on both of these tasks that worked with manually curated and thus high-quality argument col-

$\text{EDU}_1(p_1) = \text{“Less CO}_2 \text{ emissions lead to a clean environment”}$
 $\text{EDU}_1(p_2) = \text{“Higher taxes would not change anything”}$
 $\text{EDU}_1(p_3) = \text{“It does not matter that the costs for aviation are already high”}$
 $\text{EDU}_2(p_3) = \text{“as the environment can be protected by less CO}_2 \text{ emissions”}$

Fig. 1. EDUs extracted from premises in Table 1.

lections, we use a dataset that was crawled from debate portals [10]. Unlike other datasets, the premises in this dataset contain a considerably higher number of sentences and often cover multiple aspects (which is at odds with our generally micro-structural approach to arguments). In addition, as an uncurated real-world dataset, it contains many ill-formulated sentences and other defects. Our contribution is two-fold: First we provide a real-life dataset consisting of 480 premises retrieved for 30 query claims that are segmented into 4,752 EDUs. Then, for each query claim the belonging EDUs have been manually clustered by semantic equivalence. Second, we report our first results for the two tasks of EDU identification and EDU clustering on this dataset.

Our proposed method works as follows: for a given set of textual premises returned by an argument search engine for a query claim, we first identify the EDUs for each result. In the second step, we focus on the clustering of EDUs. To accomplish this, we first generate embeddings and then we cluster those with an agglomerative clustering algorithm. As an example, consider Table 1 again. Here, premise p_3 is composed of two EDUs $\text{EDU}_1(p_3)$ and $\text{EDU}_2(p_3)$ (see Figure 1). In addition to that, $\text{EDU}_2(p_3)$ and $\text{EDU}_1(p_1)$ (where $\text{EDU}_1(p_1)$ is the only EDU of p_1) have the same meaning and therefore should be assigned to the same cluster.

The remainder of this paper is structured as follows: Section 2 provides an overview of related work addressing the segmentation of argumentative texts into EDUs and clustering algorithms. In Section 3 the dataset and its manual annotation is described in more detail. Then, in Section 4 we present and evaluate our methods for extraction and clustering of EDUs. Section 5 concludes our work and provides future research directions.

2 Related Work

There is a plethora of research on *discourse segmentation* of text but to the best of our knowledge, existing approaches are designed for curated datasets. A rule-based approach including a post processing step for identification of starts and ends of EDUs was proposed by Carreras et al. [6]. Among other features they utilize chunks tags and sentence patterns. Soricut and Marcu [21] introduced a probabilistic approach based on syntactic parse trees. Tofiloski et al. [25] perform EDU segmentation based on syntactic and lexical features with the goal of capturing only interesting, not all EDUs. Here, every EDU is conditioned to contain a verb. Others suggest a classifier able to decide whether a word is the beginning, middle or end of a nested EDU using features derived from *Part of Speech* (POS) tags, chunk tags or dependencies to the root element [1]. In a recent paper, Trautmann et al. [26] also argue that “*spans of tokens*” rather than whole sentences should be annotated and define this task as Argument Unit

Recognition and Classification. We omit preprocessing of text and utilization of preconditions which is applicable to a supervised scenario as it might flaw an approach based on uncurated data as no guarantees can be made for a real-world, possibly defective, crawled dataset from debate portals.

The *clustering of similar arguments* is still a recent field of research. Boltuzic and Snajder [3] applied Word2Vec [16] with hierarchical clustering for debate portals. Reimers et al. [19] experiment with contextualized word embedding methods such as ELMO [18] and BERT [8] and show that these can be used to classify and cluster topic-dependent arguments. They use hierarchical clustering with a stopping threshold which is determined on the training set to obtain clusters of premises. However, they do not specify a concrete value. Further, Reimers et al. note that premises sometimes cover different aspects. Hence, we divide premises into their EDUs and cluster these instead. Like them, we also use uncurated data and make use of ELMO and BERT. We additionally utilize the embedding methods INFERSENT [7], and FLAIR [2]. Contrasting Reimers et al., we only consider relevant premises for the clustering as we intend to start with a step-by-step approach.

3 Dataset and Labeling

We make use of the argumentation dataset introduced in our prior work [10] where we crawled four debate portals and extracted claims with their associated textual premises. In a follow-up work [9], we built a benchmark collection for argument retrieval based on that dataset. In this former work, we picked 232 randomly chosen claims on the topic energy and used them as query claims to pool the most similar result claims retrieved by standard IR methods. In the latter [9], for 30 of these query claims, we collected the premises of all pooled result claims and manually assessed their relevance with respect to the query claim, using a three-fold scale (“very relevant”, “relevant”, “not relevant”). This resulted in 1,195 tuples of the form (query claim, result claim, result premise, assessment). Following the practice at TREC (Text REtrieval Conference), a premise is relevant if it has at least one relevant EDU, and very relevant if it contains no aspect not relevant to the initial query claim.

In this paper, we only included result premises that were assessed with “very relevant” or “relevant” to keep the effort for manual assessment reasonable. This means we consider 480 tuples for our new dataset. For each of these 480 result premises, the EDUs were identified by one annotator who is a research assistant from political science and has a deep understanding of argumentation theory. For this segmentation, the annotator followed the manual by Carlson and Marcu [5]. This resulted in a total of 4,752 EDUs for the 480 premises (on average 9.9 EDUs per premise), indicating that premises in debate portals usually cover plenty of aspects and segmentation is indispensable for argument retrieval and clustering.

In a next step, the EDUs were manually clustered by identifying semantically equivalent EDUs and putting them in the same cluster. This was done with support of a modified variant of the OVA tool [12] (<http://ova.uni-trier.de/>) for modeling complex argumentations, which was enhanced to be capable to store text positions. Since EDUs cannot be further divided by definition, clusters were

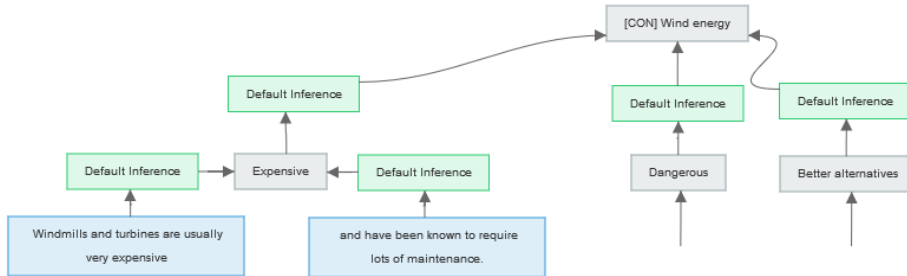


Fig. 2. Screenshot showing an excerpt of the OVA tool used for clustering similar EDUs. The blue nodes represent the EDUs, the gray nodes were added artificially by the annotator and represent the clusters. The green nodes are edges and represent the relations from the EDUs to the clusters they were assigned by the assessor.

formed manually that include all EDUs with the same meaning. For each of the 30 query claims, an OVA view was created where all EDUs identified in result premises for this query were represented as nodes. A human annotator then clustered these nodes by creating an artificial node for each cluster identified and then connecting all semantically identical EDUs to the cluster node by dragging edges. Additionally, to make the clustering more readable, the annotator created three artificial clusters “PRO”, “CON”, and “CLAIMS” and referenced the previously formed artificial clusters to them depending on their stance with respect to the query. In this paper we will not consider stances. However, since we are making the dataset available (on request), they can be important for further work, for example, for those who also want to use additional distinctions according to the stance.

Figure 2 illustrates a screenshot of the clustering annotation tool. Not all EDUs could plausibly be treated as a single premise (e.g., EDUs that are post-modifiers to noun phrases), thus we also allowed to mark EDUs as context information for other EDUs. For the clustering task, we clustered 1,044 EDUs for 11 queries, distributed to 622 clusters. Because of time constraints, we did not manage to cluster all EDUs of all 30 queries here, and instead only analyzed 11, which are after all more than 1,000 clustered EDUs. The annotators’ feedback was that the visualization helped to keep an overview as there were almost 100 EDUs per query to cluster.

4 Methodology and Evaluation

This section describes our approaches for segmenting premises into EDUs and clustering them, as well as an evaluation of the performance of these methods with respect to the ground truth. Figure 3 provides a schematic overview of the different steps. In general, our approach will retrieve clusters of EDUs for input query claims. Given a query claim q_i as well as similar result claims $c_{i,j}$ with associated premises $p_{i,j,k}$. These relevant result claims are retrieved by

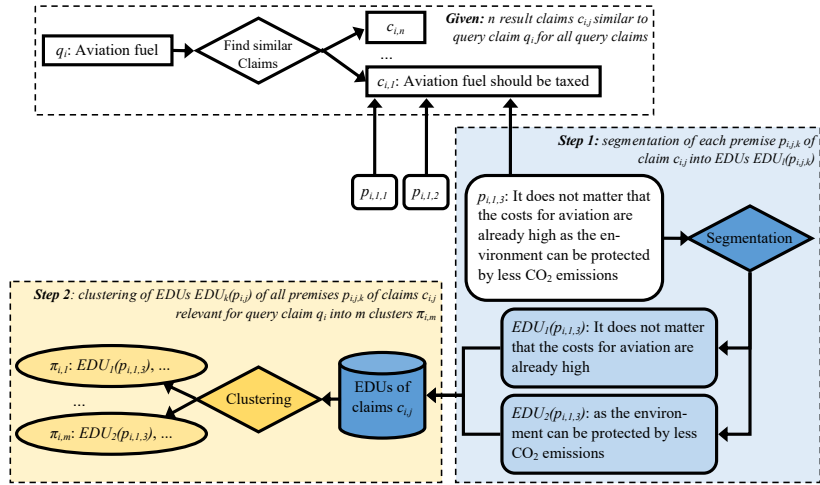


Fig. 3. Schematic overview of the two steps segmentation and clustering.

application of our prior work [10]. In the first step of this approach, premises are divided into EDUs, in the second step all EDUs of premises linked to result claims for our query claim will be clustered.

4.1 Step 1: Segmentation of Premises into EDUs

We first compare different approaches for segmentation of premises into EDUs to the ground truth segmentation from the 30 claims. We focus on basic segmentation methods generating sequential, i.e., non-overlapping EDUs in order to obtain insight into their performance on a real-world dataset as they are often used as a preprocessing step in more sophisticated segmenters [6,21,25,1].

As an initial baseline (*sentence baseline*), we split premises into sentences with CORENLP (stanfordnlp.github.io/CoreNLP) and considered each sentence as an EDU. As CORENLP also allows to extract a text’s PennTree, which contains the POS tag for each term and displays the closeness of the terms in a hierarchical structure, we also identified EDUs by cutting the PennTree of premises (*tree cut*) at height cutoffs from 1 to 10, denoted by $tc_{i,1 \leq i \leq 10}$ in the following. Additionally, we obtained subclauses from sentences which we also regarded as EDUs by applying TREGEX (nlp.stanford.edu/software/tregex) (*subclauses*). We also implemented a rule-based splitter (*splitter*) which does consider the peculiarities of our dataset but differs from the ground truth [5]. This splitter is kind of an extension of the sentence baseline, thus sentence boundaries and all kinds of punctuation marks are seen as discourse boundaries [21] so those are used to split premises into EDUs. Further, before conjunctions and terms or phrases indicating subclauses, boundaries are included.

Table 2 shows the performance of the different approaches which we compare in terms of their precision, recall, F_1 score, specificity, and accuracy. Out of the tree cut approaches, tc_3 , i.e., the tree cut with cutoff at height 3, obtained the

Table 2. Performance of the different methods to split text to EDUs. Precision, recall, F₁ score, specificity and accuracy.

Method	Prec	Rec	F ₁	Spec	Acc
SENTENCE BASELINE	0.3289	0.9561	0.4739	0.9140	0.4883
TREE CUT tc_3	0.6223	0.7064	0.6458	0.9472	0.5073
SUBCLAUSES	0.4150	0.3796	0.3905	0.9167	0.4934
SPLITTER	0.8523	0.562	0.6654	0.9768	0.5244

highest F₁ score. The rule-based splitter achieved the highest F₁ score of all tested methods. This method results in a high precision combined with a lower recall which is a property of conservative approaches [25]. The comparably poor results for the other approaches may occur since classical preprocessing steps are unfit for approximating human annotations on uncurated real-world datasets. A Kruskal-Wallis test¹ on F₁ scores for boundaries of EDUs computed for each premise of the sentence baseline, splitter and tc_3 holds for $p = 0.05$. Thus, the splitter method is significantly better than the other methods.

Evaluation of EDUs In order to evaluate the quality of EDUs obtained by the annotators as well as our best approaches we constructed triples of the form (EDU_{ground_truth}, EDU _{tc_3} , EDU_{splitter}) for 50 randomly chosen premises. Within each triple, the EDUs were ranked by their subjective perceived quality by a reviewer who is an expert in computational argumentation and familiar with argumentation theory. Note that it was not shown to the assessor how each EDU was determined and the ordering within triples was shuffled. The expert assessor assigned ranks from 1 to 3 with 1 being the best, ties were permitted.

The ground truth achieved an average rank of 1.66 (#1: 22 times, #2: 23 times, #3: 5 times), tc_3 did perform equally well (#1: 23 times, #2: 21 times, #3: 6 times). The splitter method performed considerably worse with an average rank of 2.64 (#1: 6 times, #2: 6 times, #3: 38 times). As the ground truth would be expected to outperform other approaches clearly, this outcome indicates firstly the difficulty in the annotation process, secondly the subjective perception of what is better and what is less good, and thirdly the difficulty in correctly capturing language with computers. Figure 4 shows an example of both the manually created EDUs and those created by the splitter method.

4.2 Step 2: Clustering of EDUs

In order to build clusters of EDUs automatically for each of the eleven claims, first we obtained the embedding vectors of EDUs using ELMO, BERT, FLAIR, and INFERSENT². For this task, we consider the segmentation of premises into

¹ A Kruskal-Wallis test was used as data in the three groups is not normally distributed; this was tested with a Shapiro-Wilk test.

² We used the implementations provided by <https://github.com/facebookresearch/InferSent> and <https://github.com/flairNLP/flair>.

EDUs by ground truth:

[From what I understand,] [the cheap oil is something] [that will not only effect the economy in the long run,] [but it will also hurt those] [who want to receive retirement or disability benefits at the federal level.] [It's great to finally have cheaper gas than that] [which was nearly \$3 in the past.] [I do think] [it might have an adverse effect on our economy.]

EDUs by splitter:

[From what I understand,] [the cheap oil is something] [that will not only effect the economy in the long run,] [but it will also hurt those] [who want] [to receive retirement] [or disability benefits at the federal level.] [It's great] [to finally have cheaper gas] [than] [that] [which was nearly \$3 in the past.] [I do think it might have an adverse effect on our economy.]

Fig. 4. Example of EDUs manually created and those by the method splitter. EDUs are encompassed by square brackets. Differently identified EDUs between ground truth and method splitter are underlined.

EDUs given by the ground truth of Section 4.1. Otherwise, an automatic external evaluation would be infeasible. We derived eight vectors per EDU and embedding technique by extending EDUs with context information, i.e., we obtained tuples (EDU, ctx) with context ctx from all combinations of the power set $\mathcal{P}(\{premise, result\ claim, query\ claim\})$. After that, we performed an agglomerative (hierarchical) clustering of the EDUs of all claims related to the query for each of the eleven queries as it is the state-of-the-art for clustering arguments [3,19]. Then, since we do not know the number of clusters a priori, we performed a dynamic tree cut [14]. The advantage of this approach over other approaches such as k -means is that there is no need to specify a final number of result clusters, which is not known in our case. The benefit of agglomerative clustering over divisible clustering is certainly the lower runtime. As a straightforward baseline, all EDUs from the same premise are assigned to the same cluster ($BL_{premiseAsCluster}$). Two additional baselines consist of one big cluster containing all EDUs ($BL_{oneCluster}$), as well as many clusters, each containing one EDU ($BL_{ownClusters}$). The quality of the clustering was measured with external and internal evaluation measures. While external evaluation measures base on previous knowledge, in our case the ground truth clustering formed by the assessor, the internal evaluation measures base on information that only involves the vectors of the datasets themselves [20].

With regard to the external cluster evaluation metrics, we measured the following three: the *purity*, the *adjusted mutual information* (AMI), and the *adjusted Rand index* (Rand). For the internal cluster evaluation, we measured the *Calinski-Harabasz index* (CHI) and the *Davies-Bouldin index* (DBI).³ Concise descriptions of the metrics can be found in Table 3. The results of the external and internal evaluations can be found in Table 4.

We can observe that $BL_{premiseAsCluster}$ outperforms all methods for the external evaluation measures except for the perfect purity of $BL_{ownClusters}$. In general $BL_{oneCluster}$ and $BL_{ownClusters}$ do not produce surprising results for the external evaluation. CHI and DBI are undefined for their number of clusters.

³ We used the implementations provided by <https://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics.cluster>.

Table 3. Descriptions of internal and external cluster evaluation metrics.

Type	Name	Brief description
<i>external</i>	purity	Measures the extent to which clusters contain a single class. Its value ranges from 0 to 1, with 1 being the best. Generally, a high (compared to the number of clustered entities) number of clusters results in a high purity.
<i>external</i>	AMI	Measures the mutual dependence between two random variables and qualifies the amount of information obtained about one random variable through observing the other random variable. Values are adjusted for chance. Its values range between 0 and 1, where 1 implies a perfect correlation.
<i>external</i>	Rand	Computes the accuracy and ranges from 0 to 1. It penalizes both false positive and false negative decisions with equal weights during clustering. Values are adjusted for chance.
<i>internal</i>	CHI	Rate between inter-cluster and intra-cluster dispersion. Higher values suggest dense and well-separated clusters. The number of clusters must lie between 2 and $ \text{data points} - 1$.
<i>internal</i>	DBI	Shows the average similarity of each cluster to the most similar cluster. Clusters further apart from each other produce better results. The lowest possible and best score is 0. The number of clusters must lie between 2 and $ \text{data points} - 1$.

It is remarkable that the methods that perform best have the corresponding premises as additional context information, while the worst performing methods do not utilize them. In fact, for each evaluation measure, all 16 methods that include the premise as context information achieve better values than those 16 that do not include it. The inclusion of a query claim and result claim as context information seems to have no influence on the ranking, because the methods with and without usage of this context information in the ranking are sometimes better and sometimes worse. Thus, clustering EDUs always requires the context information in the premise. Kruskal-Wallis tests with $p = 0.05$ were conducted on the three external and two internal measures of the eleven query claims of the three best performing methods as well as the baseline.⁴ For AMI, CHI and DBI significant differences were found. For purity and Rand, no significant differences could be found between the four groups.

For the internal cluster evaluations all methods that include the premise as context information produce better outcomes than those computed for the baseline $BL_{\text{premiseAsCluster}}$ clusters. The best values were achieved when using EDUs computed with ELMO or BERT embeddings. This observation clearly shows challenges in automatic clustering of arguments in difficult datasets. We conducted Mann-Whitney U tests on the five measures from the eleven clusterings for each of the three best methods and their counterpart without utilization of the premise as context-information (e.g. $ELMO_{e,p,q}$ and $ELMO_{e,q}$ were observed as a pair) with $p = 0.05$.⁵ We found significant differences in values for purity, AMI, Rand, CHI and DBI for ELMO as well as INFERSENT; for the two experiments with BERT embeddings, significant differences were found for all

⁴ Kruskal-Wallis tests were used as except for purity, data is not normally distributed in the four groups; this was tested with Shapiro-Wilk tests.

⁵ Mann-Whitney U tests were used as for all pairs, some of the measures are not normally distributed; this was tested with Shapiro-Wilk tests.

Table 4. The **external** and **internal** clustering evaluation including: mean *purity*, mean *adjusted mutual information (AMI)*, mean *adjusted Rand index (Rand)*, mean *Calinski-Harabasz index (CHI)*, and mean *Davies-Bouldin index (DBI)* for the baselines $BL_{premiseAsCluster}$, $BL_{oneCluster}$, $BL_{ownClusters}$ (see Section 4.2), for the best (marked bold) as well as worst (underlined) performing combinations of context (premise p , result claim r , query claim q) with EDU e and embedding methods for the 11 queries.

Method	External			Internal	
	purity	AMI	Rand	CHI	DBI
$BL_{premiseAsCluster}$	0.6281	0.4863	0.3618	1.337	2.882
$BL_{oneCluster}$	0.2512	0	0	-	-
$BL_{ownClusters}$	1	0	0	-	-
$ELMO_{e,p,q}$	0.6032	0.3453	0.2406	6.4446	2.4161
$INFERSENT_{e,p}$	0.5977	0.3888	0.2837	4.3765	2.5958
$BERT_{e,p,r}$	0.5996	0.3492	0.2496	4.2647	2.3412
$INFERSENT_{e,q}$	<u>0.4309</u>	<u>0.046</u>	<u>0.0255</u>	1.2496	3.1276
$Flair_{e,q}$	<u>0.4315</u>	<u>0.0465</u>	<u>0.023</u>	1.3228	3.1455
$Flair_e$	<u>0.4492</u>	<u>0.0695</u>	<u>0.0477</u>	<u>1.2346</u>	<u>3.158</u>

external measures and CHI. From this observation we derive the usefulness of premises as context information for the overall clustering quality.

Error Analysis of the Clustering We performed an additional manual evaluation of the clustering by including the three best performing methods shown in Table 4, as well as the initial manual clustering. For this evaluation we randomly picked 30 clusters which contain at least three EDUs per cluster (120 clusters in total) and added a new EDU to each of them, which two human annotators (different from the one who constructed the ground truth in Section 3) had to spot to determine the perceived soundness of the clustering. This new EDU originated from the same premise or, if no EDU was available there, from the same query. For each cluster at most five EDUs were shown. They were shuffled and the new EDU was placed at a random position. Additionally, we include a random baseline. Here, for each of the 120 evaluation clusters, the intruding EDU was picked at random.

Only the query and the EDUs were presented to the annotators. For the manually labeled clusters, both annotators managed to identify 16 out of 30 false EDUs. For $INFERSENT_{e,p}$, $BERT_{e,p,r}$, and $ELMO_{e,p,q}$, it was 11.5, 8.5, and 7 out of 30 on average, respectively.⁶ The inter-annotator agreement, calculated with Krippendorff’s α [13], was 0.463 on a nominal scale, implying that the agreement is moderate. The random baseline picked a total of 9, 4, and 6 wrong EDUs. We found no significant differences with Kruskal-Wallis tests for clustering based on BERT, Elmo and InferSent embeddings for the number of correctly identified intruding EDUs by the two annotators and the random baseline. Yet, for the ground truth, significant differences were found. The results show that

⁶ The differences in the annotations were two times 1, once 0, and once 4.

the automatic clustering of EDUs by semantics still lags behind manual annotation. However, they also reveal that even the manually produced clustering is ambiguous, as one would have expected to find (almost) all the wrong EDUs. Overall, the annotators' impression was that it was a very difficult task to spot the intruding EDU because except for the query no context information was given. In most cases, the query did not really help in identifying the out-of-place EDU. In contrast, when creating the ground truth, the (other) annotator first read the whole texts associated with result claims and then decided which EDUs should be clustered. This is an important difference.

5 Conclusion and Future Work

Segmenting complex premises and clustering of semantically similar premises are important tasks in the retrieval of arguments, as argument retrieval systems need to deal with complex natural-language statements and should not show duplicate results. This is even a problem for arguments extracted from debate portals since single textual premises often address a variety of aspects. In this paper we discussed the segmentation of premises into EDUs, as well as clustering these from an uncurated dataset. Our results show that segmenting premises into their EDUs in such a dataset with rule-based procedures that are suitable for curated datasets is feasible, in particular by following either a precision or a recall-oriented approach. Furthermore, we have seen that clustering EDUs only performs comparably well with the associated premises as context information at least. The segmentation of EDUs from noisy texts remains a difficult task for now. We provide the labeled data of EDUs and clusters of EDUs so that future argument mining methods can use it for evaluation of their performance.

Future work will include extracting unique EDUs using context information and further analyzing properties of real-world datasets which impede manual EDU extraction and clustering. With these insights, an annotation support system could be constructed to help manually identifying and clustering EDUs.

Acknowledgments We would like to thank Anna-Katharina Ludwig for her invaluable help in clustering the EDUs and Patrick J. Neumann for his help in the implementation.

This work has been funded by the Deutsche Forschungsgemeinschaft (DFG) within the project ReCAP, Grant Number 375342983 - 2018-2020, as part of the Priority Program "Robust Argumentation Machines (RATIO)" (SPP-1999).

References

1. Afantenos, S.D., Denis, P., Muller, P., Danlos, L.: Learning recursive segments for discourse parsing. In: LREC (2010)
2. Akbik, A., Blythe, D., Vollgraf, R.: Contextual string embeddings for sequence labeling. In: COLING (2018)
3. Boltuzic, F., Snajder, J.: Identifying prominent arguments in online debates using semantic textual similarity. In: ArgMining@HLT-NAACL (2015)

4. Cabrio, E., Villata, S.: Five years of argument mining: a data-driven analysis. In: IJCAI (2018)
5. Carlson, L., Marcu, D.: Discourse Tagging Reference Manual, <https://www.isi.edu/~marcu/discourse/tagging-ref-manual.pdf>
6. Carreras, X., Màrquez, L.: Boosting trees for clause splitting. In: ACL (2001)
7. Conneau, A., Kiela, D., Schwenk, H., Barrault, L., Bordes, A.: Supervised learning of universal sentence representations from natural language inference data. In: EMNLP (2017)
8. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT (2019)
9. Dumani, L., Neumann, P.J., Schenkel, R.: A framework for argument retrieval - ranking argument clusters by frequency and specificity. In: ECIR. LNCS, vol. 12035. Springer (2020)
10. Dumani, L., Schenkel, R.: A systematic comparison of methods for finding good premises for claims. In: SIGIR (2019)
11. van Eemeren, F.H., Garssen, B., Krabbe, E.C.W., Henkemans, A.F.S., Verheij, B., Wagemans, J.H.M. (eds.): Handbook of Argumentation Theory. Springer (2014)
12. Janier, M., Lawrence, J., Reed, C.: OVA+: an argument analysis interface. In: COMMA (2014)
13. Krippendorff, K.: Estimating the reliability, systematic error and random error of interval data. *Educational and Psychological Measurement* **30** (1970)
14. Langfelder, P., Zhang, B., Horvath, S.: Defining clusters from a hierarchical cluster tree: the Dynamic Tree Cut package for R. *Bioinformatics* **24**(5) (11 2007)
15. Mann, W.C., Thompson, S.A.: Rhetorical structure theory: Toward a functional theory of text organization. *Text & Talk* **8**(3) (1988)
16. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: NeurIPS (2013)
17. Peldszus, A., Stede, M.: From Argument Diagrams to Argumentation Mining in Texts. *IJCINI* **7**(1) (2013)
18. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. In: NAACL-HLT (2018)
19. Reimers, N., Schiller, B., Beck, T., Daxenberger, J., Stab, C., Gurevych, I.: Classification and clustering of arguments with contextualized word embeddings. In: ACL (2019)
20. Rendón, E., Abundez, I., Arizmendi, A., Quiroz, E.M.: Internal versus external cluster validation indexes. *Int. J. Comput. Commun.* **5**(1) (2011)
21. Soricut, R., Marcu, D.: Sentence level discourse parsing using syntactic and lexical information. In: HLT-NAACL (2003)
22. Stab, C., Daxenberger, J., Stahlhut, C., Miller, T., Schiller, B., Tauchmann, C., Eger, S., Gurevych, I.: Argumenttext: Searching for arguments in heterogeneous sources. In: NAACL-HLT (2018)
23. Stab, C., Gurevych, I.: Identifying argumentative discourse structures in persuasive essays. In: EMNLP (2014)
24. Stede, M., Afantenos, S.D., Peldszus, A., Asher, N., Perret, J.: Parallel discourse annotations on a corpus of short texts. In: LREC (2016)
25. Tofiloski, M., Brooke, J., Taboada, M.: A syntactic and lexical-based discourse segmenter. In: ACL and AFNLP (2009)
26. Trautmann, D., Daxenberger, J., Stab, C., Schütze, H., Gurevych, I.: Fine-grained argument unit recognition and classification. In: AAAI (2020)
27. Wachsmuth, H., Potthast, M., Khatib, K.A., Ajjour, Y., Puschmann, J., Qu, J., Dorsch, J., Morari, V., Bevendorff, J., Stein, B.: Building an argument search engine for the web. In: ArgMining@EMNLP (2017)

Combining Universal Adversarial Perturbations

★

Maurus Kühne¹[0000-0002-4205-3552] and Beat Tödli²[0000-0003-3674-2340]

¹ Fernfachhochschule Schweiz
maurus.kuehne@students.ffhs.ch

² Institut für Informations- und Prozessmanagement, FHS St. Gallen
beat.toedtli@ost.ch

Abstract Universal adversarial perturbations (UAPs) are small perturbations imposed on images that are able to fool a single convolutional neural network image classifier. They have been shown to generalise well to other neural networks. Here, we report on our reproduction effort of the results given in a work by Moosavi-Dezfooli et al. on UAPs and study two methods to construct UAPs for several neural networks. While the results are not strong enough to make general conclusions, they suggest that UAPs indeed profit from being constructed on several neural networks. Also, we show that a linear interpolation between two UAPs does not produce a viable UAP on both networks.

Keywords: Adversarial Training, Universal Adversarial Perturbation

1 Introduction

The discovery of Szegedy et al. [12] that several machine learning models including deep neural networks are vulnerable to *adversarial attacks* was seminal for a new subfield of studying deep learning. Probably the most intriguing, but also unsettling result was that adversarial examples can be made quite imperceptible to the human eye while still fooling a convolutional neural network to misclassify the image [3]. Subsequent work has developed various algorithms in a variety of white-box, grey-box and black-box attack scenarios as well as defensive strategies such as adversarial training [11]. Moosavi-Dezfooli et al. [5,6] have demonstrated that *universal* perturbations exist, i.e. that a single set of pixel modifications can be found that fools a network on a large fraction of the training data set. Moreover, universal adversarial perturbations (UAPs) also fool other convolutional networks. The authors of [5,6] show good generalization results for UAPs generated with their procedure *DeepFool* [6] across different deep learning architectures.

These results suggest that neural networks and convolutional neural networks for image classification in particular partly share a common structure that can be

* Copyright © 2020 by the paper's authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

exploited by universal adversarial perturbations (UAPs) while yet other aspects are different. Thus, understanding UAPs provides a window into the weaknesses of neural networks. By building UAPs that are viable on several networks, a common weakness of those networks is identified. This in turn helps in building more robust networks, for example again by adversarial training. Even more generally, therefore, we view work on UAPs as one more way of (partly) approaching the bigger question of why neural networks with a large number of weights work at all (or when they don't).

In this context, we ask whether combining two neural networks in generating adversarial perturbations can improve the transferability of UAPs to new convolutional neural networks. We observe that in the approach by Moosavi-Dezfooli et al. [5,6], implementing such a combination is particularly simple. We report on our effort to reproduce their results, provide our code and investigate whether modifications of their UAP algorithm are able to improve the generalisation capability of UAPs. To do so, we devise modifications of the UAP generation algorithm that take into account several networks at the same time³. Specifically, we assess whether incorporating information from a second neural network architecture improves the fooling rate of UAPs on a third neural network. We investigate three combination procedures and compare them with the original adversarial attack procedure.

Given the practical potential and relevance of Deep Learning and the potential security threats of adversarial perturbations, finding a robust resolution is important and urgent. However, research into the topic is hampered by the *reproducibility crisis* in machine learning [10]. Research results are often difficult to reproduce due to undocumented values for hyperparameters, software library versions etc. [4]. As this was also the case for our reproduction efforts of the UAP-results of Moosavi-Dezfooli et. al., we endorse the NeurIPS-2019 code submission policy by providing our code and including their reproducibility checklist.

This paper is organised as follows. In Sec. 2 we briefly review related work. In Sec. 3 we present the basic methods used to generate adversarial and universal adversarial perturbations as introduced by Moosavi-Dezfooli et al. [5]. We then present two modifications of their UAP generation algorithm to combine UAPs for several networks and investigate linear interpolations between UAPs. In Sec. 4 we first describe our reproduction effort of the original work of Moosavi-Dezfooli et al. [5] to produce UAPs on a set of networks and provide our fooling rates. We then present our results on the three methods to produce UAPs for several networks and show that the transferability to a third network is improved. In Sec. 5 we discuss these results and conclude with Sec. 6.

2 Related Work

Many authors have suggested adversarial perturbation generating methods in various different settings. These attempts often generate per-instance perturb-

³ Instead of combining UAPs from different networks, UAPs of networks trained on different data domains can be combined, as done e.g. by Naseer et al. [9].

ations. Among the image-agnostic methods, i.e. those generating UAPs, there are both data-driven and data-independent techniques, white-box and black-box attacks (depending on whether the internal structure of the network to be attacked is accessible to the attacker) and whether the attack is targeted or not (i.e. whether a misclassification into a particular class is required, or whether any misclassification is counted as a success). For a review and references, see [1]. UAPs can be generated e.g. by gradient descent on a loss function or learned using generative models. These methods being data-driven, they require access to training data, preferably the training data of the network to be fooled. Data-independent techniques such as Fast Feature Fool [8] or GD-UAP [7] do not need this access, but usually have access to the internal state of the network to be fooled. Our approach is a data-dependent white-box attack closely tied to the UAPs of Moosavi-Dezfooli et al. [5].

Many of these methods generate perturbations that transfer well between different model architectures. As part of our contribution we seek methods that optimise the transferability of perturbations between models. Our approach tries to achieve this by using multiple models to generate perturbations.

3 Methods

3.1 General Setting

Given an image classifier $\hat{k}(\mathbf{x}) = \text{sgn}(f(\mathbf{x}))$ that is based on the sign of a classification function $f : \mathbf{R}^n \rightarrow \mathbf{R}$, adversarial attacks seek a perturbation \mathbf{v} such that

$$\hat{k}(\mathbf{x} + \mathbf{v}) = \text{sgn}(f(\mathbf{x} + \mathbf{v})) \neq \text{sgn}(f(\mathbf{x})) = \hat{k}(\mathbf{x}).$$

In the following we briefly describe the adversarial perturbation generating method DeepFool and its generalisations to UAPs in the multi-class classification setting. We then describe the two approaches analysed here to build UAPs from several networks.

3.2 DeepFool

In the generation procedure DeepFool [6], the perturbation \mathbf{v} for an image \mathbf{x} is defined to be the shortest vector⁴ (using the L_p -norm $\|\cdot\|_p$) such that $\mathbf{x} + \mathbf{v}$ lies on a decision boundary. If $f(\mathbf{x})$ is a linear function $f(\mathbf{x}) = \mathbf{w}\mathbf{x} + \mathbf{b}$, then \mathbf{v} can be shown to be $\mathbf{v} = -\frac{f(\mathbf{x})}{\|\mathbf{w}\|_p} \mathbf{w}$. As f is nonlinear in general, the Taylor approximation of the function f around \mathbf{x} , $f(\mathbf{x} + \mathbf{v}) = f(\mathbf{x}) + (\nabla f(\mathbf{x}))^T \mathbf{v}$ is used to iteratively reach the decision boundary. In the multi-class setting considered here, an additional step is required that identifies the closest decision boundary.

⁴ In practice and also in our code, the vectors (elements of a vector space) such as \mathbf{x} and \mathbf{v} have additional structure such as (for images) height, width and depth that is used to implement the classifier \hat{k} . In the field of Deep Learning these data structures are therefore often called *tensors*.

3.3 Universal Adversarial Perturbations with DeepFool

Perturbations for each image in a dataset X (such as those generated using DeepFool) can be combined to form universal adversarial perturbations for a single network [5]. The procedure is given in Alg. 1 for a classifier set K with one element. Essentially, DeepFool perturbations of images that are not yet misclassified are added to obtain a universal perturbation. Whenever the norm of the perturbation becomes large, a rescaling is applied. The perturbation is scaled back to satisfy a norm bound given by $\|\mathbf{v}\|_p \leq \xi$ (that potentially undoes the successful perturbation of some images). For a small value of ξ , this ensures that the perturbation remains largely invisible.

Intriguingly, although the directions to the class boundaries vary for different training images, the resulting average over all image perturbations works well according to [5], even for other convolutional neural networks whose class boundaries might be expected to look rather different.

3.4 Multi-Classifier Universal Adversarial Perturbations

In the following subsections, we detail two approaches to generating UAPs for several classifiers.

Alternated Generation of Perturbations Since UAPs are constructed by adding up the perturbations generated using DeepFool, there is a natural way to combine perturbations generated by the two networks: We add up the contributions from all networks. We note that adding up the perturbations is not a commutative operation, since projections take place once the size (norm) of the perturbation becomes too large. The precise procedure used here is given in Alg. 1.

Variants of Alg. 1 exist that sample the images differently. One might for example generate perturbation by alternating the classifier for each image. We show this variant in Alg. 2. We evaluate both variants and compare their performance in Tab. 2.

Interpolation Between UAPs on Individual Networks A simple yet instructive alternative to the above rather involved perturbation construction is given by a simple weighted average of the perturbation vectors generated on the individual networks. If we restrict our attention to the combination of two neural networks for now, then the weighted averages of the perturbations lie on a line in the high-dimensional vector space of images. In the following, we specifically investigate whether any perturbation lying on this line improves on the fooling capability of the two endpoints with respect to a third network. More formally, given UAPs \mathbf{v}_f and \mathbf{v}_g of two neural networks f and g , we consider

$$\mathbf{v}_{fg}(\lambda) = \lambda \mathbf{v}_f + (1 - \lambda) \mathbf{v}_g \tag{1}$$

for values $\lambda \in [0, 1]$. We seek the value of λ that maximises the average fooling rate over f , g and a third network, given the current training data set.

Outlook to Other Approaches We have also investigated other, more involved approaches that led to unsatisfactory results. In particular, we were interested in constructing a multi-class DeepFool procedure that uses the gradients of two networks towards the next class boundary to compute a perturbation of a single image. One might try to find perturbations towards class boundaries that are aligned as much as possible, but where the class boundaries correspond to different classes in different networks. As of now, these attempts have not provided efficient UAPs for multiple networks.

```

1 Input:Data set  $X$ , set of classifiers  $K$ , desired norm  $\|\cdot\|_p$  of the perturbation  $\xi$ 
2 Output: Universal perturbation vector  $\mathbf{v}$ 
3 Initialise  $\mathbf{v} \leftarrow 0$ 
4 while Average fooling rate is too low and max. number of iterations is not
   reached do
5   foreach image  $\mathbf{x} \in X$  do
6     foreach  $\hat{k} \in K$  do
7       if  $\hat{k}(\mathbf{x}) = \hat{k}(\mathbf{x} + \mathbf{v})$  then
8          $\Delta\mathbf{v} \leftarrow \text{DeepFool}(\mathbf{x} + \mathbf{v}, \hat{k})$ 
9          $\mathbf{v} \leftarrow \mathbf{v} + \Delta\mathbf{v}$ 
10         $\mathbf{v} \leftarrow \sqrt{\xi} \mathbf{v} / \|\mathbf{v}\|_p$ 
11      end
12    end
13  end
14  Shuffle  $X$ 
15 end
16 return  $\mathbf{v}$ 

```

Algorithm 1: Computation of universal adversarial perturbations for multiple neural networks. The function `DeepFool` computes an adversarial perturbation as described in Sec. 3.2. For each image, perturbation updates are computed for all classifiers, added up and rescaled to have norm ξ . Note that the method is identical to the single classifier UAP method in [5] when the set K consists of a single classifier.

4 Results

In this section we first discuss our attempt at the reproduction of the results in [5] regarding the transferability of UAPs across neural network architectures. We then provide the results of our approaches to construct UAPs based on two neural network architectures at the same time.


```

1 Input: Data set  $X$ , ordered set of classifiers  $K$ , desired norm  $\|\cdot\|_p$  of the
   perturbation  $\xi$ 
2 Output: Universal perturbation vector  $\mathbf{v}$ 
3 Initialise  $\mathbf{v} \leftarrow 0$ 
4 while Average fooling rate is too low and maximum number of iterations is
   not reached do
5   foreach image  $\mathbf{x} \in X$  do
6      $\hat{k} \leftarrow$  first element of  $K$ 
7      $K \leftarrow$  cyclic rotation of  $K$ 
8     if  $\hat{k}(\mathbf{x}) = \hat{k}(\mathbf{x} + \mathbf{v})$  then
9        $\Delta\mathbf{v} \leftarrow \text{DeepFool}(\mathbf{x} + \mathbf{v}, \hat{k})$ 
10       $\mathbf{v} \leftarrow \mathbf{v} + \Delta\mathbf{v}$ 
11       $\mathbf{v} \leftarrow \sqrt{\xi}\mathbf{v} / \|\mathbf{v}\|_p$ 
12    end
13  end
14  Shuffle  $X$ 
15 end
16 return  $\mathbf{v}$ 

```

Algorithm 2: Computation of universal adversarial perturbations for multiple neural networks. For each image, perturbation updates are computed for the next classifier in the classifier sequence. Then the sequence is cyclically rotated.

4.1 Reproduction of the Original Results on Universal Adversarial Perturbations

Moosavi-Dezfooli et al. tested the DeepFool and Universal Adversarial Perturbations algorithms on 5 different neural networks [5]. We choose the same networks as the ones used in [5] and use publicly available pretrained weights for all models, since the weights used in [5] were not specified. We compare the achieved fooling rates with those given in the original paper. Separate experiments have been performed by training individual UAPs on each of the networks and subsequently measuring the fooling rates on all available networks. All perturbations were generated using the same random subset of 10'000 images of the ImageNet training set [2], and the fooling rates were measured on the ImageNet validation set (containing 50'000 images).

Fig. 1 shows the perturbations generated. In their general structure and appearance, they are similar to the ones reported in [5]. In particular, the fine line-shaped structures in green and magenta are quite recognisable and are present also in the original results in [5]. We believe that this indicates that deviations from the original results reported below stem from configuration details rather than a fundamental reproduction mistake. Therefore and despite the lower fooling rates reported below, we feel justified to use this setup to study our methods to generate a combined UAP for several networks.

Tab. 1 shows the achieved fooling rates for the tested models (left values, in boldface) and the fooling rates reported by Moosavi-Dezfooli et al. [5] (right

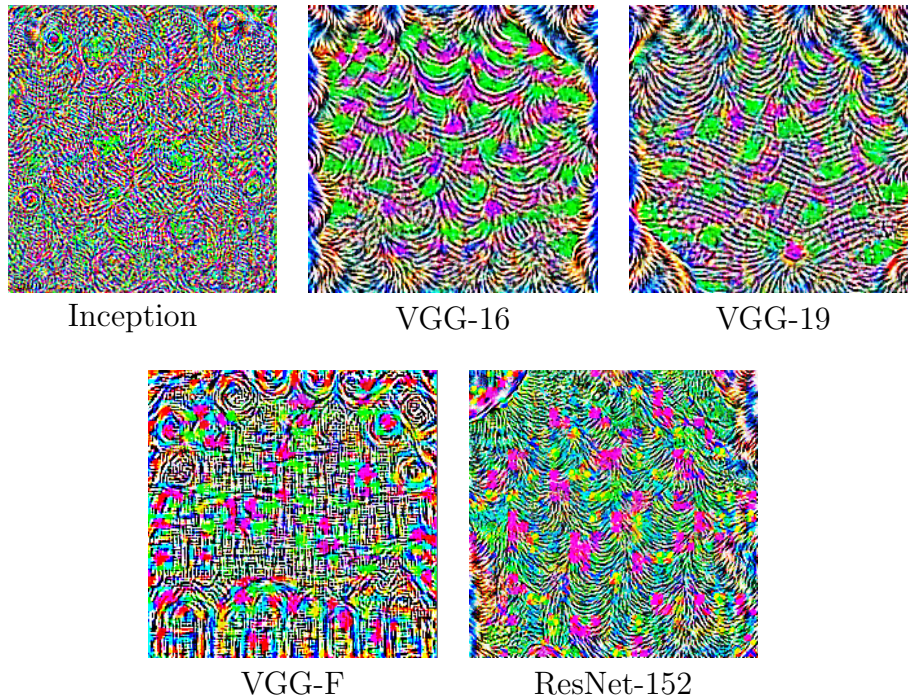


Figure 1. Visualisation of the generated perturbations. To visualise the perturbations, we shifted them by $+\xi$ and extended them to the entire colour space with a scalar multiplication.

values). The first column indicates the network used to generate the UAP while the first row gives the network on which the fooling rate is measured. The main diagonal therefore contains the self-fooling rates, i.e. the fooling rates achieved using the same model for generating the perturbation and measuring the fooling rate.

To reproduce these numbers, several insufficiently documented design choices had to be researched. For example, the original results are not stated for a given epoch but using a stopping condition on the error rate. Values reported here are for epoch 20, at which point the mean value over the last 5 epochs typically varies by less than 0.005. Parameter values are $p = \infty$ and $\xi = 10$. We used a maximum of 10 DeepFool update iterations. The overshoot parameter was $\eta = 0.02$. These parameter values were taken from the work of Moosavi-Dezfooli et al. [5,6].

Another important parameter is the number of tested class boundaries⁵, called `num_classes`. It gives the number of class boundaries in whose direction a perturbation is searched for. Computation time is highly sensitive to this para-

⁵ As given in the code at <https://github.com/LTS4/universal/>

Table 1. Fooling rates using the UAP method on the ImageNet validation set for several neural networks. Left values in bold are our reproduction results. The values to the right are the results reported in [5].

	VGG-F		Inception		VGG-16		VGG-19		ResNet-152	
VGG-F	90%	94%	56%	48%	32%	42%	32%	42%	24%	47%
Inception	42%	46%	82%	79%	16%	39%	16%	40%	16%	46%
VGG-16	46%	63%	60%	57%	59%	78%	52%	73%	38%	63%
VGG-19	45%	64%	58%	54%	51%	74%	54%	78%	33%	58%
ResNet-152	42%	46%	55%	51%	31%	47%	33%	46%	73%	84%

meter, and checking all 1000 training boundaries of ImageNet was infeasible. Higher values generally improve the fooling performance, though exceptions are observed. Other minor parameters such as random initialisation values for the train-test split, etc. were taken from the provided code. We have tried to optimise these parameters using grid searches, but the available computing resources have limited these efforts.

As part of this contribution we provide our code for the above reproduction effort and include the NeurIPS-2019 reproducibility checklist⁶.

4.2 Evaluation of Approaches to Construct Universal Adversarial Perturbations for Multiple Neural Networks

In this section we present our results for constructing UAPs that make use of two neural networks. We discuss results for the two approaches presented in Sec. 3.4, the alternating generation of perturbations and the linear interpolation between UAP on individual networks.

The alternating generation of perturbations as given in Alg. 1 has been applied to the set of classifiers $K = \{\text{Inception, VGG-16}\}$. In Tab. 2 the fooling rates are reported for these two networks and for ResNet-152. The table has two sections. In the upper section, the network listed in the first column is used to generate the perturbation. The columns give the fooling rates on the network given in the column title as well as their average as our measure for the generalisability of the UAPs. We include the original network in this average. The second section reports results by applying the alternating generation of UAPs and the interpolation method given in Eq. (1) using the value $\lambda = 0.05$.

Alg. 1 achieved results similar to the ones of a perturbation generated on VGG-16 only. Alg. 2 achieved slightly higher fooling rates on VGG-16 and ResNet-152 than a perturbation generated directly on VGG-16 (with an absolute increase of 2 and 3 percentage points, respectively). For Inception, the perturbation achieved a fooling rate of 67%. This is 15% below the measured self-fooling rate of Inception (82%) but 7% higher than the fooling rate achieved with a VGG-16 model.

⁶ See the file checklist.md in <https://github.com/mauruskuehne/lwda-paper>

For a linear interpolation between the UAPs of VGG-16 and Inception V1, the best results were achieved for $\lambda = 0.05$ (see Eq. (1)). Using this configuration, the fooling rates remained essentially unchanged compared to the UAP generated on VGG-16 only. The small value $\lambda = 0.05$ results in a perturbation that is similar to the VGG-16 perturbation as the VGG-16 perturbation is weighted with $1 - \lambda = 0.95$ while the Inception perturbation contributes only with a weight of 5%. Choosing $\lambda \in [0.1, 0.15, \dots, 0.4]$ resulted in perturbations with lower fooling rates for both models with respect to the rates achieved by separately training UAPs on the two networks. For $\lambda \in [0.4, 1.0]$, the fooling rates for Inception improved again but did not exceed the fooling rate of a UAP generated on Inception itself. The fooling rate on VGG-16 continued to deteriorate, stabilising at a low fooling rate of $\sim 15\%$ after $\lambda \geq 0.65$. This suggests that linear interpolation does not result in improved fooling rates.

Table 2. Comparison of the fooling rates by UAPs trained on individual networks (upper three rows) and of combination methods for several networks (lower three rows). Best values are shown in bold. Among the alternating generation variants (Alg. 1 and Alg. 2) and the linear interpolation procedure (Eq. (1)), Alg. 2 performs best with respect to the average fooling rate over all three networks. Results are reported on the validation set, using the l_∞ -norm, $\xi = 10$ and 20 UAP iterations.

	Inception V1	VGG-16	ResNet-152	Average
Inception V1 UAP	82%	16%	16%	38%
VGG-16 UAP	60%	59%	38%	52%
ResNet-152 UAP	55%	31%	73%	53%
linear interpolation with $\lambda = 0.05$	60%	59%	38%	52%
alternating generation of UAPs, Alg. 1	63%	55%	36%	51%
alternating generation of UAPs, Alg. 2	67%	61%	41%	56%

5 Discussion

5.1 Reproduction of the Original Results on Universal Adversarial Perturbations

For most models the fooling rates reported in the original paper could not be achieved, indicating that our reproduction results fell short of being satisfactory. For VGG-F, VGG-16, VGG-19 and ResNet-152 our self-fooling rates were between 4 and 24 absolute percentage points lower. For Inception we achieved a self-fooling rate 3 absolute percentage points higher than the one reported in the original paper. Further research is needed to state the precise conditions under which a reliable reproduction of the reported fooling rates is possible. As a step in this direction we have provided our code.

The non-diagonal values in Tab. 1 are large but typically significantly smaller than the diagonal values. They show a degree of transferability of UAPs

generated with DeepFool to other models. Therefore, despite the reproducibility problems, these results broadly confirm that UAPs generated with DeepFool generalise to other network architectures. Nevertheless, it is clear that some aspects of UAPs are specific to a given neural network architecture. We discuss our results on finding a way to improve the non-diagonal elements (potentially at the cost of the diagonal ones) in the next section. Interestingly, we achieved lower fooling rates than Moosavi-Dezfooli et al., except for the Inception network, for which we achieve 3 to 8 absolute percentage points higher fooling rates. This difference may be due to different stopping criteria, resulting in Moosavi-Dezfooli et al. running less optimisation epochs. Another possibility is that the chosen hyperparameter values for DeepFool and UAP might be particularly well suited or optimised for the Inception model. This in turn would explain the lower fooling rates achieved on other models.

5.2 Alternating Generation of Perturbations and Linear Interpolation Between UAPs on Individual Networks

As the results in Sec. 4.2 clearly show a linear interpolation between two UAPs does not give good results. This suggests that using a weighted average to combine UAPs is not a suitable approach to produce good UAPs for several neural networks. A more sophisticated UAP-combination procedure is clearly necessary to generate perturbations that fool both networks to a high degree.

The results given by the alternating generation of perturbations (Alg. 2) are much better (see Tab. 2). The fooling rate of the perturbation generated jointly on Inception and VGG-16 is better than the ones generated on any one of the two networks. A perturbation generated on Inception achieves a fooling rate of 16% on VGG-16 while a perturbation generated on VGG-16 achieves a fooling rate of 60% on Inception. Both rates are lower than the ones of a perturbation generated jointly on Inception and VGG-16, achieving 67% on Inception and 61% on VGG-16. Furthermore, the fooling rate of a jointly trained UAP on Inception and VGG-16 on a third network (Resnet-152) is better than the fooling rate of both single-network UAPs. The UAP generated jointly on both networks even worked slightly better for VGG-16 than the one trained on VGG-16 alone. The reason for this effect and its statistical significance are not yet established.

In judging these results, we note that estimates of their uncertainties are still lacking due to constraints on our computational resources. Ideally, one would like to provide mean and standard deviation values of all these numbers over multiple train-test splits as encouraged by the NeurIPS-2019 reproducibility checklist. Furthermore, the choice of the ImageNet training and testing data and the hyperparameter values (such as p , ξ , `num_classes`, etc.) should be investigated.

6 Conclusions

The results reported here on generalising UAPs across several networks clearly have to be interpreted cautiously given the fact that even the reproduction of

previously reported results has not been satisfactory. Establishing reproducibility standards for machine learning publications remains a crucial challenge that is hampering progress.

With the above caution in mind, the results reported here suggest that finding universal adversarial perturbations that generalise across different convolutional neural networks is not a hopeless endeavour. As we found, such a UAP is likely not a linear combination of UAPs of different networks but must be constructed in a more subtle way. Our best approach, Alg. 2, most certainly is not optimal. Nevertheless, it already shows some promising results: The generalisability of the fooling rates to ResNet-152 is enhanced by combining the UAPs of two networks, with respect to the UAPs generated on either one of the Inception or VGG-16 network. This suggests that combining several or even many networks might produce UAPs that are efficient on a whole class of trained convolutional neural networks.

References

1. Chaubey, A., Agrawal, N., Barnwal, K., Guliani, K.K., Mehta, P.: Universal adversarial perturbations: A survey. ArXiv (2020), <http://arxiv.org/abs/2005.08087>
2. Deng, J., Dong, W., Socher, R., Li, L., Li, K., Li, F.: ImageNet: A large-scale hierarchical image database. In: CVPR09. pp. 248–255. IEEE Computer Society (2009). <https://doi.org/10.1109/CVPR.2009.5206848>
3. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015), <http://arxiv.org/abs/1412.6572>
4. Gundersen, O.E., Kjensmo, S.: State of the art: Reproducibility in artificial intelligence. In: McIlraith, S.A., Weinberger, K.Q. (eds.) AAAI. pp. 1644–1651. AAAI Press (2018), <http://dblp.uni-trier.de/db/conf/aaai/aaai2018.html#GundersenK18>
5. Moosavi-Dezfooli, S.M., Fawzi, A., Fawzi, O., Frossard, P.: Universal Adversarial Perturbations. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 86–94 (Juli 2017). <https://doi.org/10.1109/CVPR.2017.17>, ISSN: 1063-6919
6. Moosavi-Dezfooli, S.M., Fawzi, A., Frossard, P.: DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2574–2582 (Juni 2016). <https://doi.org/10.1109/CVPR.2016.282>, ISSN: 1063-6919
7. Mopuri, K.R., Ganeshan, A., Babu, R.V.: Generalizable data-free objective for crafting universal adversarial perturbations. CoRR (2018), <http://arxiv.org/abs/1801.08092>
8. Mopuri, K.R., Garg, U., Babu, R.V.: Fast feature fool: A data independent approach to universal adversarial perturbations. CoRR (2017), <http://arxiv.org/abs/1707.05572>
9. Naseer, M.M., Khan, S.H., Khan, M.H., Shahbaz Khan, F., Porikli, F.: Cross-domain transferability of adversarial perturbations. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in

- Neural Information Processing Systems 32, pp. 12905–12915. Curran Associates, Inc. (2019), <http://papers.nips.cc/paper/9450-cross-domain-transferability-of-adversarial-perturbations.pdf>
10. Raff, E.: A step toward quantifying independently reproducible machine learning research. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32, pp. 5485–5495. Curran Associates, Inc. (2019), <http://papers.nips.cc/paper/8787-a-step-toward-quantifying-independently-reproducible-machine-learning-research.pdf>
 11. Ren, K., Zheng, T., Qin, Z., Liu, X.: Adversarial attacks and defenses in deep learning. *Engineering* **6**(3), 346 – 360 (2020). <https://doi.org/10.1016/j.eng.2019.12.012>
 12. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. In: International Conference on Learning Representations. ICLR (2014), URL <http://arxiv.org/abs/1312.6199>

Phantom Embeddings: Using Embedding Space for Model Regularization in Deep Neural Networks

Mofassir ul Islam Arif¹, Mohsan Jameel¹, Josif Grabocka², and Lars Schmidt-Thieme¹

¹ Information Systems and Machine Learning Lab, University of Hildesheim, Germany

{mofassir,mohsan.jameel,schmidt-thieme}@ismll.uni-hildesheim.de

² Department for Computer Science, Albert-Ludwigs-University, Freiburg, Germany
{grabocka}@informatik.uni-freiburg.de

Abstract. The strength of machine learning models stems from their ability to learn complex function approximations from data; however, this strength also makes training deep neural networks challenging. Notably, the complex models tend to memorize the training data, which results in poor regularization performance on test data. The regularization techniques such as L1, L2, dropout, etc. are proposed to reduce the overfitting effect; however, they bring in additional hyperparameters tuning complexity. These methods also fall short when the inter-class similarity is high due to the underlying data distribution, leading to a less accurate model.

In this paper, we present a novel approach to regularize the models by leveraging the information-rich latent embeddings and their high intra-class correlation. We create phantom embeddings from a subset of homogenous samples and use these phantom embeddings to decrease the inter-class similarity of instances in their latent embedding space. The resulting models generalize better as a combination of their embedding, regularizes them without requiring an expensive hyperparameter search. We evaluate our method on two popular and challenging image classification datasets (CIFAR and FashionMNIST) and show how our approach outperforms the standard baselines while displaying better training behavior.

Keywords: Deep Neural Networks · Regularization · Embedding Space.

1 Introduction

The field of computer vision has seen a remarkable increase in capability and complexity in recent years. The use of deep learning models in image classification [10] and object detection [4] tasks have shown a marked increase in their

Copyright © 2020 by the papers authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

ability to capture more complex scenarios. Increasingly complex deep learning models such as ResNet [7] and Inception [21] were able to capture more in-depth information from input data. The strength of these deep learning models comes from their ability to take complex data and reduce it to highly expressive latent representations. These latent representations encode an image’s spatial information into a vector through repeated convolutions and pooling operations.

Training these complex models bring their challenges. Generally, the true distribution of the data is unknown, and observations are available in a limited number. These models are trained by iteratively minimizing the empirical risk over the training data (also known as Empirical Risk minimization ERM [22]). However, the increasing complexity of the model tends to overfit the data and generalize poorly on the test data, despite using the proper regularization. The theoretical understanding of ERM guarantees convergence as long as the model complexity does not increase with the number of training data [23]. For deep neural networks, an obvious issue arises as the increase in model complexity is not always complemented by an increase in the training data.

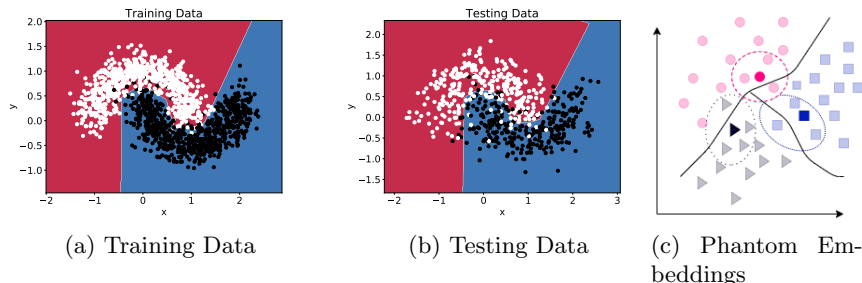


Fig. 1. Fig. 1a shows the overfitted decision boundary on training data. When evaluated on the test set in Fig. 1b the model shows poor generalization, a consequence of overfitting. In Fig. 1c, we show a hypothetical embedding space and the decision boundary created by a deep neural network. The light colors represent the original embeddings while the darker colors represent the phantom embeddings proposed by our method.

To illustrate the aforementioned problems, we train a feed-forward neural network on a synthetic binary classification dataset and visualize the decision boundary in Fig. 1. Fig. 1a shows that the model was able to learn a reasonable decision boundary on the training data. However, due to the limited training examples available to train a complex model, it could not capture a better generalizable decision boundary resulting in poor performance on the test examples, as shown in Fig. 1b. This example showcases two crucial challenges, firstly, how easy it is to overfit and perform poorly on test data. Secondly, in Fig.1a it can be seen that certain instances from differing classes are very close to each other, and ERM fails to provide a procedure to capture those instances.

The model overfitting is treated by introducing the regularization [5, 11, 19, 8] in the ERM objective. However, ERM’s problem is most evident around the vicinity of the boundary region, as samples from different classes are in close proximity. Model complexity could be increased to capture these instances, but that violates the convergence guarantee of ERM since the number of instances does not increase with the increase in model complexity. One can mitigate the ERM failure through the Vicinal Risk Minimization Principle [1] by adding a better regularization using data augmentation [16]. Data augmentation mutates the input instances, traditionally through rotating, flipping, and scaling to inject noise in the training data, thereby preventing the model from memorizing it. However, it is limited as it mutates the data within one class vicinity and not across other classes. Other regularization methods involve tunable hyperparameters requiring an expensive configuration search, and the resulting hyperparameters are non-transferable and dataset-specific.

In this paper, we propose a solution for problems stated above by leveraging the latent embeddings to create what we call a ‘phantom embedding’. This is done by aggregating the latent embeddings of a subset of the instances from the same class. Using the latent vicinal embedding space allows us to use the information-rich embeddings to inject a hyper-parameter free latent vicinal regularization and boost accuracy. Machine learning models transform input data into their representative embeddings: $\psi : \mathbb{R}^M \rightarrow \mathbb{R}^D$ where M is the original data dimensionality and D is the size of the embedding space. Therefore, by creating this phantom embedding, we create phantom data points to learn on. This is illustrated in in Fig. 1c. This phantom embedding is used to ‘pull’ the original instance away from the decision boundary and closer to the samples (of the same class) in the embedding space. For the instances already sufficiently away from the decision boundary the ‘pull’ does not adversely impact since the embedding space is already well seated in the data distribution. We validate on an image classification benchmark task that our proposed solution generalizes better as compared to the existing approaches and achieves higher test accuracies.

Our main contributions include:

- Improvement in classification accuracy by using phantom data points to overcome the base error in a dataset.
- A hyper-parameter free intrinsic regularization to enable training truly deep models.
- Evaluate our model on two popular datasets against established baselines and showcase our performance gains as well as training improvement qualitative and quantitatively.

2 Related Work

Training very deep networks effectively is an open question[20] due to the model complexity. Models with millions of parameters require a lot of data to train effectively, however, millions of training samples are not available for all tasks. A good example of the realistic amount of data needed is [2] with 16M instances.

That is not an option for all machine learning settings especially domains such as medicine [18]. Data augmentation [10] is an efficient method to ensure that data seen by the model is varied during training. Standard augmentation techniques include flipping, scaling, and padding.

Training these models from scratch can be avoided by using the weights of a model that has been trained on a similar dataset and then finetuning the model to fit your need [15] [17]. Transfer learning [15] has enabled training deeper models using a smaller dataset size however, if the goal is complete retraining than the training procedure needs to be adapted to ensure that the model does not memorize the training data.

Methods such as MaxOut [5] add layers into the architecture with a max activation function and have shown to positively impact the convergence behavior when compared to the ReLu activation [14]. DropOut, proposed in [19], addresses the problem of model overfitting by probabilistically turning off neurons in the final embedding layer to create an ensemble of models and has shown to be an effective way to regularize deep neural networks. Similarly in [25], the authors move the regularization from the final layer to the loss layer where they intentionally flip the labels in a mini-batch to ensure that the model generalizes. These methods seek to work on the architecture and loss layer to regularize the model. Methods such as weight decay [11] and batch normalization [8] are aimed at the optimizer and architecture and seek to penalize the weights while training to ensure that models generalize.

In [26] the authors propose the use of taking multiple instances and creating a linear combination of the instances and their label. Sampling from this mixup distribution allows them to learn on fabricated data points.

3 Methodology

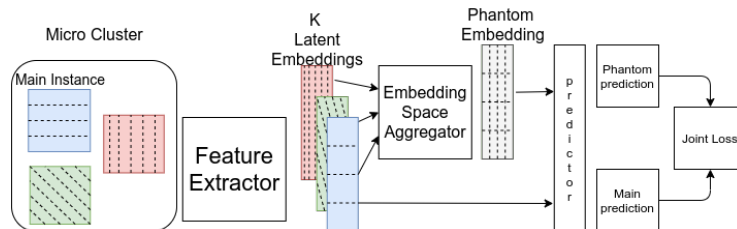


Fig. 2. A training step takes a micro-cluster with K samples, generating K embeddings which are aggregated to create the phantom embedding, This, along with the embedding of the main training instance is passed to the predictor. The combined loss for these predictions is calculated as in Eq. 3.

Consider a machine learning method $\psi(x)$ where x is a dataset sample and $x \in \mathbb{R}^{N \times M}$ corresponds to a multi-category target y where $y \in \{1, \dots, L\}^N$

among L classes. This model will produce a latent embedding: $\phi : \mathbb{R}^M \rightarrow \mathbb{R}^D$ of the features (the flattened layer after the final convolution block in our case), which is then passed to the prediction layer: $\psi : \mathbb{R}^D \rightarrow \mathbb{R}^L$, for the sake of notational brevity, we will use ϕ and ψ interchangeable with their parameters. The estimated target variable is therefore $\hat{y}_n := \psi(\phi(x_n)), \forall n \in \{1, \dots, N\}$ and the respective objective function:

$$\arg \min_{\psi, \phi} \sum_{n=1}^N \mathcal{L}(y_n, \psi(\phi(x_n))) \quad (1)$$

In this work we propose to make use of the shared similarities among the instances belonging to the same class and leveraging the collective learned representations of a small subset of instances to generalize the final embedding space. This is done by sampling a ‘micro-cluster’ of instances belonging to the same class. Note here that ‘cluster’ is being used in terms of a ‘group’ and has no relation to the unsupervised clustering methods.

Let us denote the number of instances in each micro-cluster as $K \in \mathbb{N}$ and the number of instances in each respective class as $N_l \in \mathbb{N}, \forall l \in \{1, \dots, L\}$ therefore for each class it is possible to draw $\binom{N_l}{K}$ many random choices. On these choices, consider, a new dataset transformation $(x, y) \rightarrow (x', y')$, where each element of x' represents a homogeneous cluster from x with K members and each element y' is the respective label of the instances within a homogeneous cluster. Since we are sampling homogenous clusters, $y' = y$. The total number of clusters is defined as $N' = \sum_{l=1}^L \binom{N_l}{K}$. The new input features are then $x' \in \mathbb{R}^{N' \times K \times M}$ and the new targets $y' \in \{1, \dots, L\}^{N'}$.

This new dataset transformation leads to a model output: $\hat{y}_n := \psi(\phi(x'_{n,k}))$ where $\phi(x'_{n,k})$ is the k^{th} latent embedding and $k \in K$. These K latent embeddings will be used to generalize the final learned embedding by aggregating them as see in Fig. 2. In our proposed approach we use a ”Mean Embedding Space Aggregator” which is explained as: $\phi'(x_n) = \frac{1}{K} \sum_{k=1}^K \phi(x'_{n,k})$ where $\phi'(x_n)$ is the phantom embedding from the micro-cluster. The naive approach would be to use this phantom embedding directly in the optimization, resulting in the following objective function:

$$\arg \min_{\phi, \psi} \sum_{n=1}^{N'} \mathcal{L}\left(y'_n, \psi\left(\frac{1}{K} \sum_{k=1}^K \phi(x'_{n,k})\right)\right) \quad (2)$$

However, Eq. 2 poses a problem since the intra-class variation of challenging datasets can cause the embedding to be too drastically modified, Also, datasets with multi-modal distributions and non-convex hulls can be adversely effected by the naive objective function (Eq. 2) since the micro-cluster can be sampled from the different modes of the data distribution. In its place we propose to use the phantom embedding in the loss function:

$$\mathcal{L} = \alpha \mathcal{L}(y'_n, \psi(\phi(x'_{n,k=0}))) + (1 - \alpha) \mathcal{L}(y'_n, \psi(\phi'(x'_n))) \quad (3)$$

In Eq. 3 we treat the first sample ($k = 0$) as the main instance and the others serve as a guide to improve the embedding space for this instance by ‘pulling’ the $k = 0^{th}$ towards the phantom embedding. We draw α from the beta distribution and it serves to add stochasticity in the combination of the embeddings and also removes the need for tuning α . Therefore our final objective function is:

$$\arg \min_{\phi, \psi} \sum_{n=1}^{N'} \left[\alpha \mathcal{L} \left(y'_n, \psi(\phi(x'_{n,k=0})) \right) + (1 - \alpha) \mathcal{L} \left(y'_n, \psi \left(\frac{1}{K} \sum_{k=1}^K \phi(x'_{n,k}) \right) \right) \right] \quad (4)$$

4 Experiments

In this section, we showcase the results of our approach and compare them with other methods in the domain. All the results presented have been recreated using the original author’s provided implementations. These experiments were carried out on NVIDIA 1080Ti, 2080Ti, and V100 GPUs.

4.1 Datasets and Implementation Details

To verify the efficacy of our proposed approach we have chosen two publically available datasets. CIFAR10 [9] and FashionMNIST [24] are popular image classification datasets and are widely used in the computer vision domain for testing new research. They comprise 60000 and 70000 images sized at 32x32 and 28x28 respectively. They offer a challenging problem setting due to the wide intra-class variation and inter-class similarities. Furthermore, these datasets are also easy to overfit the deep convolutional neural networks. Therefore, these datasets provide all the necessary challenges that our work proposes to address.

Our method can be readily included in any machine learning model, for our experiments we have chosen Deep Residual Networks (ResNet-18, ResNet-34, and ResNet-50) as proposed in [7] and as implemented in [13]. The networks under test were initialized as specified in [6] and optimized using Stochastic Gradient Descent (SGD) [12] with batch normalization [8] and a weight decay [11] factor of 0.0005, it should be noted here that the original ResNet architecture used 0.0001. The learning rate was set at 0.1 at the start than the scaled down by a factor of 10 at the 32k and 48k iterations as in [7], training was terminated at 64k iterations. We used a batch size of 128 and the dataset was augmented by padding 4 pixels to the image and translating the image accordingly, the images were also flipped horizontally and normalized by the mean and standard deviation of the entire dataset.

4.2 Results

In this section we evaluate our model by answering the following research question:

1. **RQ1:** Can classification accuracy be improved by creating a phantom embedding for data points?

2. **RQ2:** Can a better embedding space lead to a more robust model?
3. **RQ3:** Can we add intrinsic regularization by using the embedding space directly?

4.3 RQ1: Classification Accuracy

The baselines were chosen based on their relevance to the approach that we have outlined in this paper. We have used the DisturbLabel [25] as implemented in [3], ResNet with Dropout [19] and we also compare against the vanilla variants of the ResNet architectures. DisturbLabel seeks to regularize the loss layer rather than the parameters and DropOut seeks to create an inherent ensemble of neural networks by stochastically turning off a certain amount neurons in the embedding layer to prevent the models from learning the training data. A comparison of our method to the baselines can be seen in Tab. 1.

Table 1. Classification Accuracy on CIFAR using ResNet-18 architecture. We report the final accuracy as **Acc** and also the **Mean** and **Max** accuracies for the last 5 epochs to illustrate training stability towards convergence.

	Accuracy		
Method	Acc	Mean	Max
ResNet18	93.5	93.68	93.68
ResNet18 Dropout	94.11	94.09	94.21
ResNet18 DisturbLabel	94.2	94.28	94.33
Phantom ResNet18	94.91	94.84	94.91

It can be seen in Tab. 1 and 2 that our proposed method is performing better than all the baselines in terms of the accuracy, however, it should also be noted that the overall variance in the results at the time of convergence is also better than the baselines.

Table 2. Classification Accuracy on CIFAR using ResNet34

	Accuracy		
Method	Acc	Mean	Max
ResNet34	93.65	93.71	93.79
ResNet34 Dropout	93.92	93.97	94.03
ResNet34 DisturbLabel	93.73	93.79	93.81
Phantom ResNet34	94.52	94.52	94.6

For Phantom ResNet, we see a 1.47% and 0.88% gain for ResNet-18 and ResNet-34 accuracies respectively. The decrease in the overall ‘performance gain’ when moving from ResNet-18 to ResNet-34 can be attributed to the fact that

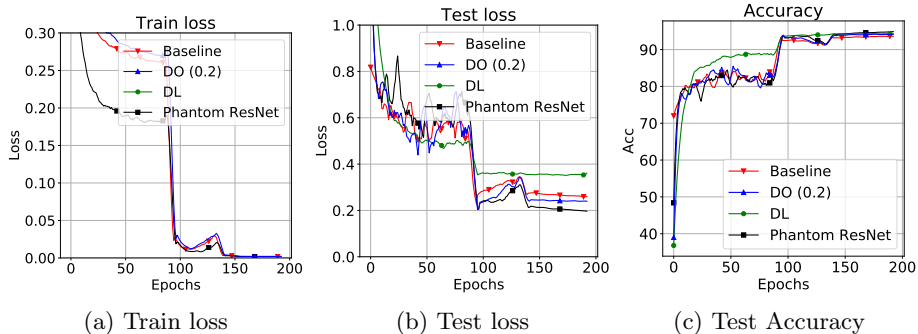


Fig. 3. ResNet-18 Training and testing behaviors: **Baseline** refers to the original network baseline while **DO** and **DL** refer to the DropOut and DisturbLabel baselines.

ResNet-34 is a more complex model. ResNet-18 has 0.27M parameters while ResNet-34 has 0.46M, so by doubling the parameters of the network we expect a more expressive model that already improves upon the shortcomings of the former. A more important trend in Tab. 2 is the behavior of ResNet34 Dropout and ResNet34 DisturbLabel values for which we only see an improvement over ResNet34 of 0.27% and 0.1% respectively. In Tab. 1, for ResNet18 Dropout and ResNet18 DisturbLabel we saw an improvement of 0.61% and 0.7% over the vanilla ResNet18. It can be seen that the Dropout and DisturbLabel, while still better than the vanilla ResNet lose a significant amount of their gains when the model parameters double from ResNet18 to ResNet34 i.e model complexity increases. These methods do not take into account the highly similar embeddings of data points from different classes during optimization and thus, suffer in final accuracies. Our method uses the latent representation from multiple instances of a class to regularize the model and prevent the highly similar data points from different classes from being too close to the decision boundary.

Table 3. Classification Accuracy on FashionMNIST

Method	Accuracy	
	ResNet-18	ResNet-34
ResNet	94.78	94.93
ResNet-Dropout	94.97	95.11
ResNet-Disturb	94.95	94.97
Phantom ResNet	95.07	95.38

In Tab. 3 we can see that the results for our approach continue to outperform the baselines on the FashionMNIST dataset which comes with its own

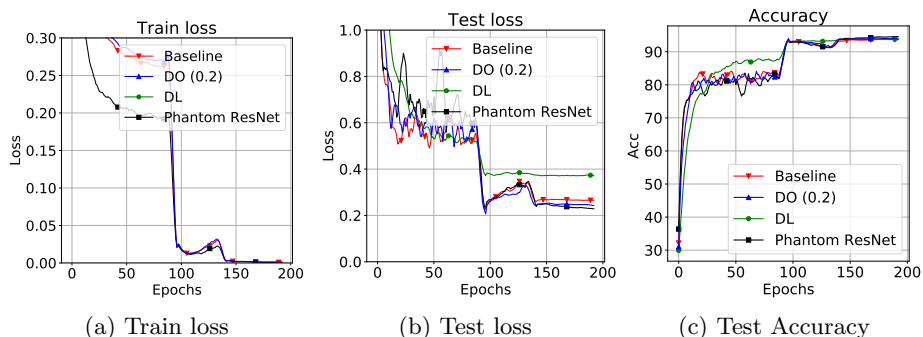


Fig. 4. ResNet-34 Training and testing behaviors: With a more complex network, our method continues to outperform the baselines.

set of challenges since the images are now 28x28 and comprise of a single channel rather than the standard RGB channels of CIFAR.

Consistent accuracy improvement across these datasets and over varying architecture complexities shows that our method is robust enough to deal with a wide variety of scenarios. Furthermore, it should be pointed out that the accuracies for the baselines required a large hyper-parameter search to get to these values whereas our proposal required no such search for performance.

4.4 RQ2: Robustness

A sufficiently well-trained algorithm should be able to reduce the error on the test set, the reduction of test error is inextricably tied to the training process. Our proposed methods seeks to mitigate overfitting by enriching the embedding space ensuring that the model generalizes well thus preventing errors in similar classes. It can be seen in Fig. 3b that our model is converging to a lower Test loss, this is an outcome of the enriched embedding space that actively helps optimize the model to learn a more general representation from the training data. The outcome of this approach reflects readily in Tab. 1 in the final accuracies, furthermore considering Fig. 3c it can be seen that our proposed model takes a more deliberative approach in the initial learning stage up to the first 100 epochs. While other models are shooting up quickly in accuracy values, and then later failing to maintain their lead, our approach focuses on learning better representations and penalizing itself when it doesn't more aggressively in order to arrive at the better final optimal network weights.

The same trend is observed when training ResNet-34 as shown in Fig. 4. The only difference being that models not trained with inherent embedding space enrichment in mind suffer more due to the higher complexity of the underlying networks. In both Fig. 3 and Fig. 4 it can be seen that ResNet-Dropout seems to be more stable in terms of its fluctuations during the middle of the training process, between epoch 100 and 150, however it still fails to match our method

Table 4. Classification Accuracy on CIFAR using ResNet50

Method	Accuracy		
	Acc	Mean	Max
ResNet50	93.86	93.25	93.34
ResNet50 Dropout	93.21	93.17	93.25
ResNet50 DisturbLabel	94.37	94.352	94.38
Phantom ResNet50	94.48	94.54	94.71

in the final loss as well as final accuracy. This highlights the problems laid out in the introduction section where a model loses on accuracy in an attempt to not overfit.

4.5 RQ3: Intrinsic Regularization

As stated earlier, training deep models are hampered by the model memorizing the training data and then showing poor performance on the test data. This problem comes to the forefront when dealing with a truly deep model like ResNet-50 which comes with 0.88M trainable parameters. Training such a model from scratch requires an immense amount of data or a clever regularization scheme. The scheme needs to be searched for over several runs and hyper-parameter configurations. This is a time-consuming and expensive procedure since training ResNet-50 can take up to 7-11 hours on a modern GPU. Our proposed method allows for the data samples to contribute not just to the learning but to the regularization as well, Tab. 4. By intrinsically learning the regularization with the help of similar images and generalizing the weights of our embedding layer with our proposed phantom embeddings we are able to regularize the model as it trains. This behavior is on display in Fig. 5 where it can be seen that our model is leading to a marked lower test loss while the baseline models struggle to match its performance. Given enough time (days) an ideal configuration for the baselines could be arrived to match the performance of our model however, our model provides it without the need for the extensive search required by the baselines.

In Fig. 5b we intentionally allowed the models to run past their convergence point to see how the baseline and our model handle such cases. It can be seen that the baselines runs off and starts to overfit, leading to an increasing test loss while our method shows a noticeably better performance and maintains a lower test loss.

4.6 Ablation Study

In order to showcase the effect of different numbers of samples from the same class (K) we varied K from 1 (baseline) to 7 and in Tab. 5. It was seen that while increasing K led to increasing performance over the baselines, the percentage gain vs model complexity didn't justify the use of higher K . All the results reported have been therefore conducted with $K = 2$

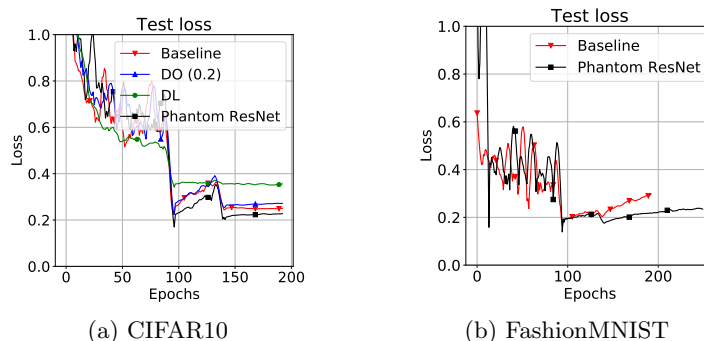


Fig. 5. Intrinsic Regularization in ResNet50: The phantom embeddings prevent overfitting even when the training regime is specifically aiming to overfit.

Table 5. Ablation Study: Investigating the effect of increasing K on the classification accuracy.

	Accuracy			
Model	$K=1$	$K=2$	$K=3$	$K=4$
ResNet 18	93.5	94.91	94.01	93.9
ResNet 34	93.65	94.58	94.3	94.2

5 Conclusion

In this paper, we have shown how embedding spaces can be directly used to regularize deeper neural networks by creating phantom embeddings around the true data points by aggregating the embeddings together and then optimizing the model with the phantom embedding as a co-target. We have shown how our method outperforms the baselines two famous and competitive datasets. Our method also introduces an intrinsic regularization which enables us to train deeper models without an extensive hyper-parameter search.

References

1. Chapelle, O., Weston, J., Bottou, L., Vapnik, V.: Vicinal risk minimization. In: Advances in neural information processing systems. pp. 416–422 (2001)
2. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
3. Farzaneh, A.: Disturblabel-pytorch (2019), <https://github.com/amirhfarzaneh/disturblabel-pytorch>
4. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 1440–1448 (2015)
5. Goodfellow, I., Warde-Farley, D., Mirza, M., Courville, A., Bengio, Y.: Maxout networks. In: International conference on machine learning. pp. 1319–1327 (2013)

6. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE international conference on computer vision. pp. 1026–1034 (2015)
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
8. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
9. Krizhevsky, A., Nair, V., Hinton, G.: The cifar-10 dataset. online: <http://www.cs.toronto.edu/kriz/cifar.html> **55** (2014)
10. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)
11. Krogh, A., Hertz, J.A.: A simple weight decay can improve generalization. In: Advances in neural information processing systems. pp. 950–957 (1992)
12. LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D.: Backpropagation applied to handwritten zip code recognition. *Neural computation* **1**(4), 541–551 (1989)
13. Li, K.: `kuangliu/pytorch-cifar` (2017), <https://github.com/kuangliu/pytorch-cifar>
14. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: ICML (2010)
15. Pratt, L.Y.: Discriminability-based transfer between neural networks. In: Advances in neural information processing systems. pp. 204–211 (1993)
16. Simard, P.Y., LeCun, Y.A., Denker, J.S., Victorri, B.: Transformation invariance in pattern recognition: tangent distance and tangent propagation. In: *Neural networks: tricks of the trade*, pp. 239–274. Springer (1998)
17. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
18. Sirinukunwattana, K., Raza, S.E.A., Tsang, Y.W., Snead, D.R., Cree, I.A., Rajpoot, N.M.: Locality sensitive deep learning for detection and classification of nuclei in routine colon cancer histology images. *IEEE transactions on medical imaging* **35**(5), 1196–1206 (2016)
19. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* **15**(1), 1929–1958 (2014)
20. Srivastava, R.K., Greff, K., Schmidhuber, J.: Training very deep networks. In: Advances in neural information processing systems. pp. 2377–2385 (2015)
21. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1–9 (2015)
22. Vapnik, V., Vapnik, V.: *Statistical learning theory* wiley. New York **1** (1998)
23. Vapnik, V.N., Chervonenkis, A.Y.: On the uniform convergence of relative frequencies of events to their probabilities. In: *Measures of complexity*, pp. 11–30. Springer (2015)
24. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747 (2017)
25. Xie, L., Wang, J., Wei, Z., Wang, M., Tian, Q.: Disturblabel: Regularizing cnn on the loss layer. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4753–4762 (2016)
26. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412 (2017)

A hierarchical multi-level product classification workbench for retail

Maximilian Harth, Christian Schorr, and Rolf Krieger

Trier University of Applied Sciences, Environmental Campus Birkenfeld,
55761 Birkenfeld, Germany
c.schorr@umwelt-campus.de

Abstract. Exploratory data analysis and especially model evaluation get difficult when facing the challenge of classifying product data according to a hierarchical classification system (HCS) containing thousands of categories as used in the retail industry. The identification of incorrectly classified products and the optimization and monitoring of automatic classification algorithms is very time-consuming. To solve this problem we propose a workbench which provides an interactive graphical user interface (GUI) for exploratory product data analysis and model evaluation taking into account the structure of an HCS and supplying statistical insights. In addition, the workbench offers an integrated machine learning based product classification module which can classify products on the fly using their names only.

Keywords: Product classification, machine learning, data exploration, hierarchical classification systems

1 Introduction

The management of product data is an important task in retail companies. Products must be classified based on a hierarchical product classification system (HCS) which defines categories of products and relations between them. The assignment of products to the categories is based by either implicitly or explicitly defined attributes [1]. A well-known product classification standard for retail is the Global Product Classification (GPC) described in [2]. It defines a four-level hierarchy consisting of 38 segments, 118 family, 823 class and 4226 brick codes to describe products. It is used by 60.000 German companies and over 1.5 million companies world-wide. In many cases standardized HCSs coexist with company-specific ones that have a similar number of categories and hierarchical levels. Consequently, new products have to be continuously classified into different hierarchical classification systems.

Product data is of central importance in retail companies. There are companies having millions of product data records. Correct assignment of products to an HCS has a

Copyright © 2020 by the paper's authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

decisive influence on data quality, the execution of business processes and on a seamless data exchange between business partners. Consequently, consistent (re)classification between different HCSs is a complex, time-consuming and error-prone task, which must often be performed manually. Due to the large number of categories it is difficult to classify the products consistently. It is therefore necessary to identify and correct misclassifications.

In addition, to reduce the effort required for the classification, there are numerous approaches for automatic classification based on machine learning. The typical data science project can be described by an iterative process encompassing the steps business and data understanding, data exploration and cleaning, modeling and deployment. Widely-used process models are the cross-industry-standard for data mining [3] or Microsoft's Team Data Science Process [4]. Modeling encompasses feature engineering, model training and evaluation. For all these steps a plethora of software and libraries is available. Metrics like precision, recall or F_1 -score can summarize the model quality in a few numbers but understanding the results that way and learning from incorrectly classified products is very time-consuming. Confusion matrices are also often employed to visually determine systematic misclassifications at a glance. Keeping in mind that a HCS may have thousands of different classes at the lowest level, though, the resulting confusion matrix has several million entries and is not easy to interpret. For a retail company a tool which allows to interactively explore a HCS and the predicted classes would be a valuable tool.

Consequently, we propose a workbench which provides an interactive graphical user interface (GUI) for exploratory product data analysis, identifying classification errors and model evaluation taking into account the structure of a HCS and supplying statistical insights using dendrograms, tree maps and box plots at the different levels of an HCS, for example. It also features a machine learning module to (re)classify products on the fly according to the underlying HCS. To summarize the workbench supports the process of the manual and automatic product classification based on several hierarchical classification systems.

2 Related Work

Much research is undertaken in the field of product classification with machine learning - a recent and comprehensive comparison of classification algorithms can be found in [5].

Sun et al. suggests a hybrid algorithm called Chimera. A mix of crowd outsourced manual classification, machine learning and data quality is used in addition to rules formulated by in-house analysts [6]. Based on product name, product description and several other attributes, several tens of millions of products are classified into more than 5000 categories. Ha et al. suggest a deep learning-based strategy employing multiple recurrent neural networks (RNNs) [7]. In addition to the product name, brand, manufacturer and the top level category – all in Korean - are used. The data consists of more than 94 million products with 4016 low-level categories.

Several classification methods on a hierarchical data set of product descriptions are studied by Ding et al. [8]. A hierarchical classification approach using the UNSPSC categories leads to a significantly worse result, contrary to intuition. Cevahir and Murakami present a classification model assigning products to one of 28.338 possible categories on a five-tier taxonomy using 172 million Japanese and English product names and descriptions [9]. The model combines deep belief nets, deep auto-encoders and k-Nearest Neighbour-classification (kNN). In [10] the authors present a classification model for GPC categorized products using partially abbreviated German product names only.

In summary, we can say that the field of automatic classification using machine learning is highly topical. Often hierarchical classification systems are considered, which contain thousands of categories at the lowest level. There are numerous models whose quality is also influenced by the field of application. Consequently, the development of suitable models in practice requires an intensive evaluation and optimization. Furthermore, the models must be continuously monitored in their practical application.

Often an evaluation is only carried out at the lowest level of a classification system. However, for the systematic evaluation of models it is advantageous to consider the entire hierarchical structure of a classification system.

Our proposed workbench could be a step to optimize the model by explicitly analyzing the results with respect to the special structure of an HCS.

Several software options for big data visualization exist for commercial use. Usually, a company using an ERP system like SAP², builds its own specific add-ons using what options the ERP system provides for customization. A visual exploratory analysis of different product classifications is only possible to a limited extent.

Beside full-fledged software solutions, numerous frameworks for building custom applications exist. Dash Open Source³ is a Python-based open source framework for building machine learning and data science web apps. While it supports integrated classification the user has to program the actual application for himself. The focus of the JavaScript framework D3.js⁴ lies on complex interactive web graphics applications. It can also be used with languages like R and Python, but does not provide classification algorithms. As with Dash, the user is required to implement what he wants on his own.

In sum we found no ready-to-install software dedicated to visualizing and manipulating hierarchical product data, especially not with added machine learning based classification capabilities. We see the express need of retail companies for such a workbench, though, which in our opinion is crucial for the successful utilization and high acceptance of automatic classification in practice.

² 1 <https://www.sap.com/germany/industries/retail.html> (last accessed 03.08.2020)

³ 3 <https://plotly.com/dash/> (last accessed 03.08.2020)

⁴ 4 <https://d3js.org/> (last accessed 03.08.2020)

3 Hierarchical classification workbench

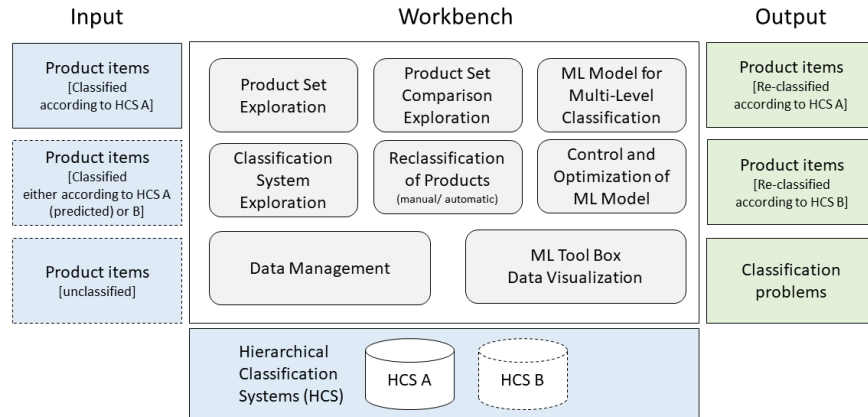


Fig. 1. Schematic overview of the proposed workbench

The main functional modules of the workbench are shown in figure 1. As input the workbench needs at least one hierarchical classification system (HCS A) and a set of classified product data. If the comparison of classification systems is desired a second classification system (HCS B) is also a required input. Using ML methods, the workbench outputs the product items reclassified according either to HCS A or HCS B. If the quality of a classification model is to be evaluated, HCS A and HCS B are identical. At the moment the (re)classification modules have been developed as a stand-alone software, but not yet been integrated into the workbench prototype.

3.1 Data exploration functions

Classification system and product set exploration

Our workbench utilizes diagrams for visual analysis to detect misclassified products in a data set. First basic information about the HCS itself is provided regarding the number of hierarchy levels, the amount of different categories for each level and the number of products in the data set. If more than one HCS is given, the number of products belonging to the same category in both HCSs is determined. In a second step, the distribution of the categories on each hierarchy level can be investigated. To this end, an interactive dendrogram picturing the hierarchical structure of the HCS can be generated in order to provide a first visual aid for further analysis (fig.2). With the help of tree maps and heat maps the distribution of products to the HCS can be visualized. This allows to identify unbalanced categories of the HCS regarding the data set. If a specific category contains only a few products compared to other categories, this could cause the corresponding training data for the subsequent model training to be unbalanced. As a consequence, the prediction quality for this category will degrade. Using our work-

bench, these imbalances can be detected and mitigating actions taken before the prediction model is trained. Additional product data from external data pools could be requested or oversampling algorithms employed to counter the imbalance of the specific categories.

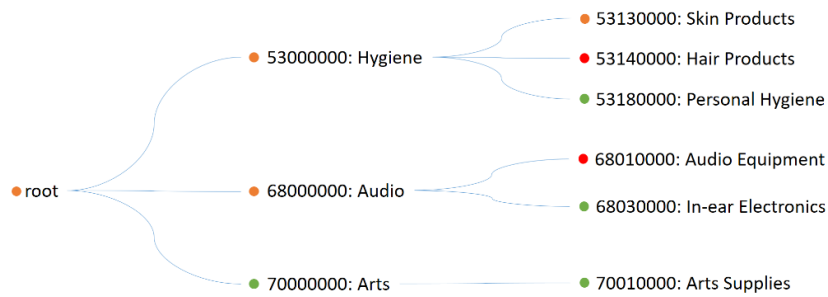


Fig. 2. Section of an interactive dendrogram of a hierarchical product classification as provided by the workbench. Dots denote categories where all (green), some (orange) or no (red) sub-categories contain at least one product.

Product set comparison exploration

The workbench supplies a comparison exploration module to compare two different HCSs in order to detect and identify misclassified products (fig. 3). It has to be kept in mind, that either of the two HCSs may contain misclassified products and that a seemingly wrong predicted classification according to HCS A could also mean that the classification of HCS B is already wrong. These errors can and do happen with real-world data. For example, the classification of a product p according to HCS B is suspect if a high proportion of the products that belong to the same category as p in HCS A belong to another category in HCS B. To identify these products and to check their classification is important since the ML model is based on the assumption that all products in the training data set are correctly classified. A model based on erroneous data usually delivers inferior results.

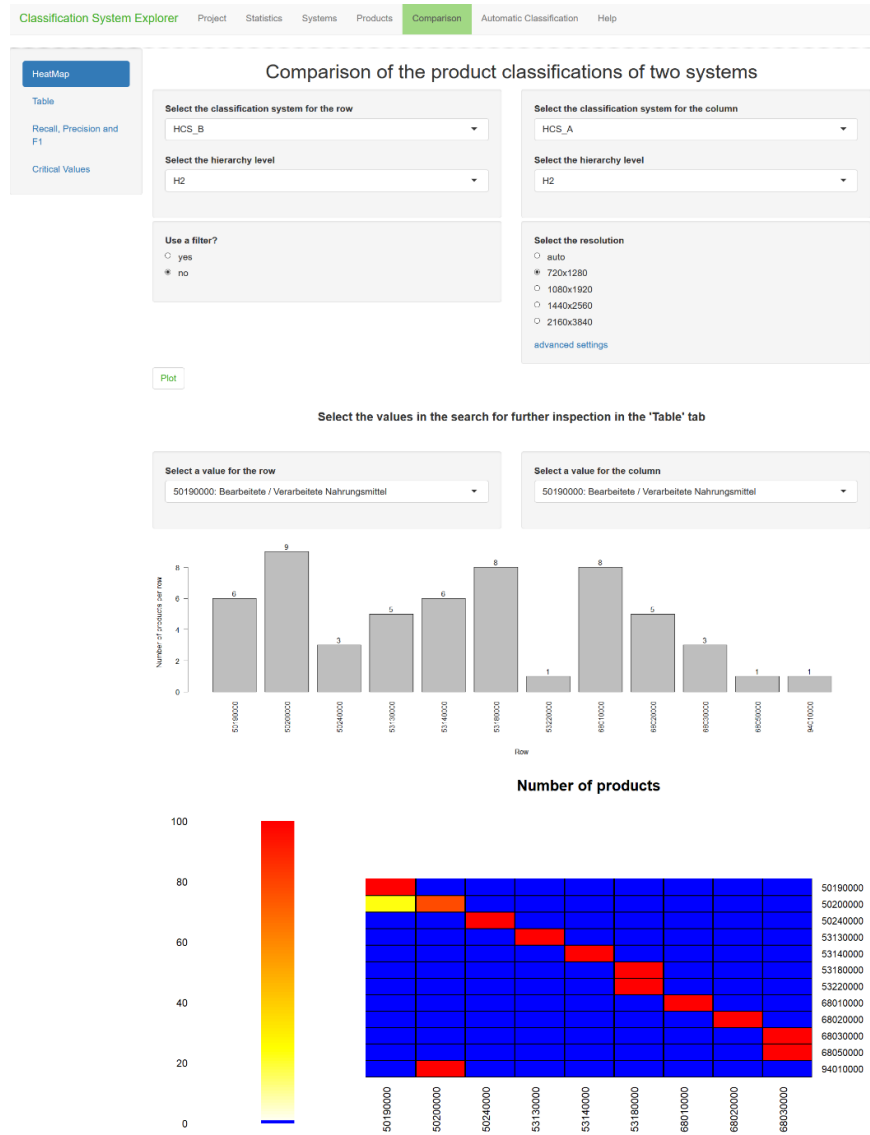


Fig. 3. Graphical user interface showing the comparison of two classifications based on a heat map. The hierarchical level can be selected by the user.

3.2 Classification functions

ML Model for multi-level classification

One of the main advantages of the workbench is the integrated module for interactive multi-level classification of products. It is planned that the user will be able to call the module from the dendrogram view of the workbench and classify a product on the fly. Depending on the company-specific classification system the user employs, different pre-trained models can be added.

The prototype of the workbench has an integrated machine learning model for multi-level classification of products according to the Global Product Classification (GPC) standard, based on the results of [10]. To measure the model performance we use the weighted metrics precision (P_{wg}), recall (R_{wg}) and F_1 score (F_{1wg}). These weighted versions explicitly take the multi-class structure of a product data set into account by computing a weighted average of the respective micro and macro metrics. This makes it possible to calculate the metrics at different hierarchical levels and to identify main categories, categories or sub-categories where the quality of the model is insufficient.

Control and Optimization of ML models

The usual quality metrics model evaluation are accuracy, precision, recall and F_1 -score, which only assess the quality, but do not provide explanations why the model performs as it does. In multi-level classification a confusion matrix is often used for deeper model evaluation. However, it only shows the distribution of the products of a given category to all possible categories and does not take into account that the categories themselves are ordered hierarchically. If a product is not predicted correctly on a given category, it nonetheless may be correct regarding the next higher hierarchical category level. This would be a less grave misclassification than predicting a wrong category within an also wrong overlying category. We define the severity of such an error according to the hierarchy level on which the error first starts. For GPC with four hierarchy levels, a product with a wrongly predicted category on brick level, but correct on class level is called a “level 4 error”. If the product classification is also wrong on class level, but correct on family level it is a “level 3 error” and so on (fig. 4).

In addition our workbench offers heat maps to show the products and their predicted classification embedded in the overall hierarchy. The user has the possibility to list all misclassified products and to correct their classification manually. Figure 5 shows an example where the product “Breaded cauliflower” belonging to the category “Vegetables – Prepared/Processed (Frozen)” has been wrongly assigned to the category “Vegetables – Unprepared/Unprocessed (Frozen)”. The workbench shows this error by colouring the product in red. Following the hierarchy levels up one can see that the classification has already failed on family level and is thus a level 2 error.

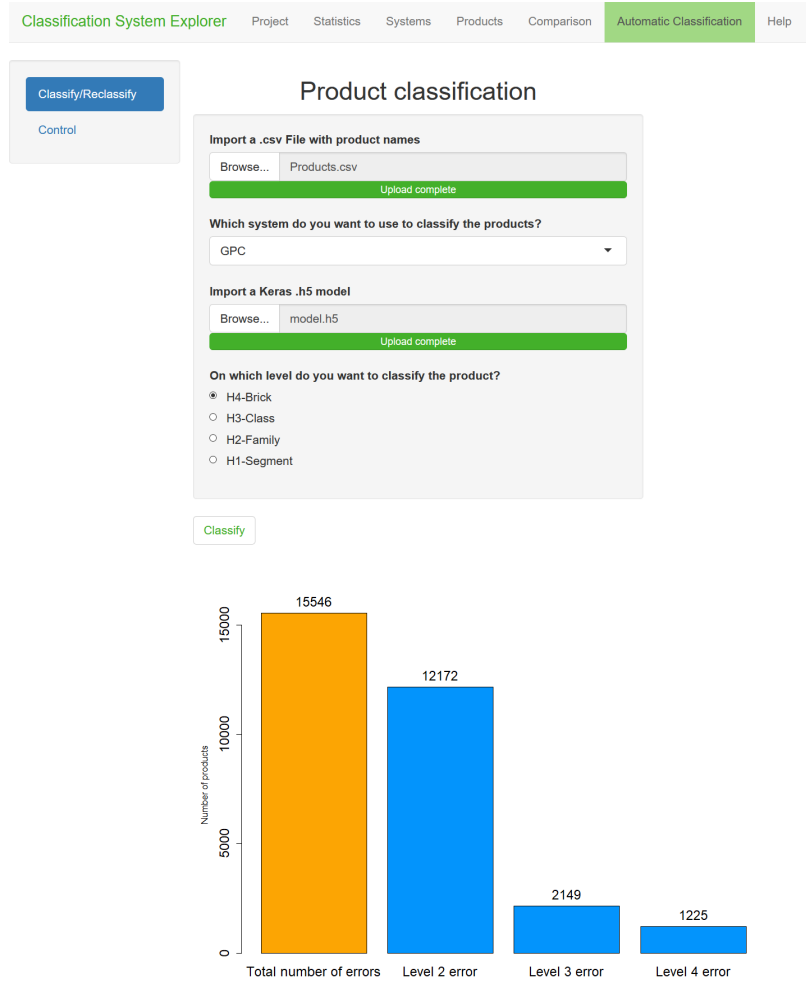


Fig. 4. Prototype of the user interface for bulk product classification. (The number of wrongly classified products regarding the hierarchy level on which the products become correctly classified are displayed.)

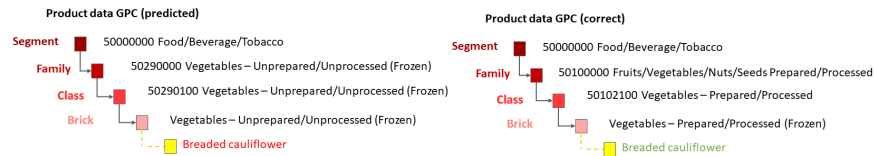


Fig. 5. Predicted / correct GPC brick (level 2 error)

(Re)classification of products

The option to (re)classify a product by its name is one of the main advantages of the proposed workbench. Usually a company uses not only its own custom classification system but is forced to handle also the classification systems its suppliers use. Importing product data from a supplier's data pool necessitates a reclassification into the company classification system. This is often done manually or using mapping tables, both error-prone or time-consuming methods. The proposed workbench offers the option to classify a product according to a given classification system by entering the product name into a field. The integrated ML model returns a suggestion for the appropriate classification regarding the desired classification system. A prototype of this functionality is shown in figure 6.

Classification System Explorer Project Statistics Systems Products Comparison Automatic Classification Help

Classify/Reclassify
Control

Product classification

Enter a product name

Which system do you want to use to classify the product?

Import a Keras .h5 model

On which level do you want to classify the product?
 H4-Brick
 H3-Class
 H2-Family
 H1-Segment

TOP 3 Suggestions

Highest probability	Second highest probability	Third highest probability
10000248	10000249	10000598

Fig. 6. Prototype of the user interface for on-the-fly classification for single products using the product name providing the three predictions with highest probability (Top 3)

4 Conclusion

The proposed workbench is especially designed to support the exploratory analysis of product data classified according to a given hierarchical classification system, the comparison of different product classifications to identify errors and inconsistencies and the evaluation of machine learning models for the automatic classification of products. It supports the whole machine learning pipeline for product classification in an integrated environment. We expect that the usage of the workbench will significantly accelerate and simplify the model evaluation process in practice.

Preliminary evaluation in a major German retail company has been met with success. The workbench was tested on 40.000 products classified according to both a company specific classification system and to GPC managed by an ERP system. Especially, the use of heat maps to compare the product classification at different levels showed hitherto undetected errors in the current classification and was greatly appreciated. The visualization with interactive dendrograms also served to analyze problems in the existing data base. All these features were not available in the company's ERP system but were much valued by the customer.

5 Outlook

Our next step is to integrate all of the already developed ML components into the workbench and to add the option to interactively assign the correct classification to the wrongly classified product directly in the training data set using the dendrogram view. After error correction, the machine learning model can be retrained to improve the classification quality. Using the workbench, users can then monitor the results of the automatic classification. In doing so we hope to increase the acceptance of machine learning methods for automatic product classification in practical applications.

We also plan to evaluate the complete workbench thoroughly with a another major retail company in order to ready it for actual deployment in a productive environment.

Acknowledgements

Part of the research presented in this paper was funded by the German Ministry of Education and Research under grant FKZ 01|S18018.

References

1. Hepp, M., Leukel, J., Schmitz, V.: A quantitative analysis of product categorization standards: content, coverage, and maintenance of eCI@ss, UNSPSC, eOTD, and the RosettaNet Technical Dictionary, Knowledge and Information Systems 13.1, pp. 77–114 (2007)
2. GS 1 homepage, <https://www.gs1-germany.de/gsl-standards/klaskifikation/produktklaskifikation-gpc/> (last accessed 03.08.2020)

3. Shearer C.: The CRISP-DM model: the new blueprint for data mining, *J Data Warehousing* (2000); 5:13—22
4. Microsoft: Team Data Science Process (TDSP). <https://docs.microsoft.com/en-us/azure/machine-learning/team-data-science-process/overview> (last accessed 03.08.2020)
5. Chavaltada, C., Pasupa, K., Haroon, D.R.: A Comparative Study of Machine Learning Techniques for Automatic Product Categorisation. In: *Proceedings of Advances in Neural Networks - ISNN* (2017)
6. Sun, C., Rampalli, N., Yang, F., Doan, A.. (2014) Chimera: Large-Scale Classification using Machine Learning, Rules, and Crowdsourcing. *Proceedings of the VLDB Endowment*, Vol. 7, No. 13
7. Ha, J.W., H. Pyo, Kim, J. (2016). Large-scale item categorization in e-commerce using multiple recurrent neural networks. *Proceedings of the 22nd ACM SIGKDD*
8. Ding, Y., M. Korotkiy, B. Omelayenko, V. Kartseva, V. Zykov, M. Klein, E. Schulten and D. Fensel (2002). GoldenBullet: Automated Classification of Product Data in E-commerce. *Proceedings of BIS 2002*
9. Cevahir, A., Murakami, K.: Large-scale Multi-class and Hierarchical Product Categorization for an E-commerce Giant. In: *Proceedings of COLING 2016*, pp. 525–535 (2016)
10. Allweyer, O., Schorr, C., Krieger, R., Mohr, A.: Product classification based on partially abbreviated product names in retail. In: *9th International Conference on Data Science, Technology and Applications, Online* (2020)

Comparison of knowledge based feature vector extraction and geometrical parameters of Photovoltaic I-V Curves

C. Basoglu^{1,4}, G. Behrens¹, K. Mertens², and M. Diehl³

¹ Fachhochschule Bielefeld, Solar Computing Lab, Artilleriestraße 9, 32427 Minden, Germany

² Fachhochschule Münster, Fachbereich Elektrotechnik und Informatik, Photovoltaik-Prüflabor, Steinfurt

³ photovoltaikbuero, Ternus und Diehl GbR, Rüsselsheim

⁴ Contact: cbasoglu@fh-bielefeld.de

Abstract. Current methods for evaluating the performance of PV modules and systems in the field are exposed to weather conditions during system evaluation. The experimental measurement of performance naturally requires a corresponding amount of solar radiation, which is not available at all times of the year. The aim of this work is the development of a method for the weather-independent PV plant evaluation using the so-called dark I-V curve and an artificial neural network (ANN). The dark I-V curve can be measured at any time of the year and in any weather condition. In combination with the performance measurements from conventional methods an extensive database is already available, which was used as the ground truth for the development of the proposed model. The results show that with the proposed method a prediction of the power output for illumination levels above $800W/m^2$ a maximum prediction error below 10% is achieved. Thus, the dark I-V curve can be used for a weather-independent evaluation of PV systems in order to show first indications of performance losses and further analysis.

Keywords: Machine Learning · Artificial Neural Network · I-V Curve

1 Introduction

The in-field evaluation and fault diagnosis is crucial for a high-yield operation of photovoltaic plants. Analyzing the light I-V curve (current-voltage curve) of a PV array is the commonly used method for in-field evaluation and characterisation [2,3,4,5,9,12]. The I-V curve (see Figure 1) describes the energy conversion capacity under given conditions of irradiation and temperature. Only the experimental measurement of the I-V curve is able to specify with precision the electrical parameters of a photovoltaic cell, module or array.

Copyright © 2020 by the paper's authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

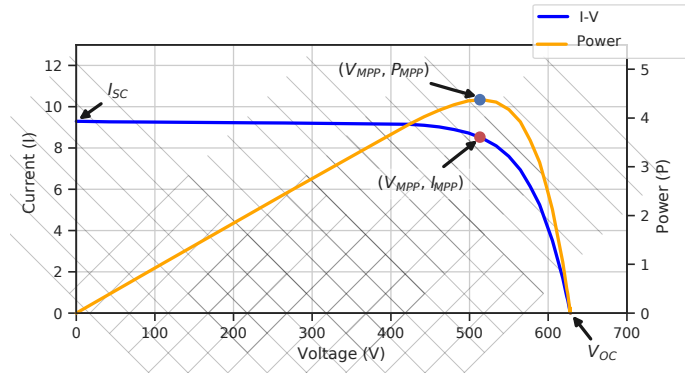


Fig. 1: I-V curve of a solar cell

The I-V curve starts at the short-circuit condition, I_{SC} , where the voltage is zero. The current decreases slightly as the voltage is increased, until the curve nears the open-circuit condition where the current rapidly drops off. The curve ends at the open-circuit condition, V_{OC} , with the current at zero. At some point on the I-V curve, the power of the cell is at its maximum. This point is known as the maximum power point (MPP), and solar cells are the most efficient at converting light energy into electrical energy at this point.

The in-field experimental measurement of the I-V curve is highly dependent on the weather conditions during the evaluation of the system. Passing clouds or shadows from other objects at certain times of the day result in a not negligible loss of time. Furthermore, the measurement requires sufficient light irradiation which is not available in all seasons of the year. To address these issues, this study proposes a novel method by predicting the light I-V curve using the so called dark I-V curve and an ANN. The dark I-V curve is measured without illumination by using an external power supply as reverse current source and is commonly used in the manufacturing process of solar modules [1,6,8]. Thus makes the dark I-V curve independent of weather conditions while maintaining many of the previously described electric characteristics.

2 Related Work

There are already several approaches which can identify faults using dark I-V curves for diagnostic purposes. For example, the method proposed by Mertens, K. et al. [10] is able to detect potential induced degradation (PID) and diode errors using numerical analysis of the dark I-V curve.

Besides the diagnostic value of the dark I-V curves, there are only a few methods which focus on predicting the light conversion performance under different illumination levels. King et al. [8] uses the two diode model (see Equation 1) with some experience based parameter assumptions to extract the remaining parameters of the model. Mertens et al. [11] uses also the two diode model and

module parameters from the manufacturer data-sheet to solve remaining parameters of the model. Both methods extract the two diode model parameters using the dark I-V curve and use these parameters to calculate the light I-V curve. In this work, one approach also use the two diode model to extract the model parameters of both, the dark and light I-V curve, and uses this knowledge to learn the relationship between them.

3 Method

In cooperation with our research partners, a database with 3424 light and 1656 dark I-V curves from field and laboratory measurements of 131 different module types has been collected. Each I-V curve consists of 200 current-voltage pairs and additional metadata like irradiance level and temperature. To build the ground truth for the proposed method, the dark I-V curves of each module type are combined with all light I-V curves of the same module type, which results in 37686 training and validation data sets for the neural network. Three different feature extraction approaches for the I-V curve are considered. Each approach generates a feature set of the dark and light I-V curve. The feature set of the dark I-V curve is used as the input vector and the light I-V curve as the output vector for the neural network.

The first approach (E1) uses the the electrical two diode model [7] and performs the levenberg-marquardt curve fitting algorithm to fit the following function to the measured data points.

$$I = I_{PH} - I_{S_1} \cdot \left(e^{\frac{U+I \cdot R_S}{\eta_1 \cdot U_T}} - 1 \right) - I_{S_2} \cdot \left(e^{\frac{U+I \cdot R_S}{\eta_2 \cdot U_T}} - 1 \right) - \frac{U + I \cdot R_S}{R_P} \quad (1)$$

The extracted series resistance (R_S), parallel shunt resistance (R_P), saturation currents ($I_{S_{1,2}}$), diode ideality factors ($\eta_{1,2}$) and temperature coefficient (U_T) of the dark and light I-V curves, are used as an input and respectively output vector for the neural network.

The second approach (E2) for feature extraction performs a principal component analysis of the I-V curves while retaining 95% of its variance. This approach generates 64 components for the dark I-V curve and 76 components for the light I-V curve.

The last approach (E3) uses a barycentric lagrange interpolation to extract 20 equidistant points for each I-V curve. Since the points are equidistant, the values on the x-axis (voltage) are removed and only implied by its position in the vector.

The architecture of the neural network for all approaches consists of the input, output and a single hidden layer. The number of nodes in the hidden layer equals half the average of the input and output layers. The output of the nodes in the hidden layer are controlled by the tangens hyperbolicus (tanh) activation function and the network is optimized using the RMSE error function with RMSprop optimization algorithm. Finally the dense networks are trained with 75% of the combined data sets, using 80% of it for the training and 20% for the optimizer validation to avoid over fitting.

4 Experimental Results

For the final validation of the different approaches, 25% of the combined datasets are used. Table 1 shows the mean percentage error (MPE) using min/max and percentiles for the error distribution. For this purpose the light I-V curves are reconstructed from the features described and compared to the measured I-V curves. Overall, the third approach (E3) using the equidistant interpolation of

	Mean	SD (σ)	Min.	25%	50%	75%	Max.
E1	6.08	4.82	0.86	2.89	4.80	8.30	25.28
E2	9.12	6.41	1.45	4.53	8.67	12.58	32.94
E3	4.12	3.44	0.39	1.66	3.28	5.58	21.19

Table 1: Mean percentage error (MPE) and percentiles using the different pre-processing methods

the I-V curve yields the smallest prediction error. Figure 2 shows the predicted I-V curve (green) and the measured I-V curve (dashed black) for the third approach (E3). With some exceptions, the prediction becomes more accurate with

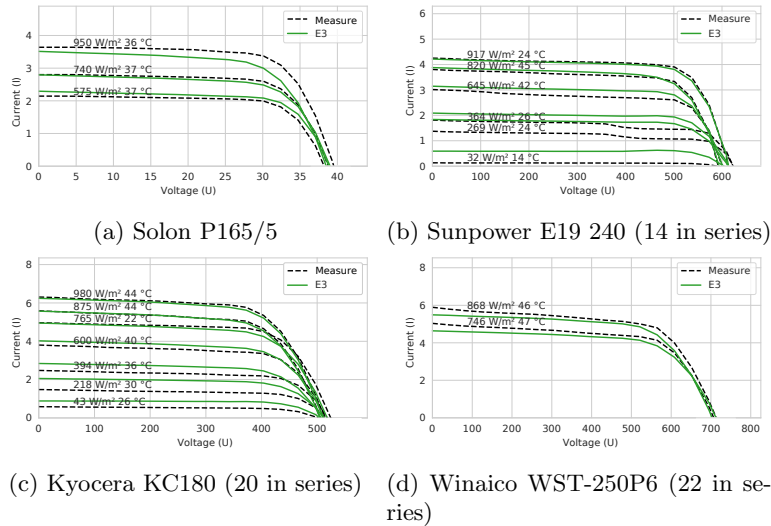


Fig. 2: Prediction examples for different string configurations of the 3. approach

increasing irradiation. Above 800 W/m^2 the maximum prediction error is below 10%, which is in the range of peak performance prediction of commercially available light I-V curve measurement devices [13].

5 Conclusion and Future Work

In summary we tested different pre-processing approaches in order to predict the light I-V curve using the dark I-V curve and an ANN. The experimental results shows that using interpolated points of the I-V curve yields better results than using a PCA or the electrical two diode model for feature extraction. We plan to further investigate recurrent neural networks with an increased number of I-V curve points to further improve the prediction accuracy.

Acknowledgements

This work has been developed in the project PVServ 2.0 (reference number: ZF4401205LT7) and is funded by the German ministry of economic and energy (BMWi) within the research programme ZIM 2018.

References

1. Beier, J., Voss, B.: Humps in dark i-v-curves-analysis and explanation. In: Conference Record of the Twenty Third IEEE Photovoltaic Specialists Conference - 1993 (Cat. No.93CH3283-9). pp. 321–326 (1993)
2. Bressan, M., El Basri, Y., Galeano, A., Alonso, C.: A shadow fault detection method based on the standard error analysis of iv curves. *Renewable energy* **99**, 1181–1190 (2016)
3. Fadhel, S., Delpha, C., Diallo, D., Bahri, I., Migan, A., Trabelsi, M., Mimouni, M.: Pv shading fault detection and classification based on iv curve using principal component analysis: Application to isolated pv system. *Solar Energy* **179**, 1–10 (2019)
4. Huang, J.M., Wai, R.J., Gao, W.: Newly-designed fault diagnostic method for solar photovoltaic generation system based on iv-curve measurement. *IEEE Access* **7**, 70919–70932 (2019)
5. Jones, C.B., Mart?nez-Ram?n, M., Smith, R., Carmignani, C.K., Lavrova, O., Robinson, C., Stein, J.S.: Automatic fault classification of photovoltaic strings based on an in situ iv characterization system and a gaussian process algorithm. In: 2016 IEEE 43rd Photovoltaic Specialists Conference (PVSC). pp. 1708–1713 (2016)
6. Kaminski, A., Marchand, J.J., Fave, A., Laugier, A.: New method of parameters extraction from dark i-v curve. In: Conference Record of the Twenty Sixth IEEE Photovoltaic Specialists Conference - 1997. pp. 203–206 (1997)
7. Kawamura, H., Naka, K., Yonekura, N., Yamanaka, S., Kawamura, H., Ohno, H., Naito, K.: Simulation of i-v characteristics of a pv module with shaded pv cells. *Solar Energy Materials and Solar Cells* **75**(3-4), 613–621 (2003)
8. King, D.L., Hansen, B.R., Kratochvil, J.A., Quintana, M.A.: Dark current-voltage measurements on photovoltaic modules as a diagnostic or manufacturing tool. In: Conference Record of the Twenty Sixth IEEE Photovoltaic Specialists Conference - 1997. pp. 1125–1128 (1997)
9. Mellit, A., Tina, G., Kalogirou, S.: Fault detection and diagnosis methods for photovoltaic systems: A review. *Renewable and Sustainable Energy Reviews* **91**, 1 – 17 (2018). <https://doi.org/https://doi.org/10.1016/j.rser.2018.03.062>, <http://www.sciencedirect.com/science/article/pii/S1364032118301370>

10. Mertens, K., Arnds, A., Diehl, M.: Quick and effective plant evaluation using dark-iv string curves. In: Proceedings of 33rd European Photovoltaic Solar Energy Conference, Amsterdam, 2017. pp. 2346 – 2348 (2017). <https://doi.org/10.4229/EUPVSEC201>
11. Mertens, K.: String-dunkelkennlinien: Eine neue effiziente methode zur anlagenevaluation, 33. symposium photovoltaische solarenergie, staffelstein, 27.04.2018 (2019)
12. Sarikh, S., Raoufi, M., Bennouna, A., Benlarabi, A., Ikken, B.: Fault diagnosis in a photovoltaic system through iv characteristics analysis. In: 2018 9th International Renewable Energy Congress (IREC). pp. 1–6. IEEE (2018)
13. Wagner, A.: Photovoltaik Engineering. Springer (2006)

Using Probabilistic Soft Logic to Improve Information Extraction in the Legal Domain

Birgit Kirsch¹, Sven Giesselbach¹, Timothée Schmude¹, Malte Völkening³,
Frauke Rostalski³, and Stefan Rüping^{1,2}

¹ Fraunhofer IAIS, Schloss Birlinghoven, Sankt Augustin
<name>.<surname>@iais.fraunhofer.de

² Fraunhofer Center for Machine Learning, Schloss Birlinghoven, Sankt Augustin

³ Institute for Criminal Law and Criminal Procedure, University of Cologne, Cologne

Abstract. Extracting information from court process documents to populate a knowledge base produces data valuable to legal faculties, publishers and law firms. A challenge lies in the fact that the relevant information is interdependent and structured by numerous semantic constraints of the legal domain. Ignoring these dependencies leads to inferior solutions. Hence, the objective of this paper is to demonstrate how the extraction pipeline can be improved by the use of probabilistic soft logic rules that reflect both legal and linguistic knowledge. We propose a probabilistic rule model for the overall extraction pipeline, which enables to both map dependencies between local extraction models and to integrate additional domain knowledge in the form of logical constraints. We evaluate the performance of the model on a German court sentences corpus.

Keywords: Information Extraction · Probabilistic Soft Logic · Legal-Tech

1 Introduction

In the year 2018 alone, there were approximately 870,000 court procedures in Germany⁴. All of them are documented in text, however to this date there are still only rudimentary solutions as to how to search for information within these documents. Transforming these documents into structured form produces data that can give insight into court processes and may be a valuable source to examine research questions such as the perceived imbalance in degrees of penalty between different regions in Germany, as described in Grundies (2018). According to the examinations of Grundies, there are substantial deviations (up to 15%) in how the same crime is punished in southern and northern Germany. The objective of this paper is to extract relevant information from these documents to populate a database that can be used for further analysis. While one

Copyright © 2020 by the paper's authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

⁴ https://www.destatis.de/DE/Themen/Staat/Justiz-Rechtspflege/_inhalt.html

might think of using standard natural language understanding approaches to tackle this task, the particular challenge lies in the fact that the relevant information in these texts is interdependent and structured by complex legal domain knowledge. For example, the probability of occurrence of a monetary fine in the court sentence obviously depends on the type of court sentence (imprisonment or fine). Ignoring these dependencies leads to inferior solutions for information extraction. Hence, we demonstrate how the extraction of information from legal texts can be improved by the use of probabilistic soft logic rules that reflect legal knowledge.

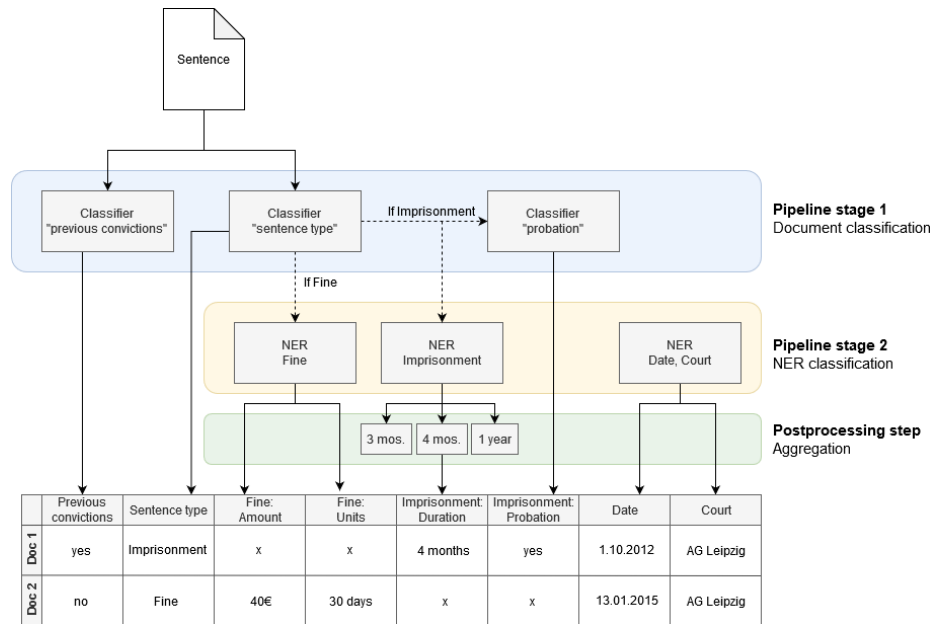


Fig. 1. Extraction process

Each document results in one entry in a knowledge base with seven facts about the document specific case and defendant, delineated in Section 3. Figure 1 displays the extraction pipeline to populate such a knowledge base and visualizes the dependencies between local components. It consists of the following two stages:

1. **Document Classification:** classification on document level predicts database entries with a fixed set of possible values, such as the type of court sentence (fine, imprisonment or acquittal).
2. **Named Entity Recognition (NER):** classification on token basis extracts all information with an unlimited set of possible values that have to be extracted from the document text itself, such as the duration of imprisonment. This will be treated as a named entity recognition task with the objective to assign a fixed set of classes to each token and extract the information

from the token text that is assigned the respective entity class. If a model assigns the same class to multiple text spans, they have to be consolidated to extract one value per document.

This baseline approach introduced for the use case raises multiple challenges: error propagation in the pipeline, value consolidation of NER predictions and the sparsity of training data.

Error propagation: One major disadvantage of this approach is that errors are propagated from the first classification component on document basis to the second classification component on token basis. This may lead to a lower performance with respect to the final aggregated document results. This problem is known from other NLP tasks such as knowledge base population. Recent work therefore focused on solving multiple tasks and modeling it as an end-to-end statistical inference problem (Sachan et al., 2018).

Value consolidation: One additional source of error is the consolidation of multiple token spans in the same document to extract one value per document. A NER classifier may in one document classify multiple token spans with the same class and as a consequence create multiple candidates that need to be consolidated to generate one valid value per NER class and document. Assigning the same NER class to multiple spans in one document is only valid when they refer to the same entity, e.g. the text span is similar. Integrating this as a hard constraint in a post-processing step may introduce additional errors.

Label sparsity: Since obtaining, anonymizing and labeling court sentence documents is a tedious process, the corpus available to train the pipeline classifiers is rather small. Beside the raw data there is valuable knowledge, both about dependencies between the information (e.g., a defendant with previous convictions is more likely to get imprisoned) and about the document structure (e.g., date and location of the court are likely to be in the beginning of the document). This sort of knowledge is not exploited in traditional approaches.

In order to address these challenges, we propose a probabilistic model constructed with a logical templating language that models a joint objective over the whole pipeline. By reasoning over all pipeline tasks jointly, we try to weaken the effect of propagated errors. By then modeling dependencies between NER candidates per document we address value consolidation challenges and with using a logical templating language, we enable to integrate additional background knowledge. Contributions of this paper can be summed up as the following:

- We propose a probabilistic model based on PSL for the overall extraction pipeline, which enables to map dependencies between local classification models.
- We model relevant concepts from the legal domain in the form of logical constraints and integrate them into the probabilistic model.
- We provide an empirical evaluation of the approach on a data set of court sentences and compare results with the traditional pipeline approach.

The remainder of this paper is organized as follows: Section 2 gives an overview of related work in the field of joint information extraction and probabilistic

pipelines, Section 3 describes our approach, Section 4 provides an experimental evaluation on a German sentencing corpus and reviews benchmarks in comparison with a traditional pipeline approach, Section 5 concludes with a summary and future work.

2 Related Work

Modeling pipelines in NLP and handling error propagation is a well known problem in multiple fields of research. Former work proposed diverse approaches to tackle this, e.g. by using graphical models or inductive logic programming. Marciniak and Strube (2005) introduced a model built upon linear programming for NLP pipelines of cascading classifiers and Roth and Yih (2002) use a bayesian belief network for joint prediction for entity and relation classification models. Singh et al. (2013) apply a joint graphical model for the tasks entity tagging and relation extraction including co-reference resolution to allow flow of uncertainty across task boundaries. Besides the mere modeling of multiple tasks, former work also focused on incorporating on additional domain knowledge. Pawar et al. (2017) introduce a neural model to address boundary identification, entity type classification and relation type classification jointly (AWP-NN) and show that refining the output with a Markov Logic Network to incorporate additional knowledge improves the results. Min et al. (2017) propose a probabilistic graphical model to extract facts from documents end-to-end to fill a knowledge base. Besides taking into account full corpus information for joint inference, they empirically show that integrating knowledge about entity and relation occurrences improves the results. This work is based on Sachan et al. (2018), who propose to use the statistical relational learning framework PSL to model the pipeline in a probabilistic way. To our knowledge we are the first to introduce a rule based probabilistic model for the aforementioned NLP extraction pipeline that is able to incorporate additional domain knowledge and apply and evaluate it on a legal corpus. The concept of automatically extracting information from court procedure documents was proposed and developed as a prototype in the Legal Tech Lab Cologne⁵ in early 2019 – an initiative, formed to find solutions for digitisation of legal procedures.

3 Approach

We propose a probabilistic model for an information extraction pipeline that jointly reasons over two pipeline stages: the **Document Classification Stage** and the **NER Stage**. This model is constructed using the logical templating language Probabilistic Soft Logic introduced by Bach et al. (2017). The following subsections explain both pipeline stages and the probabilistic model in detail.

⁵ <https://legaltechcologne.de/>

3.1 Document Classification Stage

In this stage, all defendant and case-related information such as the type of court sentence is extracted. The objective of this stage is to perform three separate classification tasks and assign the following categories to each document:

- Previous Convictions (PC) : Yes/No
- Type of court sentence⁶ (TS): Imprisonment/Fine/Acquittal
- Probation (PR): Yes/No

For each classification task, two separate model architectures are trained and applied, the inbuilt convolutional neural network (CNN) based model provided by *spaCy*⁷, as well as a transformer based classification model, BERT (Devlin et al., 2018), which we will shortly discuss in the following subsections. The labels for the classification are assigned per court sentence in the documents.

BERT Classifier: Devlin et al. (2018) introduce a language model based on transformer networks (Vaswani et al., 2017), which has been shown to yield state-of-the-art performance in many natural language understanding tasks. We use the small German BERT model integrated in the hugging-face library⁸. Since BERT is restricted by memory constraints in the amount of tokens it can process and the document classes are available on a sentence level in the training set, indicating whether a sentence mentions a previous conviction (yes, no), a probation (yes, no) and the type of sentence, we train BERT to classify sentences instead of document classes. We train a separate classification model for each of the aforementioned categories and introduce the class *Other* for sentences which do not contain any class annotation. The values are aggregated to document classes. Therefore, we first filter out sentences in which the model predicts *Other* as the most likely class. Out of all remaining sentences we return the class probabilities for the sentence with the highest confidence of the classifier, i.e. where the maximum probability is the highest. If no sentences remain, we pick the highest probability for each of the classes, normalize and return the tuple of the new class probabilities.

spaCy Classifier: This architecture is based on a CNN with mean pooling and a final feed-forward layer. The network is fed with pretrained word embeddings trained on the German Wikipedia and the German common crawl (Ortiz Suárez et al., 2019).⁹

⁶ Note that in the following it will be important to carefully distinguish between court sentences and sentences as a linguistic construct!

⁷ An open-source library for Natural Language Processing, <https://spacy.io/>

⁸ Model pretrained on the German Wikipedia, an online collection of legal court sentences and news texts, <https://huggingface.co/bert-base-german-cased>

⁹ <https://oscar-corpus.com/>

3.2 Named Entity Recognition (NER) Stage

In the NER stage all document-class specific information such as amount of the fine have to be extracted from the document. Objective of the NER stage is to assign one of the following classes to each token in the document: date of the court sentence (date), court location (loc), amount of the fine (f_amnt), number of day-fines (f_units)¹⁰ and duration of imprisonment (d_impr). Whether f_amnt and f_units or d_impr are present in a document depends on the type of court sentence (imprisonment or fine). For this classification task we rely on two architectures described below.

BERT NER: The named entity recognition model based on BERT has the same general transformer architecture as the classifier. Instead of predicting the class of sentences, a per-token classification takes place. We use a single feedforward layer with softmax activation to calculate the token labels.

spaCy NER: The spaCy NER model¹¹ utilizes a different architecture. Its embedding layer consists of two parts incorporating syntactic knowledge about the word, and a stack of 4 residual CNN layers to capture contextual information. Both layers are followed by a feed-forward network. Finally, a form of attention mechanism is used to incorporate additional information about the previous tokens and previous entity predictions. The final layer is a feed-forward and correction layer, which prevents illegal state shifts

3.3 Probabilistic Pipeline

We introduce a graphical model of a joint probability distribution to reason over all stage outputs at the same time. This model integrates both local stage model predictions, dependencies between the stages and additional background knowledge. For model construction we use the Statistical Relational Learning Framework PSL, introduced by Bach et al. (2017). It provides a First Order Logic templating language to define a joint probability distribution over a set of random variables. Therefore, rule templates are translated to a special type of Markov Random field, a *Hinge-Loss - Markov Random Field (HL-MRF)*. In this graphical model, each node represents a random variable and each edge a dependency between variables. Given a set of observed variables $X = (X_1, \dots, X_n)$, a set of random variables $Y = (Y_1, \dots, Y_{n'})$, a set of potential functions $\phi = (\phi_1, \dots, \phi_m)$ and a set of weights $\omega = (\omega_1, \dots, \omega_m)$, a *HL-MRF* represents the following probability density function over Y conditioned on X :

$$P(Y|X) = \frac{1}{Z(\omega, X)} \exp\left[-\sum_{j=1}^m \omega_j \phi_j(X, Y)\right] \quad (1)$$

¹⁰ In German criminal law, fines are calculated in day-fines. The number of day-fines depends on the severeness of the offense while the amount of each day-fine is based on the offender’s personal income.

¹¹ <https://spacy.io/universe/project/video-spacys-ner-model>

with Z as a normalization factor.

$$\phi_j(X, Y) = (\max\{l_j(X, Y), 0\})^{p_j} \quad (2)$$

with l_j representing a linear function and $p_j \in \{1, 2\}$.

Potential functions ϕ_j are defined per clique, a subset of fully connected nodes in the graph, assigning a probability mass to each clique state. A clique state is one assignment of values to all random variables participating in the clique. Assignment of a higher value to one clique state means that this state will be interpreted as being more likely. In *PSL*, these potential functions are generated using logical first order rules, such as:

$$w : \text{Friends}(A, B) \wedge \text{Friends}(B, C) \Rightarrow \text{Friends}(A, C) \quad (3)$$

A weight w is assigned to each rule and indicates its importance. *Friends* is called a *predicate*. *Predicates* can take one to multiple arguments, such as A and B . Both A and B are *variables* and serve as placeholders. They can be substituted by concrete instances, referred to as *constants*. A *PSL*-model itself consists of a set of these template rules and a weight for each. Substituting all variables in the rule template set with their respective constants is called grounding. Every grounded rule represents one clique in the underlying graph structure and every grounded predicate is an observed or unobserved random variable mapped to a node in the clique. The weight of a grounded rule determines the weight ω_j of a potential function. A potential function assignment denotes the degree to which a rule is satisfied. Clique states that lead to satisfying a rule will be assigned a higher value than clique states that lead to the violation of a rule. Intuitively spoken, assignments of Y are more likely the fewer rules they violate.

Our proposed rule set can be categorized into four rule types: *basic_rules*, *domain_rules*, *pipeline_rules* and *similarity_rules*, examined in the following subsections. Table 1 provides a detailed explanation for all elements participating in each rule.

Basic Rule Set: The basic set models the relationship between the local classification models (reflected by the observed predicate $f_cls_pc^{(s1)}$ for stage one and $f_cls^{(s2)}$ by stage two respectively) and the true unobserved stage prediction ($cls_pc^{(s1)}$ for stage one and $cls^{(s2)}$ for stage two).

$$f_cls_pc^{(s1)}(d, m, c_pc) \wedge T(m, c_pc) \Rightarrow cls_pc^{(s1)}(d, c_pc)^2 \quad (4)$$

$$!f_cls_pc^{(s1)}(d, m, c_pc) \wedge T(m, c_pc) \Rightarrow !cls_pc^{(s1)}(d, c_pc)^2 \quad (5)$$

$$f_cls^{(s2)}(d, z, m, c) \wedge T(m, c) \Rightarrow cls^{(s2)}(d, z, c)^2 \quad (6)$$

$$!f_cls^{(s2)}(d, z, m, c) \wedge T(m, c) \Rightarrow !cls^{(s2)}(d, z, c)^2 \quad (7)$$

$$cls_pc^{(s1)}(d, +c_pc) = 1. \quad (8)$$

$$!cls^{(s2)}(d, z, c) \quad (9)$$

Domain	
$d = \{1, \dots, D\}$	with D as the number of documents in the corpus
$m = \{SPACY, BERT\}$	local stage model
$z = \{1, \dots, Z\}$	with Z as the number of token candidates
$c_pc = \{yes, no\}$	class types predicted by <i>PreviousConviction</i> classifier
$c_st = \{Fine, Imprisonm., Acq.\}$	class types predicted by <i>courtsentencetype</i> classifier
$c_pr = \{yes, no, other\}$	class types predicted by <i>probation</i> classifier
$c = \{date, \dots, loc\}$	class types predicted by <i>NER</i> classifier
Observed Predicates	
$f_cls_pc^{(s1)}(d, m, c_pc)$	prediction output of the local <i>PreviousConviction</i> classifier model m for document d and class c_pc
$f_cls_st^{(s1)}(d, m, c_st)$	prediction output of the local <i>courtsentencetype</i> classifier model m for document d and class c_st
$f_cls_pr^{(s1)}(d, m, c_pr)$	prediction output of the local <i>probation</i> classifier model m for document d and class c_pr
$f_cls^{(s2)}(d, z, m, c)$	prediction output of the local model m for document d , token candidate z and class c
$CLOSE(d, z1, z2)$	assigned 1, when there are at most 20 token between $z1$ and $z2$, 0 otherwise
$SIM(d, z1, z2)$	assigned 1, when the text $z1$ and $z2$ span over is equal, 0 otherwise
Unobserved Predicates	
$cls_pc^{(s1)}(d, c_pc)$	global prediction of stage one for document d and document class type c_pc
$cls_ts^{(s1)}(d, c_ts)$	global prediction of stage one for document d and document class type c_ts
$cls_pr^{(s1)}(d, c_pr)$	global prediction of stage one for document d and document class type c_pr
$cls^{(s2)}(d, z, c)$	global prediction of stage two for document d , candidate z and <i>NER</i> type c

Table 1. Variables and predicates participating in the rule model.

Rules 4 and 5 display the rules for the local classification models from stage one for *PreviousConvictions*. $T(m, c)$ is a trust score introduced by Sachan et al. (2018) that denotes the trustworthiness of a local model m when predicting a class c . Intuitively spoken, Rule 4 encodes that when a local model m predicts a class c_pc for a document d and the model is trustworthy, then the predicted unobserved class is more likely to be c_pc . A similar rule set exists for the class types *Typeofsentence* and *Probation*. Rules 6 and 7 show the rules for the local *NER* classification models from stage two respectively. Rules 8 and 9 model the prior beliefs that the prediction probability for all class types of the *PC*-classifier sum up to one and that a token candidate z is not assigned the *NER*-class type.

Figure 2 displays a simplified model of the grounded observed (grey nodes) and unobserved predicates (white nodes) according to the above rule sets. The model is grounded only for one class ($c1_pv$, $c1_pr$) of the document classifiers *pc* and *pr*, for one document $d1$, two local stage one models $m1$ and $m2$, two local stage two models $m3$ and $m4$ to predict one class $c3$ and two *NER* candidates

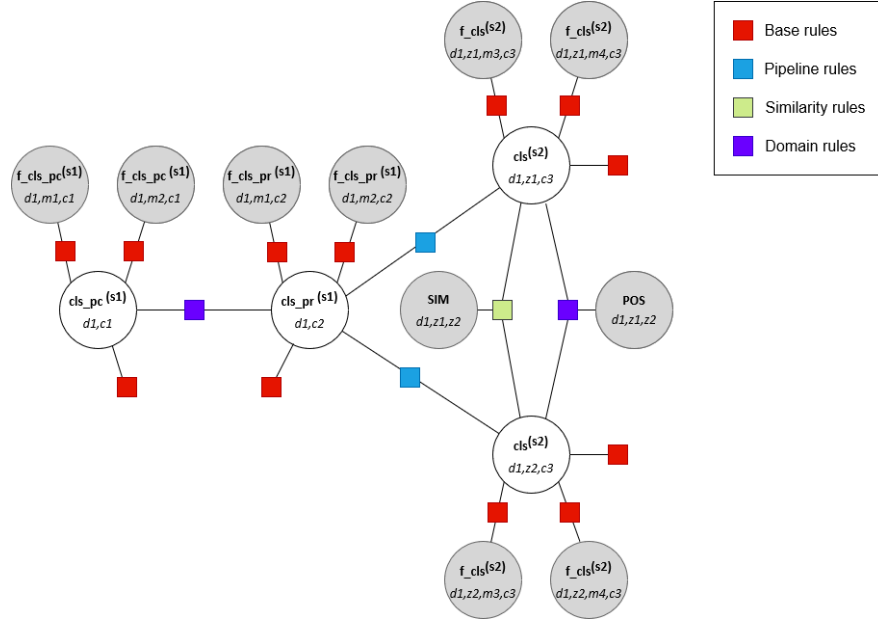


Fig. 2. Simplified grounding for rule model

$z1$ and $z2$ for which class $c3$ should be predicted. Potential functions resulting from the baseline rules are presented as red squares.

Pipeline Rule Set: This rule set incorporates relationships between prediction outputs of the document and NER classification stages and relationships between local models from one stage. Rule 10 for example expresses the dependency between the st classifier and the NER classifier. Whenever the predicted value for court sentence type is not "fine", it is unlikely that the information about the amount of the fine is present in the document

Rule 11 models the dependency between the PR and the ST classifiers for of the first stage.

$$\neg cls_st^{(s1)}(d, 'fine') \Rightarrow \neg cls^{(s2)}(d, z, 'f_amount')^2 \quad (10)$$

$$cls_pr^{(s1)}(d, 'no') \Rightarrow cls_st^{(s1)}(d, 'imprisonment')^2 \quad (11)$$

Domain Rule Set: The domain rules permit to incorporate knowledge about the document structure (e.g. in Rule 12) and class dependencies such as that defendants with previous convictions are more likely to be sentenced to imprisonment or that the height of the fine and the number of day-fines are most likely mentioned in the same paragraph (see Rule 13).

$$cls_pc^{(s1)}(d, 'yes') \Rightarrow cls_st^{(s1)}(d, 'imprisonment')^2 \quad (12)$$

$$cls^{(s2)}(d, z1, 'f_amnt') \wedge \neg CLOSE(d, z1, z2) \Rightarrow cls^{(s2)}(d, z1, 'f_units')^2 \quad (13)$$

Similarity Rule Set: The similarity rules model the relationship between multiple mentions of the same entity in one document. Rule 14 denotes that when two candidates in one document span over similar strings, they are more likely to be assigned the same NER type. This ensures that when one information like the date of the court sentence is mentioned multiple times in the document it is always assigned the same NER class type and when two candidates don’t have equal content, they are unlikely to refer to the same entity (Rule 15).

$$cls^{(s2)}(d, z1, c) \wedge SIM(d, z1, z2) \Rightarrow cls^{(s2)}(d, z2, c)^2 \quad (14)$$

$$cls^{(s2)}(d, z1, c) \wedge !SIM(d, z1, z2) \Rightarrow !cls^{(s2)}(d, z2, c)^2 \quad (15)$$

4 Evaluation

Data: We evaluate the performance of the model introduced in Section 3 using a corpus with 146 German court sentences¹². Each document describes an offence (theft), where the defendant is either sentenced to imprisonment or to a fine, and which is only related to one case and one defendant. The documents are split into training, validation and test set.

PSL Pipeline Model: For the probabilistic model we train local stage classifiers for document and token classification using the architectures described in Section 3. All classifiers are trained on the training set and applied to the validation and test set. We use the validation set to estimate the trustworthiness scores (Rules 4, 5, 6, 7) and to refine weights. To evaluate the performance of the rule model and its subsets, we do not provide an evaluation for the local stage tasks (document and token classification), but focus on the amount of correctly extracted values per document. We perform inference for the unobserved variables listed in Table 1 to predict all layer outputs at the same time. As proposed in Bach et al. (2017), we use the consensus optimization approach Alternating Direction Method of Multipliers (ADMM).

Basic Pipeline Model: We compare the proposed solution to the pipeline approach visualized in Figure 1, where all local models are trained using the BERT architecture (see Section 3). Since the models that were trained using the architecture of spaCy had a poor performance on the test set, the performance of the pipeline model based on these classifiers is not evaluated here. The local document classification models are trained on both training and validation set and are applied to the test set to predict document class labels. The local token classification models are trained both on training and validation set and are then applied to the test set. The *NER* classification results per document are consolidated by selecting the candidate with the highest probability score predicted by the model for each *NER* class. We model dependencies between the

¹² manually annotated by the University of Cologne

predicted type of sentence and the *NER* prediction as hard constraints. When the predicted class is "Fine", prediction for "f_amnt" and "f_units" are set to "NaN" (for document class "Imprisonment", prediction for "d_impr" is set to "NaN" respectively). We additionally provide results for the extraction workflow where this stage dependency is not modeled and contradicting results are not removed. Table 2 provides a comparison of model performances. The first seven entries denote the performance of the rule models consisting of different rule subset: basic (basic rules), sim (similarity rules), pipe (pipeline rules) and domain (domain rules). The scores per extracted information type reflect the amount of correctly extracted values on the test corpus.

Model	Date	Loc	f_amnt	f_units	d_impr	pc	st	pr	avg.
<i>PSL: basic sim domain</i>	0.783	0.913	0.826	0.826	0.870	0.870	0.913	0.783	0.848
<i>PSL: basic sim</i>	0.783	0.913	0.826	0.826	0.870	0.826	0.957	0.783	0.848
<i>PSL: basic domain</i>	0.783	0.913	0.826	0.870	0.826	0.870	0.913	0.783	0.848
<i>PSL: basic pipe</i>	0.783	0.913	0.826	0.783	0.957	0.826	0.913	0.913	0.864
<i>PSL: basic sim pipe</i>	0.783	0.913	0.826	0.783	0.957	0.826	0.913	0.913	0.864
<i>PSL: basic sim pipe domain</i>	0.783	0.913	0.870	0.870	0.913	0.913	0.913	0.913	0.886
<i>PSL: basic pipe domain</i>	0.783	0.913	0.870	0.870	0.913	0.913	0.913	0.913	0.886
<i>BERT: constraints</i>	0.826	0.870	0.870	0.826	0.957	0.826	0.957	0.783	0.864
<i>BERT: no constraints</i>	0.826	0.870	0.826	0.783	0.957	0.826	0.957	0.783	0.853

Table 2. Extraction performance of rule models

The two PSL models achieving the highest overall matching rate both contain basic, pipeline and domain rules. Modeling relations between *NER* candidates by integrating the similarity rules does not seem to have an influence on the performance in this experiment. On average over all information types, the highest scoring PSL model outperforms both BERT benchmark models. For some particular types (*Date*, *d_impr* and *st*) the BERT benchmark model achieves a higher matching rate. These results indicate that modeling the pipeline in a probabilistic way and integrating domain knowledge improves the overall performance. Since the data set is very small, further analysis on a bigger corpus and a more detailed analysis of the rule model effects are required.

5 Summary

This paper proposes a rule based probabilistic model to improve the extraction pipeline for information extraction from court sentence documents. The model enables to both map dependencies between local extraction components and to integrate additional domain knowledge in the form of logical constraints. We evaluate the performance of the model on a German court sentences corpus and show that the model improves results compared to a BERT benchmark model.

6 Acknowledgements

This research has been funded by the Federal Ministry of Education and Research of Germany as part of the competence center for machine learning ML2R (01IS18038B)

Bibliography

- Bach, S.H., Broecheler, M., Huang, B., Getoor, L.: Hinge-loss markov random fields and probabilistic soft logic. *J. Mach. Learn. Res.* 18(1), 3846–3912 (2017)
- Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR* abs/1810.04805 (2018)
- Grundies, V.: Regionale unterschiede in der gerichtlichen sanktionspraxis in der bundesrepublik deutschland. eine empirische analyse. In: *Kriminalsoziologie. Handbuch für Wissenschaft und Praxis*. pp. 295–316. Baden-Baden, Germany (2018)
- Marciniak, T., Strube, M.: Beyond the pipeline: Discrete optimization in NLP. In: *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*. pp. 136–143. Association for Computational Linguistics, Ann Arbor, Michigan (Jun 2005)
- Min, B., Freedman, M., Meltzer, T.: Probabilistic Inference for Cold Start Knowledge Base Population with Prior World Knowledge 1, 601–612 (2017)
- Ortiz Suárez, P.J., Sagot, B., Romary, L.: Asynchronous Pipeline for Processing Huge Corpora on Medium to Low Resource Infrastructures. In: Bański, P., Barbaresi, A., Biber, H., Breiteneder, E., Clematide, S., Kupietz, M., Lungen, H., Iliadi, C. (eds.) *7th Workshop on the Challenges in the Management of Large Corpora (CMLC-7)*. Leibniz-Institut für Deutsche Sprache, Cardiff, United Kingdom (Jul 2019), <https://hal.inria.fr/hal-02148693>
- Pawar, S., Bhattacharyya, P., Palshikar, G.: End-to-end relation extraction using neural networks and Markov logic networks. In: *Proceedings of the 15th Conference of EACL: Volume 1, Long Papers*. pp. 818–827. Association for Computational Linguistics, Valencia, Spain (Apr 2017)
- Roth, D., Yih, W.t.: Probabilistic reasoning for entity & relation recognition. In: *COLING 2002: 19th Intern. Conf. on Computational Linguistics* (2002)
- Sachan, M., Dubey, K.A., Mitchell, T.M., Roth, D., Xing, E.P.: Learning pipelines with limited data and domain knowledge: A study in parsing physics problems. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) *Advances in NIPS 31*, pp. 140–151. Curran Associates, Inc. (2018)
- Singh, S., Riedel, S., Martin, B., Zheng, J., McCallum, A.: Joint inference of entities, relations, and coreference. In: *Proceedings of the 2013 Workshop on AKBC*. p. 1–6. AKBC '13, Association for Computing Machinery, New York, NY, USA (2013), <https://doi.org/10.1145/2509558.2509559>
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) *Advances in NIPS 30*, pp. 5998–6008. Curran Associates, Inc. (2017)

Fusing Multi-label Classification and Semantic Tagging

Jörg Kindermann^{1,2} and Katharina Beckh^{1,2}

¹ Fraunhofer IAIS, Sankt Augustin, Germany

² Competence Center for Machine Learning Rhine-Ruhr
{joerg.kindermann,katharina.beckh}@iais.fraunhofer.de

Abstract. Companies have an increasing demand for enriching documents with metadata. In an applied setting, we present a three-part workflow for the combination of multi-label classification and semantic tagging using a collection of key-phrases. The workflow is illustrated on the basis of patent abstracts with the CPC scheme. The key-phrases are drawn from a training set collection of documents without manual interaction. The union of CPC labels and key-phrases provides a label set on which a multi-label classifier model is generated by supervised training. We show learning curves for both key-phrases and classification categories, and a semantic graph generated from cosine similarities. We conclude that, given sufficient training data, the number of label categories is highly scalable.

Keywords: multi-label classification · semantic tagging · prediction-based embedding spaces · patents.

1 Introduction

For strategic developments, businesses and research organizations have an interest in identifying competences or trends in their respective organization and in comparison to competing institutions. Extracting this information manually among heterogeneous data is time-consuming which is partly complicated by different underlying classification schemes, e.g. from patents or publications. Therefore, there is an increasing demand for metadata [8] that combines categories from classification schemes with semantic tags.

The automatic single-label classification of documents is well-researched [21] [1] while multi-label classification with large numbers of labels still is a challenge [16]. The combination of classification and semantic tagging is also less explored. Advances in the distributed representation of words have provided the necessary basis for this combination [14] and recent work allows to achieve both steps together in a document processing workflow [18].

To tackle the fusion of classification and semantic tagging in an applied setting, we introduce a basis workflow which allows to classify and tag documents

Copyright © 2020 by the paper's authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

at once. For that we start by introducing the tools, namely the model, data and evaluation metrics (Section 3). Subsequently, we put the approach into context by describing a use case within the Fraunhofer society that aims to extract information from existing data sources (Section 4.1). As patent data is an important base for innovation research and because it exhibits one of the largest and prominent classification schemes, we employ it to demonstrate the workings of our approach.

Following the use case, we describe the three-part workflow in detail (Section 4.2). A set of key-phrases is collected in an unsupervised procedure from a training set of documents. The union of category labels and key-phrases provides a label set on which a multi-label classifier model is trained. Following the model training, we furthermore describe how to extract embedding vectors to visually represent classification categories and key-phrases together in a semantic graph. We depict learning curves with appropriate metrics and a cutout of the semantic graph. We conclude that the workflow scales to a larger amount of documents and can be applied on documents in various domains.

2 Related Work

Multi-label classification with a large number of categories has been notoriously difficult. A first break-through that made classification of texts possible without relying on manually designed features was the Support Vector Machine [5], [10]. However, the computational effort grows considerably with the number of labels, making the training of classification problems with thousands of labels intractable. Semantic tagging, i.e. the assignment of key-phrases to a text, in an unsupervised way was achieved by applications of the Latent Dirichlet Allocation topic model [3].

Both steps, multi-label classification and semantic tagging, in a document processing workflow could recently be combined with the advent of the *StarSpace* algorithm [18] based on embedding vector spaces. This algorithm implements the concept of prediction-based embedding spaces.

Since Elman’s seminal paper [7] on recurrent neural networks and their training on sequences, in particular sentences as sequences of words, there have been many efforts to improve the storage capacity and reduce the computational complexity of such systems. The *Word2Vec* algorithms [14] were a path-breaking invention in this direction which for the first time made it possible to represent semantic properties of words derived from their actual usage in large quantities of texts. This algorithm exceeded capacities of systems known so far by orders of magnitude. Levy and Goldberg [12] showed that the *Word2Vec* algorithms are closely related to counting-based vector representations by matrix-factorization mappings. An example is a vector-space based on *PMI* (point-wise mutual information) values. This finding supports confidence in the semantic properties of prediction-based embedding spaces, such as the *StarSpace* model, which are explored by cosine similarity. This is due to their close relationship to *PMI*-based

representations. Important follow-up developments of Word2Vec were Glove [15] and FastText [4].

Recent applications of StarSpace have been published in the areas of ontologies [9] and knowledge graphs [20] that are related to our use case. Regarding other recent work, transformer-based architectures [6] are also suitable for multi-label classification.

3 Methods

3.1 StarSpace

We chose StarSpace [18], a general-purpose neural embedding model which can be used for multi-label classification and tagging. It is based on a *bag of entities* representation. *Entities* can be texts, labels, meta-data like authors, source URLs etc. Starspace thus is capable of learning relations between items of various types and origins. The bag of entities representation is a high dimensional vector in an embedding space which may include labels. The actual learning algorithm is a stochastic gradient descent optimization of a special loss function

$$\sum_{(a,b) \in E^+, b^- \in E^-} L^{batch}(sim(a, b), sim(a, b_1^-), \dots, sim(a, b_k^-)) \quad (1)$$

where entities a and b are drawn from the set E^+ of positive examples, and entities b^- are drawn from the set E^- of negative examples. In our use case (section 4.1) the entities are the patent abstracts and their labels and key-phrases. The k -negative sampling strategy of [14] is used. The similarity function can be chosen from $\{cosine, dot\ product\}$. The loss function L_{batch} has two implementations:

- margin ranking loss: $\max(0, \mu - sim(a, b))$ with margin parameter μ
- the negative log loss of the softmax function: $-\log\left(\frac{e^{y_i}}{\sum_j e^{y_j}}\right)$

During the optimization run, the similarity function $sim(\Delta, \Delta)$ is "learned". It can subsequently be used to measure the similarity between *entities*. For classification, a label is predicted for a given input a as $\max_{\hat{b}}(sim(a, \hat{b}))$ over the set of possible labels \hat{b} . This feature can be used to output a ranking of labels according to their similarity, implementing multi-label classification.

3.2 Data

In our experiments, we employ a sample of patent abstracts from the United States Patent and Trademark Office (USPTO)³ from the month of January 2020 which amounts to 22.000 abstracts. The classification scheme that we use is the Cooperative Patent Classification (CPC). The CPC hierarchy is illustrated in Fig.1 and consists of *section, class, subclass, maingroup* and *subgroup*.

³ <https://developer.uspto.gov/product/patent-grant-full-text-dataxml>

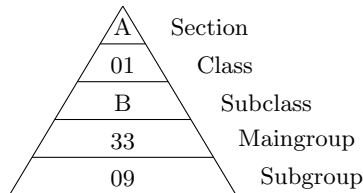


Fig. 1: CPC hierarchy illustrated with an example category

Label	Quantity
Main-CPC	100
Further-CPC	250
Key-phrases	200

Table 1: Number of labels per category

We focus on the first three levels, namely section, class and subclass. The data contains a *Main-CPC* which serves as the main category of the patent and *Further-CPC* categories which are also applicable categories (see Fig. 5(b) for examples). We selected a subset of all possible labels with respect to the number of examples available in our data collection. Table 1 shows the numbers of selected labels in both categories. For the category *key-phrases* see section 4.2.

3.3 Evaluation Metrics

We evaluated the experiments based on two metrics:

- **F1 value** is the well-known harmonic mean of *precision* and *recall* measures. We used the F1 value to assess the performance on the Main-CPC labels, because it is suited to evaluate single-label classification tasks mainly.
- **Coverage-rank** [19] with a real-valued ranking function $f(.,.)$

$$\text{coverage}(f) = \frac{1}{p} \sum_i \max_{y \in Y} \text{rank}_f(x_i, y) - 1 \quad (2)$$

counts how many steps have to be taken to move down the ranked label list to cover all the relevant labels of the example. The coverage-rank was used to assess the performance on the Further-CPC labels and key-phrases. It seems to be more adequate to multi-label classification than the F1 value. Another important reason is that we want to train the model on a semantic tagging task, which would be thwarted by an exclusive optimization according to F1 values. The reason is that semantic tagging is expected to tag documents with a certain key-phrase that is not literally contained in the document but is nevertheless highly relevant to the document content and topic. This desired behavior would, however, result in a degraded F1 value because it would be counted as a false positive.

4 Experiments

4.1 Use Case

Here, we first describe the applied benefit of our approach in the context of a current project. Within the project "Fraunhofer Digital" a data hub has been

created which will cover a variety of datasets, ranging from publications and patents to project descriptions. All the datasets contain valuable information about the competence landscape and, in particular, patent data is important for the strategic technology and innovation management within Fraunhofer.

One key challenge is that patents are only mapped to a patent classification system. There is no basis in linking the classification to information outside of the scheme. In this use case it is desired to find similarities between patents and at a glance we want to identify the most suitable key-phrases. This makes it for example easier to determine current technologies and technology trends.

Our approach is to extract and assign information inherent in the patents that exceeds the common patent classification. We achieve this by employing key-phrase extraction. By providing key-phrases on top of the classification, the model provides comprehensible information for readers and therefore serves as a base to facilitate work for employees. In the "Fraunhofer Digital" use case we apply this approach also to publication data using more data to create several classification models. For this paper, we narrow our focus to patent samples. In the following, we describe the workflow in more detail.

4.2 Workflows

Key-phrase Extraction. We collect a list of key-phrases from the pool of training documents using the *RAKE* (Rapid Automatic Keyword Extraction) algorithm [17]. We chose RAKE, because it does not depend on sophisticated preprocessing operations as named-entity recognition and training of neural networks as in [13]. RAKE operates in an unsupervised manner on individual documents. It identifies key-phrases by extracting phrases between stopwords (e.g. "the", "a") and by analyzing the frequency of word appearance and word co-occurrence.

Because RAKE works on single documents, the frequent extraction of non-informative standard key-phrases like section headings ("Related Work", etc.) is expected. It can be avoided by detecting and eliminating those phrases based on an information-theoretic measure like TF-IDF (Term Frequency - Inverse Document Frequency) [2] or Importance Weight [11]: We chose TF-IDF and keep only those phrases which contain at least one term with a value above a certain threshold (to be set as a hyper-parameter). The resulting list usually is still too large. Therefore, we select the n most frequent phrases. In the experiment described here, we chose 200 key-phrases (see Table 1). Examples from this set of key-phrases are "search engine" or "application programming interface" and more are depicted in Fig. 5. The selected key-phrases define the gold-standard for F1 value optimization.

Model Training. The key-phrases together with the Main-CPC and Further-CPC labels define the set of StarSpace labels to be trained (see Fig. 5(b) for examples). Taking the abstracts and the labels, the StarSpace model is trained (Fig. 2 top) with a pre-determined number of iterations on the training set. From

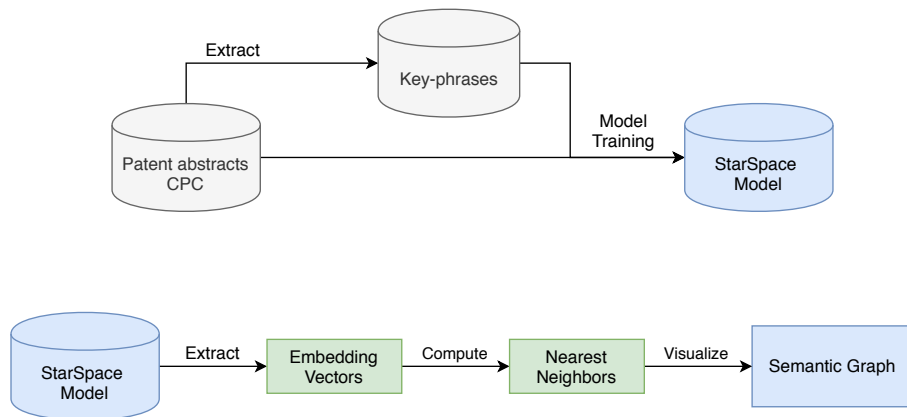


Fig. 2: The training workflows. The top shows key-phrase extraction and the bottom illustrates the construction of a semantic graph

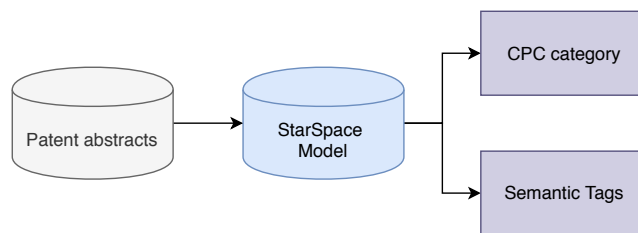


Fig. 3: The prediction workflow. Patent abstracts are fed into the StarSpace model which computes CPC categories and tags

the trained model we export the embedding vectors of the labels and construct a semantic graph that represents the cosine-similarity based k -nearest-neighbor relations of the labels (Fig. 2 bottom). This graph serves as a human-readable quality reference of the model. It is not directly used for the prediction workflow.

To optimize hyper-parameters we used a fixed training dataset of $\sim 13,000$ documents and a test set of $\sim 8,800$ documents (60%/40% split). We evaluated model performances for the CPC scheme from level 1 *Section* to level 4 *Main-group* (see Fig. 1). Results are reported exclusively for level 3 *Subclass*, because this was the most detailed level for which we could achieve satisfactory results.

The StarSpace algorithm has several hyper-parameters⁴ which need to be explored in separate evaluations. We optimized 9 of them (see Table 2).

StarSpace param.	Description	Explanation
iterations	number of training iterations	an iteration includes n minibatches
minCount	min frequency of terms	less frequent terms are eliminated
ngrams	ngrams of terms	ngrams up to n terms
dim	embedding dimension	the dimension of embedding vectors
lr	learning rate	learning rates are set to ≤ 0.05
batchSize	batch size	number of items in a minibatch
loss	loss function	the functions <i>hinge</i> (i.e. margin ranking) or <i>softmax</i>
similarity	similarity measure	<i>cosine similarity</i> or <i>dot product</i> of embedding vectors
adagrad	stochastic gradient optimizer	adagrad can be switched on or off

Table 2: Description of hyperparameters that we optimized

Model Prediction. New documents (without CPC-label) are assigned their CPC-labels and key-phrases by the trained StarSpace model (see Fig. 3). For each test document the model outputs a weight for each of the labels. Therefore, we need another hyper-parameter *weight-threshold* to cut-off the list of output labels sorted decreasingly by weight to achieve adequate F1 values.

4.3 Results

Attainable Model Performance Figure 4 shows a typical development of F1 and coverage-rank values during a training run of 640 iterations, a weight-

⁴ see <https://github.com/facebookresearch/StarSpace>

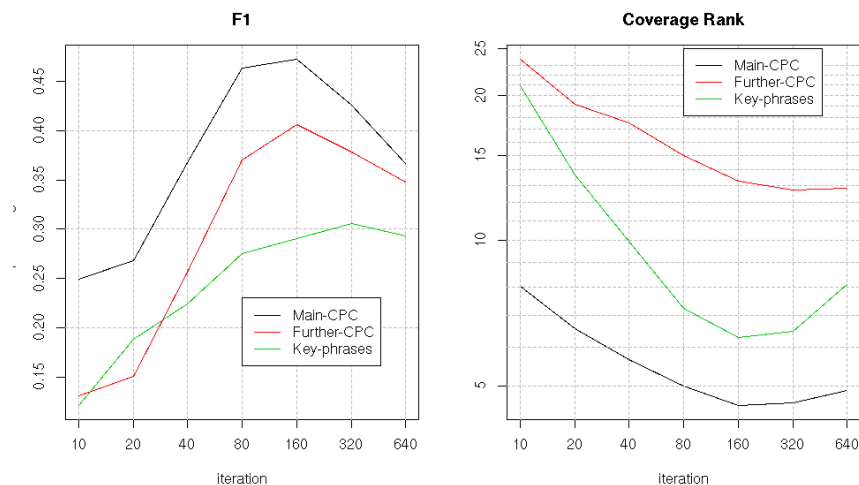


Fig. 4: Example illustration of learning curves of F1 value and coverage-rank for Main-CPC (black), Further-CPC (red), and key-phrase (green) labels.

threshold of 0.35 and otherwise optimal StarSpace parameters. We see that optimal values of F1 and coverage-rank occur in the same range of iterations. Note that large F1 values but small coverage-rank values are better. The overall F1 values are not very competitive. This is partly due to the limited number of documents we use. Moreover, optimizing the F1 value is only a secondary goal. It only makes sense for the Main-CPC values, because they are single-label categories. For the Further-CPC labels and a fortiori for the key-phrases we cannot define the F1 measure in a fully consistent way. This would require a predefined ordering on the multi-label categories which is not given. After all, the behavior of the different label sets is as expected: the single-label Main-CPC categories show better performance with respect to F1 compared to the multi-label categories Further-CPC and key-phrases.

The more important evaluation criterion is the coverage-rank, because it gives an estimate on the precision of the output of non-sorted multi-labels. Here we see the Main-CPC labels again performing best, as expected. The second-best performance of key-phrases and the rather large distance of the Further-CPC values to the other two cases is not expected and needs an explanation: All Further-CPC labels are drawn from the same category system as the Main-CPC labels. The most relevant of them is the Main-CPC label, and all others are Further-CPC labels. The sequence of CPC categories may thus be different for thematically closely related patent abstracts and result in different Main/Further-CPC label sets. This seems to be more difficult to learn for a model than categorizations from disjoint label sets. The fact that we have more Further-CPC labels than keywords may also add to the performance differences.

Semantic Tagging A trained StarSpace model contains exportable embedding vectors for both the terms occurring in the training documents **and** all category labels. This allows to define a k -nearest-neighbor relation on the labels with the cosine-similarity of their embedding vectors. A similar relation exists between the label embeddings and document texts based on the *bag-of-ngrams* representation of the documents⁵. This allows to assign k -nearest-neighbor key-phrase labels as semantic tags to documents. It is difficult to rate the appropriateness of such tagging directly. We therefore display a k -nearest-neighbor graph of labels from all three categories in Fig. 5.

This sub-graph is centered around the Main-CPC level 3 category "G06F - electric digital data processing" and shows the neighboring color-coded Main-CPC (red), Further-CPC (light blue) and key-phrase (cyan) labels⁶. The complete graph contains all 550 labels as nodes. The directed edges in the graph code the cosine similarity between the label embeddings. More similar labels are connected by stronger edges. Note that the linear distance of labels in this graph therefore is **not** an indicator of their embedding similarity. The edge color is set by its source label. In particular, we can observe that the Main-CPC labels and the Further-CPC labels of identical categories (for example G06F) are connected strongly vice-versa, as one would expect.

Semantic tagging now works as follows: if a document is classified, for example as M.G06F, it gets assigned the Further-CPC labels G06F and H04L, as well as the key-phrases "search engine", "client system", "operating system", "computer processor" and possibly more key-phrases that are not displayed in this graph cutout. This tagging behavior is a major difference from other tagging algorithms in that it may assign key-phrases to a document that are **not** contained in the document itself.

4.4 Limitations and Recommendations

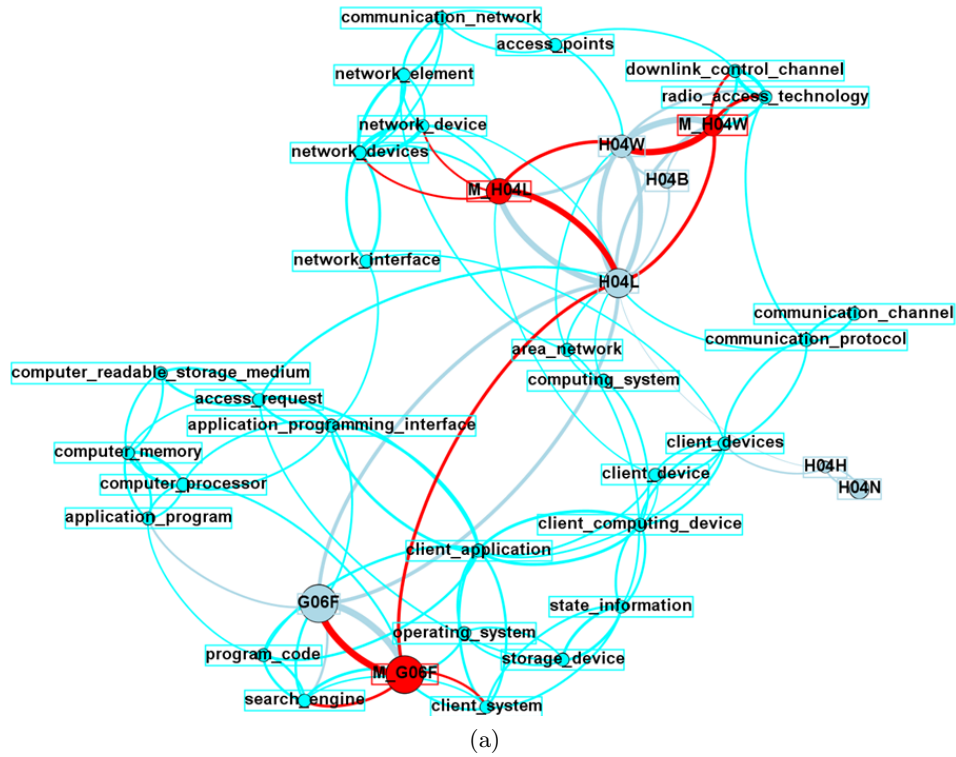
The classification and tagging workflow presented here has some intrinsic limitations which we will shortly discuss in this section.

- **Specificity of key-phrases:** We advise to investigate the specificity of the key-phrases that are extracted by the RAKE algorithm followed by TF-IDF filtering. Depending on the particular properties of a training collection, many of the key-phrases may occur in a large number of multi-label categories. It is up to the experimenter to create a mix of more frequent and more specific key-phrases if required.
- **Number of labels:** Though scalable in a large range there surely exist upper limits of the number of labels in a multi-label classification regime. These limits are related to the number of documents in the training set, but also to the skewedness of label distributions. We did not run quantitative investigations on this topic but from our general experience with StarSpace

⁵ For details see <https://github.com/facebookresearch/StarSpace>

⁶ For details see

<https://www.cooperativepatentclassification.org/cpcSchemeAndDefinitions/table>



CPC category	Description
G06F	Electric digital data processing
H04	Electric communication technique
H04B	Transmission
H04H	Broadcast communication
H04L	Transmission of digital information
H04N	Pictorial communication
H04W	Wireless communication networks

(b)

Fig. 5: (a) Semantic graph generated from cosine similarities of labels and key-phrases. Main-CPC is illustrated in red, Further-CPC in light blue and key-phrases in cyan. (b) The CPC categories and their description.

models in several domains we would state the following: The number of labels should not exceed 1-2% of the number of training data, and with respect to skewedness of distribution the frequency ratio of the least frequent and the most frequent label should not exceed 0.01. One way to circumvent the limit of label numbers would be to split labels into subsets and train several StarSpace models, one on each subset. Doing this, one has to take into account that the label weights in the model output cannot be compared across models. Therefore it makes sense to define subsets accordingly - for example *category labels*, *frequent key-phrases*, and *specific key-phrases*.

- **Model and processing resources:** StarSpace models can be very large with large numbers of training data and large n for the *ngram* parameter. Model sizes of more than 10GB are common, which also require corresponding RAM sizes to process. The StarSpace program is thread-parallel, but training wall-clock times can nevertheless exceed a day for large training sets and many training iterations. Compared to training times, the prediction time of a single document is small in the range of milliseconds.

5 Conclusion

We presented a detailed three-part workflow that allows to combine multi-label classification with semantic tagging demonstrated on patent abstracts with more than 200 CPC categories. An annotated large training set is needed to accomplish good results. The semantic tagging is based on a set of key-phrases extracted by an unsupervised algorithm from a training set. The predicted key-phrases do not have to occur literally in the tagged document. The number of labels and key-phrases is highly scalable, given sufficient training data.

For future work, we plan to test our approach by replacing StarSpace with a deep neural network architecture. We already performed preliminary experiments with Transformer architectures, i.e. BERT [6], on the patent dataset and also on other textual datasets with different classification systems. The results on the patent dataset suggest that the performance of BERT is significantly worse than StarSpace with this amount of data and tests of both StarSpace and BERT on much larger datasets resulted in equal performance. We are planning to consolidate this hypothesis in more experiments.

Acknowledgements We thank the project team of Fraunhofer Digital for the opportunity, and Sven Giesselbach for helpful comments. This research has been funded by the Federal Ministry of Education and Research of Germany as part of the competence center for machine learning ML2R (01IS18038B).

References

1. Adhikari, A., Ram, A., Tang, R., Lin, J.: Docbert: Bert for document classification. arXiv preprint arXiv:1904.08398 (2019)

2. Aizawa, A.: An information-theoretic perspective of tf-idf measures. *Information Processing & Management* **39**(1), 45–65 (2003)
3. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. *Journal of machine Learning research* **3**(Jan), 993–1022 (2003)
4. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics* **5**, 135–146 (2017)
5. Cortes, C., Vapnik, V.: Support-vector networks. *Machine learning* **20**(3), 273–297 (1995)
6. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
7. Elman, J.L.: Finding structure in time. *Cognitive science* **14**(2), 179–211 (1990)
8. Hirschmeier, S., Schoder, D.: Combining word embeddings with taxonomy information for multi-label document classification. In: *Proceedings of the ACM Symposium on Document Engineering 2019*. pp. 1–4 (2019)
9. Jiménez-Ruiz, E., Agibetov, A., Chen, J., Samwald, M., Cross, V.: Dividing the ontology alignment task with semantic embeddings and logic-based modules. arXiv preprint arXiv:2003.05370 (2020)
10. Joachims, T.: Svm-light: Support vector machine. SVM-Light Support Vector Machine <http://svmlight.joachims.org/>, University of Dortmund **19**(4) (1999)
11. Leopold, E., Kindermann, J.: Text categorization with support vector machines. how to represent texts in input space? *Machine Learning* **46**(1-3), 423–444 (2002)
12. Levy, O., Goldberg, Y.: Neural word embedding as implicit matrix factorization. In: *Advances in neural information processing systems*. pp. 2177–2185 (2014)
13. Mahata, D., Kuriakose, J., Shah, R., Zimmermann, R.: Key2vec: Automatic ranked keyphrase extraction from scientific articles using phrase embeddings. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. pp. 634–639 (2018)
14. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: *Advances in neural information processing systems*. pp. 3111–3119 (2013)
15. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. pp. 1532–1543 (2014)
16. Prabhu, Y., Kag, A., Harsola, S., Agrawal, R., Varma, M.: Parabel: Partitioned label trees for extreme classification with application to dynamic search advertising. In: *Proceedings of the 2018 World Wide Web Conference*. pp. 993–1002 (2018)
17. Rose, S., Engel, D., Cramer, N., Cowley, W.: Automatic keyword extraction from individual documents. *Text mining: applications and theory* **1**, 1–20 (2010)
18. Wu, L.Y., Fisch, A., Chopra, S., Adams, K., Bordes, A., Weston, J.: Starspace: Embed all the things! In: *Thirty-Second AAAI Conference on Artificial Intelligence* (2018)
19. Zhang, M.L., Zhou, Z.H.: A review on multi-label learning algorithms. *IEEE transactions on knowledge and data engineering* **26**(8), 1819–1837 (2013)
20. Zhang, Q., Sun, Z., Hu, W., Chen, M., Guo, L., Qu, Y.: Multi-view knowledge graph embedding for entity alignment. arXiv preprint arXiv:1906.02390 (2019)
21. Zhang, X., Zhao, J., LeCun, Y.: Character-level convolutional networks for text classification. In: *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*. pp. 649–657. NIPS’15 (2015)

Native sentiment analysis tools vs. translation services - Comparing GerVADER and VADER

Karsten Michael Tymann, Louis Steinkamp, Oxana Zhurakovskaya, and
Carsten Gips

FH Bielefeld University of Applied Sciences, Minden, Germany
ktymann@fh-bielefeld.de, louis.steinkamp@fh-bielefeld.de,
oxana.zhurakovskaya@fh-bielefeld.de, carsten.gips@fh-bielefeld.de
<https://www.fh-bielefeld.de>

Abstract. VADER is a rule-based sentiment analysis tool for English texts with a social media focus. GerVADER is a German adaptation of VADER, which was developed following the steps of VADER’s development process. VADER showed high F1 scores especially for the social media domain, whereas the German adaptation achieved much lower results within the same domain, although on other test data. In this work we examine the question of whether these differences are language-specific. Therefore we apply an improved version of GerVADER to German texts and compare the results with the application of VADER to the same texts that are automatically translated into English. The benchmarking showed, that the translation combined with VADER achieves up to 5% higher F1 scores in all test cases, which can be explained by the translation tools automatic fixing of flawed sentences. However, native language tools can still be viable, since it saves time and costs and does not need another dependency to a third party service.

Keywords: VADER· GerVADER· sentiment analysis· translation

1 Introduction

Sentiment analysis describes the process of automatically rating texts or sentences with a sentiment value. The sentiment value ranges from negative, to neutral to positive and can be expressed as a numeric value or a classification in one of the three sentiment categories. Compared to machine learning based approaches classification can also be done by rule-based algorithms which have the advantage that they do not require any training data. However, developing a rule-based tool requires linguistic knowledge and is significantly more dependent on the target language. Hence one can not simply transfer one language’s features, e.g. German, to another language, e.g. English. The languages can differ in their grammar and overall sentence structure, making it very error prone

Copyright © 2020 by the paper’s authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

to simply transfer the algorithm to another language. How can negation be detected, or what is the meaning of a specific punctuation? Are there words that can have different meanings in different contexts and how can you derive those? Machine learning approaches simply train on lots of annotated data and extract the features with techniques like word embedding, but for a rule-based approach the developer has to design the process of detecting word patterns themselves. While a rule-based tool does not need bootstrap data, it will still need some form of lexicon with individual words or phrases to derive sentiments from. These lexicons are usually created by humans and are often rated by different people to offer an average sentiment value.

In this work, which was part of a student project at Bielefeld University of Applied Sciences, we will first discuss some improvements to our rule-based sentiment analysis tool GerVADER [5]. Second, we examine the question whether the effort to develop adaptations in the native language like GerVADER is worthwhile at all or whether one could achieve similarly good results by translating the corpora to be examined into English and using VADER [2] subsequently for the analysis. In order to investigate on this, we translated our test corpora to the English language with 3rd party tools and benchmarked the data with the English tool VADER.

2 Background

2.1 VADER & GerVADER

VADER is an abbreviation for “Valence Aware Dictionary and sEntiment Reasoner” and is free to use. It is a rule based tool for analyzing sentiments for English sentences. VADER managed to outperform other lexicon-based approaches as well as machine learning models. Especially in the domain of social media VADER achieved high F1 scores of up to 96%. [2]

GerVADER is an adaptation of the VADER tool for the German language. The process of VADER has been replicated in some steps, such as the crowd rating for the lexicon which is based on the SentiWS lexicon [3], while others have been simply transferred to the German language, such as the heuristics. GerVADER is as well free to use. [5]

2.2 Benchmarking corpora

SCARE is a corpus consisting of Google Play Store reviews. The reviews are categorized by their star ratings (1 to 5) and are split into 11 app categories. In total there are over 800.000 user reviews. [4]

The SB10k corpus consists of German tweets that are humanly labeled into the three sentiment categories: positive, neutral, negative. It consists of almost 10.000 tweets. Both corpora will be used for benchmarking purposes. [1]

2.3 Translation tools

For translating our test data, we have mostly relied on Googles translation service. One of the datasets has been additionally translated with MyMemory.

Googles translation service is based on the Google Neural Machine Translation (GNMT) system. Its hybrid model consists of a Transformer [6] encoder and RNN decoder. The learning is based on sequence-to-sequence neural network learning and is a mix between character and word-delimited models. [7]

MyMemory¹ is a large collection of Translation Memories that are collected and provided by humans and organizations. The translations are saved as words or sequences in databases which can then be matched by the users input. As of now there are over 4 billion human contributions.

3 Process

The process section is divided into two subsections. In the first we analyze the flaws of GerVADER and how we improved the algorithm. In the second subsection we will give insight on the benchmarking itself.

3.1 Flaws in GerVADER

The analysis of the initial version of GerVADER [5] showed three flaws that promised room for improvement. Firstly the negation detection is inaccurate and can not be converted to the German language by simply translating the negation keywords (e.g. 'not'). Secondly booster words (e.g. 'super', 'very') are sometimes the only words with a sentiment meaning in a sentence, but they do not get noticed, since they simply serve as booster for following words valences. Thus the sentences receive a neutral rating, although the booster word itself might carry sentiment meaning (e.g. 'super'). Thirdly misspelled words do not get noticed, since the words have to be written exactly like in the lexicon. For every problem case we developed test corpora so that we were able to tell whether our changes improved the overall rating. Details on the changes can be found on GitHub².

3.2 VADER vs. GerVADER

Both VADER tools cover their own languages. A question however arises whether it even makes sense to translate a sentiment analysis tool to one's native language. To investigate on this we translated our test corpora with Google and MyMemory.

For MyMemory only the SB10k corpus was used, whereas for Google Translate we tested multiple corpora (SB10k, SCARE). Additionally we constructed a SCARE_Balanced corpus, consisting of 400 positive, 400 negative and 400 neutral reviews of each SCARE corpus file. Thus a balanced file of all 3 sentiments is built, consisting of 13.200 entries.

¹ MyMemory by translated LABS <https://mymemory.translated.net/>

² GitHub - GerVADER <https://github.com/KarstenAMF/GerVADER>

Table 1. Benchmarking results for GerVADER and VADER (F1-3: mean of positive, negative, neutral F1-scores)

Tool	Corpus	F1	F1-3
GerVADER2.0	SB10k	39,63%	38,97%
VADER (Google)	SB10k	42,91%	44,20%
VADER (MyMemory)	SB10k	42,95%	44,44%
GerVADER2.0	SportNews (SCARE)	70,26%	50,98%
VADER (Google)	SportNews (SCARE)	71,62%	51,27%
GerVADER2.0	SCARE_Balanced (SCARE)	55,37%	44,06%
VADER (Google)	SCARE_Balanced (SCARE)	58,52%	45,69%

4 Results

GerVADER improved in all three areas for our test corpora by adjusting the original algorithm rules as well as adding new features such as fuzzy-matching. Thus the overall classification score of German texts has increased.

When comparing VADER with GerVADER Table 1 shows that VADER outperforms GerVADER of up to 5%. Even with the improvements in GerVADER, it is still outmatched. This is due to the fact, that translation tools do not just translate sentences word by word, but also consist of features such as fuzzy-matching and entity or POS (part-of-speech) tagging. Those are features, that we have partly integrated into GerVADER as well. Googles API is pre-trained with methods of Deep Learning on millions of data and adjusted to word and phrase sequences and not a simple word-to-word translation. Therefore it goes beyond a simple translation mechanism. As a result, spelling errors are corrected and the overall structure of the sentences is adapted to the desired output language. Thus it is not just a simple translation but also a text correction, which may explain the better results for VADER in the benchmark.

5 Conclusion & Future work

While GerVADER has been improved, looking at the translation comparison the question arises whether GerVADER serves any purpose. Developing a native language adopted tool is challenging and has lots of potential for creating new flaws. It requires linguistic knowledge in the target language, but this allows one to address language specific characteristics more appropriately than with a translation. Also, the translation service is an additional dependency, which can be problematic in factors of costs, time and complexity. With the current version it might not be worth it to trade GerVADER for VADER for a maximum of 5% F1 score improvement. However, translation tools handle the linguistic features for the developer and are therefore an interesting research topic for VADER and similar tools that are available in several languages.

References

- [1] Mark Cieliebak, Jan Deriu, Dominic Egger, and Fatih Uzdilli. “A Twitter Corpus and Benchmark Resources for German Sentiment Analysis.” In: *Proceedings of the 4th International Workshop on Natural Language Processing for Social Media (SocialNLP 2017)*, Valencia, Spain, 2017. 2017. DOI: 10.18653/v1/W17-1106.
- [2] C. Hutto and Eric Gilbert. *VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text*. 2014. URL: <https://www.aaii.org/ocs/index.php/ICWSM/ICWSM14/paper/view/8109>.
- [3] R. Remus, U. Quasthoff, and G. Heyer. “SentiWS - a Publicly Available German-language Resource for Sentiment Analysis.” In: *Proceedings of the 7th International Language Resources and Evaluation (LREC’10)*, pp. 1168-1171. (2010). 2010.
- [4] Mario Sanger, Ulf Leser, Steffen Kemmerer, Peter Adolphs, and Roman Klinger. “SCARE - The Sentiment Corpus of App Reviews with Fine-grained Annotations in German”. In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. Portorož, Slovenia, 2016. ISBN: 978-2-9517408-9-1. URL: <https://www.aclweb.org/anthology/L16-1178/>.
- [5] Karsten Michael Tymann, Matthias Lutz, Patrick Palsbroker, and Carsten Gips. “GerVADER - A German adaptation of the VADER sentiment analysis tool for social media texts.” In: *In Proceedings of the Conference “Lernen, Wissen, Daten, Analysen” (LWDA 2019), Berlin, Germany, September 30 - October 2, 2019*. 2019. URL: http://ceur-ws.org/Vol-2454/paper_14.pdf.
- [6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. “Attention Is All You Need”. In: *CoRR* abs/1706.03762 (2017). arXiv: 1706.03762. URL: <http://arxiv.org/abs/1706.03762>.
- [7] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. “Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation”. In: *CoRR* abs/1609.08144 (2016). arXiv: 1609.08144. URL: <http://arxiv.org/abs/1609.08144>.

EmoDex

An emotion detection tool composed of established techniques

Oxana Zhurakovskaya, Louis Steinkamp, Karsten Michael Tymann, and
Carsten Gips

FH Bielefeld University of Applied Sciences, Minden, Germany
oxana.zhurakovskaya@fh-bielefeld.de, louis.steinkamp@fh-bielefeld.de,
ktymann@fh-bielefeld.de, carsten.gips@fh-bielefeld.de
<https://www.fh-bielefeld.de>

Abstract. In this work we created an emotion analysis tool consisting of established models and techniques: Ekman's and Plutchik's emotion models, WordEmbedding (GloVe), VADER sentiment analysis, emoji features and a RandomForest classifier. Additionally we composed a corpus based on existing corpora and with the help of distant supervision. As a result, our approach achieves an accuracy increase of up to 10% compared to other emotion analysis tools (ParallelDots and Twitter Emotion Recognition), while at the same time offering a broader set of emotion classes. In addition, adding a sentiment feature increased the accuracy by about 2%. We make the conclusion that a combination of features from multiple sources such as GloVe and VADER offer a good basis for a RandomForest classifier while only training on a very small set of texts (less than 70k sentences).

Keywords: Emotion detection · Random Forest · Word Embedding · GloVe · VADER · Emoji labelling · Sentiment analysis · Emotions · Ekman · Plutchik · Distant supervision · CrowdFlower · Feature selection

1 Introduction

Emotions are an important part of human communication. They influence the semantic meaning of sentences and can therefore convey additional information. While having a face to face conversation one is able to derive the emotional meaning of the partner's message not only by the actual spoken sentences but also by incorporating other factors such as facial expressions, gestures and voice. When reading texts, these natural factors are lost. Communications in textual form can therefore be misinterpreted, especially by machines. But being able to identify the emotion of an online text can be beneficial for multiple applications. With modern natural language processing (NLP) it is possible to process text messages to detect which emotions are expressed.

Copyright © 2020 by the paper's authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

Interpreting the emotions of a text can be done by a rule based algorithm or machine learning which requires a lot of training data. When relying on supervised learning the data needs to be labeled with emotion categories. To bootstrap the machine learning model, there exists a variety of already labeled corpora. There are also methods such as distant supervision (see section 2.2) to automatically annotate texts with emotions. Different models can be used for emotion labels, like Ekman's [5] and Plutchik's [13]. The more emotion categories, the more complex the classification can become. Thus emotion classification can be often harder than just deriving a sentence's sentiment.

Common approaches in using the text as classification features involve training on the word embeddings of the texts. Some also include separately crawled corpora for emojis which can serve as additional features. Others might as well include separately trained features such as hashtags or features by other classifiers such as sentiment analysis tools.

In this work, which was part of a student project at Bielefeld University of Applied Sciences, we created a free to use emotion analysis webservice, called EmoDex¹. We focus on existing tools and models such as Plutchik's emotion model [13], GloVe [12] for word embedding and existing corpora to train a Random-Forest-Classifer with scikit-learn². Additionally we add as separate features emoji categories as well as a sentiment rating provided by VADER [8]. We will describe all steps from collecting and preprocessing the texts, to testing out whether a separately calculated sentiment score or using distant supervision can improve the classification. EmoDex will be compared to two models, one being ParallelDots API³ while the other being TwitterEmotionRecognition [4].

2 Background

2.1 Emotion Models

To classify text with emotions first we should decide which are the basic emotions that can be identified. Two often used models are Ekman and Plutchik. Ekman's model highlights 6 basic emotions: sadness, happiness, anger, fear, disgust, surprise. The emotions selected in this model are discrete and based on facial expressions as well as neurobiological processes independent of cultural differences. Whereas Plutchik's multidimensional model of emotions is based on the psychoevolutionary theory of emotions. This model identifies 8 basic emotions: joy, trust, fear, surprise, sadness, disgust, anger, anticipation (see Fig. 1). Each have additional intensity levels. [5, 13, 16]

Another model type describes emotions in dimensions. The Valence-Arousal-Dominance (VAD) or also called Pleasure-Arousal-Dominance (PAD) model defines three axes which locate emotions in a space. First the pleasure or valence

¹ EmoDex <https://emodex.net/>

² <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>

³ ParallelDots Inc. <https://www.paralleldots.com/emotion-analysis>

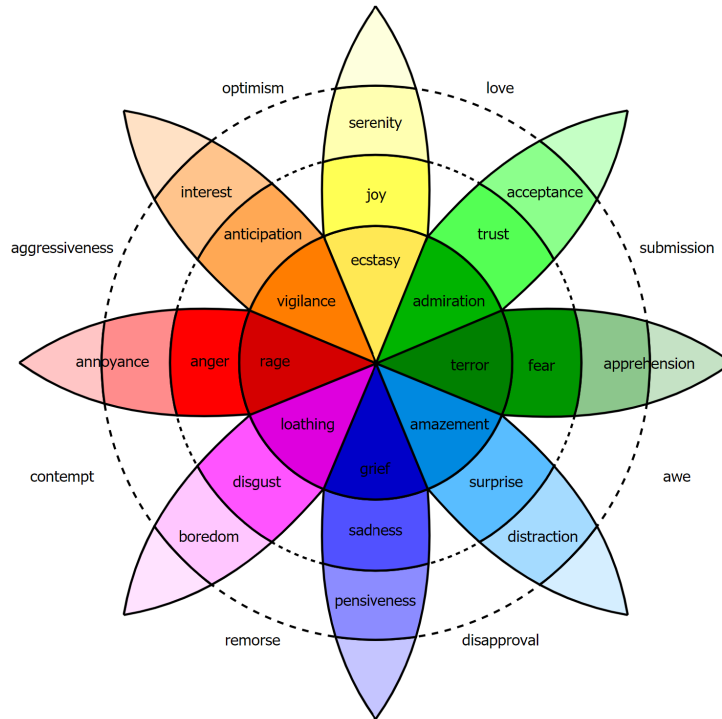


Fig. 1: Plutchik's wheel of emotion with the 8 inner basic emotions [7]

axis describes how pleasant a feeling is. Second the arousal axis shows how much a person feels “activated”. Being excited would for example be high in arousal whereas sadness or calmness have a low arousal value. From high arousal feelings an action can be more expected by the individual than from a person having a low arousal emotion. Thirdly the dominance scale shows how dominant or submissive the persons feeling is. Being angry would be a very dominant feeling while sadness would indicate a more submissive behaviour. [19, 14]

All three models are types of different emotions categorization. Ekman's model describes emotions as discrete emotions whereas the VAD/PAD model by A. Russell describes them dimensional. Plutchik's can be regarded as a hybrid model, where the 8 basic emotions can be extended by further emotions as dimensions. [19, 14, 16]

2.2 Distant supervision & expert labeling

To label the data for training purposes we used expert labeling in combination with distant supervision.

Expert labeling describes the process that the test data has been annotated by human experts. In the best case, a test data set is evaluated by several experts

so that the label of the data set is as accurate as possible. However, this type of labeling is very labor-intensive and subjective due to the human judgement [18].

The other type of labeling is the automatic creation of labeled datasets, called distant supervision. Especially via Twitter, the hashtag search can be used to filter for emotions. For example, if one searches with the hashtag #joy, tweets that are annotated with the hashtag will be returned. The assumption is that the user has used this hashtag to express his emotions in this tweet. Therefore the tweet will also be labeled with this emotion label in the data set.

The authors of the paper [18] compared the accuracy of expert annotation and remote supervision and created a test corpus of 400 tweets, which was annotated using both methods. The result is that the labels match 93.16% and are therefore suitable as a meaningful label for the dataset.

2.3 Corpora compilation

One of the main components of emotion recognition in texts is the corpus on which a ML algorithm can be trained. Since there are already several papers that have dealt with the topic of corpora creation, a collection of corpora which are free to use for research purposes will be used in this work.

The paper [3] has already examined various of such corpora in detail and analysed their suitability for classification. The corpus is based on 14 different corpora, which are labelled according to different emotion categories and come from different topics like news, blogs, weather or general. Furthermore their type of labeling process (distant supervision & expert labeling) is shown and they are differentiated in their granularity, like tweets, headlines or simple sentences. The result of their work included also a mapping of emotion classification, in order to merge all individual corpora in one data set. Only two of the corpora have Ekman and Plutchik as emotions model. As a result of this mapping, seven data sets use Ekman as a basis, while they extend the model by one or two additional emotions. Two corpora are according to the VA/PA and VAD/PAD model respectively and therefore find no observation in our further investigation. One model is only divided into happy and sad and is therefore comparable to a sentiment classification.

2.4 Detecting emotions with WordEmbedding

WordEmbedding describes a recent trend in ML and NLP where words are represented as vectors in a vector space. The embedded words are therefore in a relation to each other that can be measured as the vector distance. [9]

Word2Vec and GloVe, both being WordEmbedding techniques, are useful for emotion analysis since they represent the words with their semantic meaning in a vector space. Therefore the assumption is that the words in the same clusters offer a similar emotion. For an emotion analysis the sentence can therefore be split into the dimensional representation of the total of the words dimension vectors. With the so received dimension vector of the sentence, a machine learning classifier can be trained. [9, 1, 12]

2.5 Emoji Labelling

Emojis make a significant contribution to non-verbal communication in texts. [17] Through them, users are given another opportunity to express their emotions. [6] Normally displayed as icons, it is also possible to interpret emojis as unicodes which are defined by the Unicode Consortium⁴.

Emojis can therefore be used to abstract further information about the emotions in texts. The basis for this is an Emoji Emotion Mapping which assigns an emotion to selected emojis. The mapping makes it possible to classify a text with an emotion only on the basis of emojis in text.

The authors of the paper [6] have crowdsourced 202 emojis with an emotion label. In total 308 users submitted 15155 ratings. The result is an emoji emotion label mapping. As soon as an emoji was rated over 50% with this label, it was assigned to this emotion. Their work is used as a basis for our emoji classification. [6]

2.6 VADER

VADER is a sentiment analysis tool that is based on a crowd rated sentiment lexicon used in a rule based algorithm. The python tool rates sentences on a scale from -1 (negative) to 0 (neutral) to 1 (positive). The tool showed good results in the domain of social media. [8]

2.7 Benchmarking with other tools

For benchmark purposes and comparing our results to other approaches we picked two tools. We chose the ParallelDots API as well as the Twitter Emotion Recognition tool for our comparison.

ParallelDots is developing different NLP and AI products. They offer an API for their emotion recognition tool, which can detect emotions of 6 different categories: happy, sad, angry, fear, excited and indifferent. According to their blog, their model is based on Convolutional Neural Networks (CNNs).

The Twitter Emotion Recognition tool is able to predict emotions for English tweets. [4] It requires no preprocessing since it works on the words characters. It provides a trained Recurrent neural network (RNN) which can predict one of the following categories: Ekman's six basic emotions, Plutchik's eight basic emotions and Profile of Mood States six mood states.

3 Process

3.1 Corpus in detail

As described in chapter 2, the basis of the corpus in this work is a collection of different corpora consisting of tweets. For this purpose five corpora were used, which are described in more detail in the following:

⁴ Unicode Org <https://unicode.org/emoji/charts/full-emoji-list.html>

Name	Source	Emotion	Size	Labeling
Crowd-Flower	Unify Emotion Datasets [3]	Ekman + Love + NoEmotion	39.740	Crowdsourcing
Electoral-Tweets	Mohammad [11]	Plutchik	4.058	Crowdsourcing
EmoInt	Mohammad [2]	Ekman - Disgust - Surprise	7.097	Crowdsourcing
SSEC	Schuff et al. [15]	Plutchik	4.868	Expert Annotation
TEC	Mohammad [10]	Ekman	21.051	Distant Supervision

Table 1: Corpora

As shown in Table 1, three of the corpora are based on Ekman and two on Plutchik. CrowdFlower is extended by the emotion 'Love' and the label 'No Emotion'. EmoInt is shortened to four emotions in which 'disgust' and 'surprise' are removed. This results in a total of 10 labels with which the respective data records can be marked and a total number of 76.814 entries.

Plutchiks emotions 'trust' and 'anticipation' were removed from the data set for this work, because their share is too small in comparison. What remains is a corpus consisting of 8 emotion labels and 72.762 data sets.

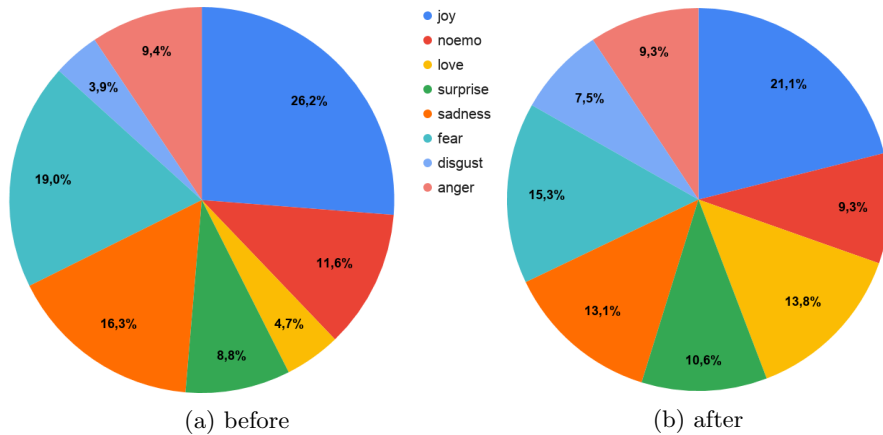


Fig. 2: Emotions distribution before and after distant supervision

Fig. 2a shows how the individual proportions of the respective emotion labels in the corpus are distributed. For eight emotions the average is 12.5 percent. While emotions like joy and fear are far above this average, emotions like disgust and love are far below.

If an ML algorithm is trained with this data set, emotions like disgust or love are hardly recognized because their share is too small. To compensate for this, the proportion of emotions in the corpus can be influenced. There are two possibilities for this. Either the shares of the dominant labels are reduced or the shares of the neglected labels are increased. Since reducing the data is not desirable, we have added more data to the data set using distant supervision.

Due to the fact that our selected corpora are all based on tweets, we decided to use Twitter as data source as well. To crawl Twitter we use the hashtag based search provided by the Twitter API. The hashtags we use for search are based on the National Research Council of Canada (NRC) Hashtag Lexicon for non-commercial purposes by Saif Mohammad [10] which provide multiple hashtags for six of our eight emotion labels. Every hashtag from the lexicon has a score which represents the strength of association between the hashtag and the emotion. We have chosen the highest rated hashtags and added our own tags, so that we arrived at ten hashtags per emotion. For the emotion label 'Love' we have only used our own hashtags, since the label is missing in the Hashtag Lexicon. Finally with this selection we have crawled further 19662 tweets for the emotion labels 'angry', 'disgust', 'surprise' and 'love', so that the corpus for this work comes to 92452 tweets in total.

A percentage distribution of the emotion labels in the final corpus can be seen in Fig. 2b. The percentages of the dominating labels have been reduced, while the percentages of the neglected labels have been increased.

3.2 Preprocessing

The prepared corpus should be preprocessed to remove irrelevant words and signs as well as emoticons and emojis. In order to take emojis into account we considered to add additional features, that represent the emotion category of emojis based on the idea of emoji labelling (see section 2.5). In contrast to the referenced approach, the emojis in this work are categorised by only one human and each emoji is assigned to only one category. There are eight categories in total, that are appropriate to the selected emotions features: joy, love, surprise, disgust, sadness, fear and neutral. Each text should have a count for each of these emojis categories. Thus in the first step emojis in each text should be counted and the count should be added to the feature vector. Fig. 3a shows an example of categorised emojis. All emojis are removed from the original text after counting.

In order to process the emoticons, the emoticons were replaced with a word representing them. Fig. 3b shows an example of emoticons and their descriptions. The replacement of emoticons with appropriate description is the second step in the preprocessing pipeline.

As the third step all letters in the text are converted to lowercase. In order to reduce the word amount to be processed, some unnecessary words should be removed. Thus stopwords, URLs, usernames like "@name" and hashtags like "#tag" are removed. The negation words were however left in the text, because these influence the emotional meaning of the text. After the removal manipulation there can be left empty texts. Consequently the empty texts are removed from

Emoji	Category
😊	joy
😍	love
😲	surprise
😐	neutral
😬	disgust
😞	sadness
😨	fear
😡	anger

(a) Emoji category mapping

Emoticon	Description for replacement
&:	disgust
(%:	confused
((-:	smile
(-:<	gloatingly

(b) Emoticon replacement mapping

Fig. 3: Emoji and Emoticon mappings

the corpus. Additionally we add a sentiment classification score of the text to the feature vector. The idea was to enhance the classification of emotions by providing an additional feature, that allows to better distinguish negative and positive emotions like “joy” and “sadness”. Therefore we processed each text of the corpus with the VADER tool (see section 2.6) and added the result as a feature. In order to compare the influence of sentiment features on emotion classification we store one corpus with VADER preprocessing and one corpus without VADER classification.

3.3 Use of GloVe

After preprocessing, the texts can be converted into their vector representation. For this purpose a pre-trained word embedding model from GloVe is used in this paper. The model was trained on 2B tweets and contains a total vocabulary of 1.2 M words. For our work we used a word vector resolution of 100 dimensions. For each word in a tweet the vector was calculated from the model. Then the average of the sum of the words was used as a vector representation of the tweet and got appended to the corpus to serve as features. [12]

3.4 Random Forest Classifier

For classification we selected the Random Forest (RF) algorithm, which consists of multiple Decision Trees, each predicting the outcome class independently of each other. The class that gets the most “votes” is the result of the whole Random Forest. One of the advantages of this method is reducing the errors

of predictions that can occur when predicting only with a single individual tree. Another advantage is overfitting control.

For implementing the Random Forest we use the scikit-learn⁵ framework, that provides ready to use functions with configuration options. We use the Random Forest classifier with the default options, with 200 trees and the random state set to 42. We train the classifier on 75% (67.5k) of our corpus and tested it with the other 25% (22.5k).

4 Results

For the training and testing of the data we have divided our corpus in a ratio of 3 to 1. The test data was randomly selected from the entire corpus. We have trained four different models (with/without VADER, with 8/4 emotions) and use two types of analysis (with/without 20% threshold) to evaluate the results. We compared our results with the two tools explained in chapter 2.7: ParallelDots (PD) and Twitter Emotion Recognition Tool (TER). We have benchmarked both by predicting parts of our corpus, however some mappings for the emotions had to be made. The results are shown in Table 2.

The ParallelDots API only returned classification for the five emotions: happy, sad, angry, fear and excited. We mapped the emotion 'happy' to our emotion 'joy'. The emotion excited has been removed from the evaluation. We tested the API with the 4 corresponding emotions from our corpus. Thus we have a total of 26028 rated tweets and an accuracy of 40.84% (see Table 2 row 2).

TER uses Plutchik's emotion model and thus directly covers six of our eight emotions. We have removed the two additional emotions trust and anticipation as well as our data sets labeled love or noemo. For this tool we have had 8938 tweets predicted and have an accuracy of 31,02% (row 1).

Our tool was trained with 75% and tested with the other 25%. This resulted in a test data set of 22470 data sets which our model with VADER (row 4) rated correctly with 45,11%. Thereby all eight emotions were used. Without VADER as an additional feature (row 3), the result is 43,76% with eight emotions.

For both versions, with VADER as a feature and without, a model with only four emotions was trained and tested. These are joy, sadness, anger and fear and are in accordance with the four emotions, which are also used for our benchmarking for ParallelDots. For training and testing, the data sets with the remaining four emotions were removed from the corpus. The results are 53,35% accuracy with VADER (row 6) and 51,39% (row 5) without.

All results were evaluated by using the emotion label with the highest classification accuracy (row 1-6). Since the ratings of eight labels can be close to each other, we used a second rating strategy for this model. It was checked whether an emotion label was rated with more than 20%. If so, it was added to the result set. The prediction was then marked as true if the test record label was within the result set. As a result the classification accuracy increased (row 7, 8).

⁵ <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>

Row	Tool	Features	Tweets	True	False	Accuracy	Emotion Label
1	TER	Pretrained RNN	8938	2613	6325	31,02%	Anger, Disgust, Fear, Joy, Sadness, Surprise, Trust , Anticipation
2	PD	Pretrained CNN	26028	9526	16499	40,84%	Happy (Joy), Sad, Angry, Fear, Excited
3	EmoDex	RF, GloVe	22463	9829	12634	43,76%	Joy, Anger, Sad, Disgust, Fear, Surprise, Love, NoEmo
4	EmoDex	RF, GloVe, VADER	22470	10136	12334	45,11%	Joy, Anger, Sad, Disgust, Fear, Surprise, Love, NoEmo
5	EmoDex	RF, GloVe	12585	6467	6118	51,39%	Joy, Anger, Sad, Fear
6	EmoDex	RF, GloVe, VADER	12585	6714	5871	53,35%	Joy, Anger, Sad, Fear
7	EmoDex	RF, GloVe, 20% threshold	22463	13300	9163	59,21%	Joy, Anger, Sad, Fear
8	EmoDex	RF, GloVe, VADER, 20% threshold	22470	14227	8243	63,32%	Joy, Anger, Sad, Fear

Table 2: Benchmark results

In summary it can be said that the use of VADER as an additional feature brought a gain in accuracy of 1.35% for eight emotion labels and a gain of almost 2% for four emotion labels. The reduction from eight to four emotions brought a gain of about 8%, whereby it should be noted that the model with four emotions was trained and tested on a corpus that was almost 44% smaller.

5 Conclusion and future work

This works approach is mostly based on already established techniques in NLP. We have shown that combined techniques – labeled corpora, distant supervision, Glove, emoji categories, VADER, and random forest classifiers – complement each other to an efficient tool.

In this work the emojis were categorized by one human, thus the emoji labelling is subjective. In future works the emojis can be labeled using crowdsourcing methods or automatically with machine learning methods.

The emotions distribution on the used corpus was not even, thus the results can be affected by this. For future testing the corpus should be build with respect to emotion distribution. Compared to other papers, we have worked with a small dataset, thus our results should be tested with more data in a future analysis.

Compared to the other tools, this works approach has the highest classification score while also offering the most emotion categories in the described test environment. Moreover it is demonstrated that adding sentiment features by third party tools to the feature vector can increase the accuracy result. Additionally distant supervision proved itself to be useful for expanding the corpus.

References

- [1] Tomas Mikolov et al. *Learning Representations of Text using Neural Networks*. 2013. URL: <http://www.micc.unifi.it/downloads/readinggroup/TextRepresentationNeuralNetwork.pdf> (visited on 02/01/2020).
- [2] Alexandra Balahur, Saif M. Mohammad, and Erik van der Goot, eds. *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017. DOI: 10.18653/v1/W17-52.
- [3] Laura Ana Maria Bostan and Roman Klinger. “An Analysis of Annotated Corpora for Emotion Classification in Text”. In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, 2018, pp. 2104–2119. URL: <http://aclweb.org/anthology/C18-1179>.
- [4] N. Colnerić and J. Demsar. “Emotion Recognition on Twitter: Comparative Study and Training a Unison Model”. In: *IEEE Transactions on Affective Computing* (2018), pp. 1–1. DOI: 10.1109/TAFFC.2018.2807817.
- [5] Paul Ekman. “Are there basic emotions?” In: *Psychological Review* 99.3 (1992), pp. 550–553. DOI: 10.1037/0033-295X.99.3.550.
- [6] Abdallah El Ali, Torben Wallbaum, Merlin Wasmann, Wilko Heuten, and Susanne Boll. “Face2Emoji: Using Facial Emotional Expressions to Filter Emojis”. In: May 2017, pp. 1577–1584. DOI: 10.1145/3027063.3053086.
- [7] Wikipedia - The Free Encyclopedia. *Robert Plutchik*. URL: https://de.wikipedia.org/wiki/Robert_Plutchik (visited on 02/01/2020).
- [8] C. Hutto and Eric Gilbert. *VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text*. 2014. URL: <https://www.aaai.org/ocs/index.php/ICWSM/ICWSM14/paper/view/8109>.
- [9] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. “Distributed Representations of Words and Phrases and Their Compositionality”. In: *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*. NIPS’13. Lake Tahoe, Nevada: Curran Associates Inc., 2013, pp. 3111–3119.
- [10] Saif Mohammad. “#Emotional Tweets”. In: **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings*

- of the *Sixth International Workshop on Semantic Evaluation (SemEval 2012)*. Montréal, Canada: Association for Computational Linguistics, July 2012, pp. 246–255. URL: <http://www.aclweb.org/anthology/S12-1033>.
- [11] Saif Mohammad, Xiaodan Zhu, Svetlana Kiritchenko, and Joel Martin. “Sentiment, emotion, purpose, and style in electoral tweets”. In: *Information Processing & Management* 51 (Oct. 2014). DOI: 10.1016/j.ipm.2014.09.003.
- [12] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. “GloVe: Global Vectors for Word Representation”. In: *Empirical Methods in Natural Language Processing (EMNLP)*. 2014, pp. 1532–1543. URL: <http://www.aclweb.org/anthology/D14-1162>.
- [13] Robert Plutchik. “A psychoevolutionary theory of emotions”. In: *Social Science Information* 21.4-5 (1982), pp. 529–553. DOI: 10.1177/053901882021004003.
- [14] James A Russell and Albert Mehrabian. “Evidence for a three-factor theory of emotions”. In: *Journal of Research in Personality* 11.3 (1977), pp. 273–294. DOI: [https://doi.org/10.1016/0092-6566\(77\)90037-X](https://doi.org/10.1016/0092-6566(77)90037-X).
- [15] Hendrik Schuff, Jeremy Barnes, Julian Mohme, Sebastian Padó, and Roman Klinger. “Annotation, Modelling and Analysis of Fine-Grained Emotions on a Stance and Sentiment Detection Corpus”. In: *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017, pp. 13–23. DOI: 10.18653/v1/W17-5203.
- [16] Armin Seyeditabari, Narges Tabari, and Wlodek Zadrozny. “Emotion Detection in Text: a Review”. In: *CoRR* (2018). arXiv: 1806.00674.
- [17] Jessica L Tracy, Daniel Randles, and Conor M Steckler. “The nonverbal communication of emotions”. In: *Current Opinion in Behavioral Sciences* 3 (2015). Social behavior, pp. 25–30. DOI: <https://doi.org/10.1016/j.cobeha.2015.01.001>.
- [18] Wenbo Wang, Lu Chen, Krishnaprasad Thirunarayan, and Amit P. Sheth. “Harnessing Twitter “Big Data” for Automatic Emotion Identification”. In: *Proceedings of the 2012 ASE/IEEE International Conference on Social Computing and 2012 ASE/IEEE International Conference on Privacy, Security, Risk and Trust*. SOCIALCOM-PASSAT '12. USA: IEEE Computer Society, 2012, pp. 587–592. ISBN: 9780769548487. DOI: 10.1109/SocialCom-PASSAT.2012.119.
- [19] Shen Zhang, Zhiyong Wu, Helen Meng, and Lianhong Cai. “Facial Expression Synthesis Based on Emotion Dimensions for Affective Talking Avatar”. In: vol. 2010. June 2010, pp. 109–132. DOI: 10.1007/978-3-642-12604-8_6.

FGWM Workshop

Construction of a Corpus for the Evaluation of Textual Case-based Reasoning Architectures

Andreas Korger¹ and Joachim Baumeister^{2,3}

¹ Angesagt GmbH, Dettelbachergasse 2, D-97070 Würzburg

² denkbares GmbH, Friedrich-Bergius-Ring 15, D-97076 Würzburg

³ University of Würzburg, Am Hubland, D-97074 Würzburg

Abstract. Regulatory documents denote an interesting application domain for case-based knowledge management. These documents enumerate situations with conditions, that are often dangerous for human and environment and they give advice, rules, and instructions for prevention or handling. That type of documents is eminent in many domains and provides valuable experience knowledge which makes it a remarkable application and research domain for (textual) case-based reasoning. In this paper, an initial case-based representation of regulatory documents is introduced. We report on the construction of an open corpus of regulatory documents in the domain of nuclear safety regulations.

Keywords: Case-based Reasoning · Experience Management · Knowledge Management · Textual Case-based Reasoning · Corpus Annotation · Natural Language Processing.

1 Introduction

Case-based knowledge management approaches seem promising to handle regulatory documents. In general, regulatory documents are published by authorities covering the handling of situations of a specific domain. A document enumerates noteworthy situations with conditions, that are often dangerous for human and environment. Based on a detected situation the document gives advice, rules, and instructions for preventing or handling a particular situation.

Examples of regulatory documents are *compliance documents* of large companies, *safety documents* for conventions and festivals, and *legislative agreements* between parties.

This type of document is eminent in many areas and provides valuable experience knowledge. Furthermore, a single document often covers experience knowledge from different domains thus requiring the collaboration of many domain experts. For instance, the compliance document of a company will require prevention and handling rules defined by experts from the law and human resources departments. A safety document of a festival will cover regulatory situations defined by experts from the fire department and the police.

Copyright © 2020 by the paper's authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

Today, the use of regulatory documents faces two essential challenges:

- Creation: Documents are commonly created manually and collaboratively by domain experts in a knowledge intensive process. The reuse of existing knowledge especially found in existing regulatory documents, is mostly not present.
- Retrieval: The documents are usually available in plain text and therefore the retrieval of suitable information for a given situation is complex and time intensive.

We see *regulatory documents* as an interesting application and research domain for (textual) case-based reasoning and we formulate the following hypotheses:

1. Case-based reasoning can provide a natural representation for covering experience knowledge.
2. Case-based reasoning can handle incomplete input for the formulation and retrieval of experience knowledge.
3. Case-based reasoning is very suitable to integrate background knowledge into the process provided, e.g., by ontologies.

To tackle these hypotheses, we started to build an open corpus of regulatory documents that can be used to evaluate interesting research questions:

- Case-oriented representation of regulatory documents
- Textual CBR in the domain of regulatory documents
- Retrieval of regulatory experience knowledge
- Reuse and generation of new experience knowledge
- Case-based review critique of regulatory documents
- (Textual) quality assessment of regulatory documents

To the knowledge of the authors there exists currently no open corpus of regulatory documents, that can be used to work on the research questions stated above.

In this paper, we introduce an initial case-based representation of regulatory documents and we report on the construction of an open corpus of regulatory documents in the domain of *nuclear safety regulations*. We invite volunteers to join this construction process. Preceding work in the field of knowledge management and case-based reasoning laying fundamentals was presented by Korger and Baumeister [22, 24].

The paper is organized as follows: In Section 2 we introduce the PIRI structure representing regulatory experience knowledge as a case-based interpretation and we sketch how case-based reasoning is used to work with regulatory documents. The construction of the corpus of nuclear safety regulations is described in Section 3, where we introduce the intention and the statistics of the corpus. We show the methods used for the construction of the corpus and we explain how to obtain the corpus for own research. In Section 4 we discuss a number of use cases applicable for the introduced corpus. The paper is concluded with related and future work in Section 5.

2 Case-Based Representation of Regulatory Documents

A regulatory document lists a collection of incidents of interest for a given life or work context. For each incident of interest, the document usually describes measures for prevention and measures for handling an occurring incident. To know, which incidents are actually of interest in a certain scenario and which measures are effectively applicable, is the result of experience collected in the past. A key to access this tacit knowledge is to identify parts of the document corresponding to incidents and measures, as well as to find a metric to make them comparable within a certain context. These considerations will be reflected in the case structure presented in the following.

2.1 The Domain of Regulatory Documents

A regulatory document consists of a sequence of text passages describing specific aspects of the domain. We call interesting passages within a document *information units*.

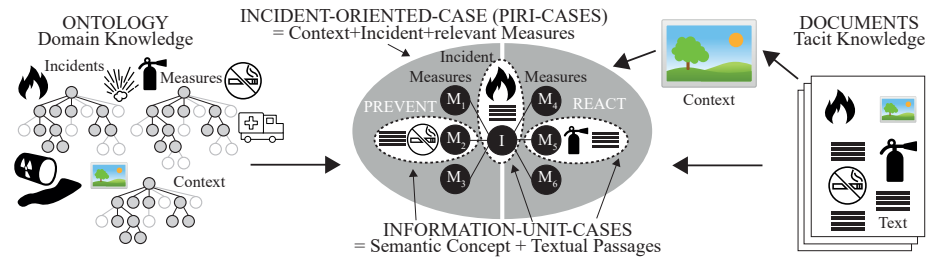


Fig. 1: Domain knowledge with semantic information represented in an ontology and tacit knowledge encoded in documents yielding two different case structures.

Definition 1 (Information Unit and Corpus). An information unit u is defined as a text passage within a document, i.e., $u = (d, f, t)$, where d is a document URI and f, t are offset information describing the extract of the document. For a corpus \mathcal{RD} , the universal set of all information units is defined as $IU_{\mathcal{RD}}$.

For semantic interpretation of the text passage, such information units need to be annotated by metadata, explicitly describing the content.

Definition 2 (Ontology). An ontology $\mathcal{O} = (E, R)$ contains metadata relevant for the considered domain. Here, relevant semantic concepts represented as entities $e \in E$, that are connected by relations $r \in R$.

Typical examples for entities within a regulatory domain are *fire*, *smoking prohibition*, *explosion*, and *evacuation*. Common relations in this context are

partOf and *requires*. A symbolic depiction of the ontology, the documents, and the case considerations can be seen in Figure 1.

When defining regulatory documents in a case-based representation, we distinguish two design alternatives:

- *Information unit cases*: A case is represented by a text passage (information unit) describing an incident or measure. The case also contains annotations explicitly describing the incident/measure using elements of the ontology. A case base then contains a collection of information units as cases and incident/measure annotations are attributes representing the particular cases.
- *Incident oriented cases*: One distinct case is represented by one incident together with all measures mentioned in a document for preventing and handling the incident. The case base then contains a collection of incidents with measures as attributes representing the particular cases. It is worth noticing, that incident oriented cases are constructed by aggregating information unit cases.

In the following, we describe both representation alternatives in more detail and we motivate their use by retrieval and reuse examples.

2.2 Information Unit Case Structure

To build a bridge between semantic concepts and free text an *information unit case* combines a distinct information unit with a corresponding metadata annotation. Having in mind that the overall goal is to reuse documents, it is convenient to consider textual passages as solutions to structured problem descriptions [14]. Subsequently, it is assumed that the textual passage is reusable for similar problems.

Definition 3 (Information Unit Case). *An information unit case c_u is defined as follows: $c_u = (a, u)$, where $u \in IU_{\mathcal{RD}}$ is an information unit from corpus \mathcal{RD} and $a \subseteq \mathcal{O}$ is describing metadata from an ontology \mathcal{O} . We call the case base of information unit cases $\mathcal{CB}_U = \{c_{u_1}, \dots, c_{u_n}\}$ the collection of all cases $c_{u_i}, \forall i \in \{1, \dots, n\}$ that are extracted and annotated from available regulatory documents.*

A case $c_u = (p_u, s_u)$ representing one information unit is defined by a set of named entities and relations between them as problem description p_u and textual passage contained in a document indicating the fulfillment as solution s_u to the problem. These textual passages are referred as fulfilling textual features. A textual passage may be just one word up to some sentences. Depending on the use case scenario the description and the solution of the case might switch. For instance given a textual passage as problem description the metadata is the solution. An exemplary information unit case is:

$c_x = (\text{piri:manualFireFighting}, \text{"find a fire extinguisher and put out the fire"})$

Similarity of Information Units To define similarity functions for information unit cases, we exploit the taxonomic interrelation of semantic concepts. Both incidents and measures can be well classified into a taxonomy, building the base for the similarity assessment and the adaptation. The similarities are calculated via the taxonomic order of its elements. Each element of the hierarchy is assigned with a likelihood symbolizing the similarity of its sub-elements. The similarity of the leaf elements is set to 1 and to 0 for the root element. The similarity increases with depth d of the element according to for instance $sim_d = 1 - 1/2^d$, [11]. With this atomic case structure, basic retrieval and reuse is possible. Picking up the previous example one could state the question, what to do, if there is no fire extinguisher in reach. This can be solved e.g. by the retrieval of measures more special than *manual fire fighting*, like *manual fire fighting with clothes*. A retrieved case would be for instance:

$c_x = (\text{piri:manualFireFightingWithClothes, "take off your jacket and use it to put out the fire by throwing it onto the flame"})$

2.3 Incident-Oriented Case Structure (PIRI)

A reoccurring pattern sharpens the focus on the essence of the regulatory documents. The pattern selected for this purpose bases on the assumption that a regulatory document delivers a core message. In terms of safety, the most important content are the incidents it mentions as well as the measures to prevent them or to react to their consequences. For a given context and relevant incident induced by the context the according measures are ordered by importance and classified into preventive and reactive measures. Other patterns might be considered for other tasks in an analogous way.

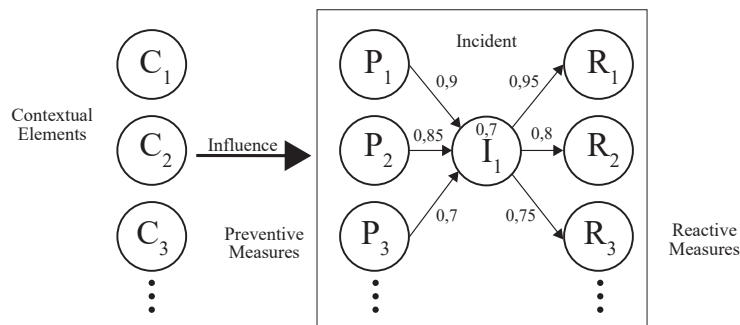


Fig. 2: PIRI-diagram under a context $C = (C_1, \dots, C_j)$ showing the ranked preventive (P) and reactive measures (R) with the according importance weights for the incident and measures.

We call the instances of these information patterns PIRI-snippets (Preventive-Incident-Reactive-Interrelation) [23]. The presented model reduces the complex-

ity of the real world for facilitation of assessment. Typically there is a cascade of measures that are executed in a specific order. For instance in the case of fire, first evacuate all people, then close the doors and windows. The PIRI-pattern in its graphical representation can be seen in Figure 2. The corresponding case structure is given in the following definition:

Definition 4 (The Incident-Oriented PIRI-Case). *A PIRI-case is defined as follows: $c_{PIRI} = (P_I, I, R_I, C_I)$, where $I \in \mathcal{CB}_U$ is a case describing a specific incident, $P_I \subseteq \mathcal{CB}_U$ denotes a set of cases describing preventive measures for I , $R_I \subseteq \mathcal{CB}_U$ denotes a set of cases describing reactive measures for I , $C_I \subseteq \mathcal{CB}_U$ denotes a set of cases describing the context of the incident I . We call the case base of PIRI-cases $\mathcal{CB}_{PIRI} \subseteq \mathcal{P}(\mathcal{CB}_U)$, where $\mathcal{P}(\mathcal{CB}_U)$ is the set of all subsets from \mathcal{CB}_U .*

A case describing one PIRI-snippet is the subset of the case base \mathcal{CB}_U containing only cases of information units that are related to an incident I . In this case definition the importance weights are neglected. We see, that PIRI-cases are an aggregated form of information unit cases, and more sophisticated retrieval and reuse scenarios are possible. An exemplary PIRI-case for the incident *fire* looks as follows.

$c_x = ([\text{piri:fireAlarmSystem, piri:fireDrill}], \text{piri:fireIncident},$
 $[\text{piri:callFireDepartment, piri>manualFireFighting, piri:evacuate}],$
 $\text{piri:doc_id123_fire_fighting_in_power_plants})$

Similarity of PIRI Cases To retrieve similar cases for instance similar PIRI-snippets, the case base is searched for similar problem descriptions p_i to the query q_1 . A query is made up by a set of named entities, relations between them and textual passages. With an aggregation function a global similarity measure is composed by weighting the previously described local similarity functions of information unit cases with the parameters $(\omega_P, \omega_I, \omega_R)$ and summed up as follows:

$$Sim_{PIRI}(c_k, c_l) = \frac{\omega_P Sim_P(P_k, P_l) + \omega_I Sim_I(i_k, i_l) + \omega_R Sim_R(R_k, R_l)}{\omega_P + \omega_I + \omega_R} \quad (1)$$

If we want to compare two PIRI-snippets, then it is desirable to consider the context. For this reason we define the following extended similarity measure under the context C :

$$Sim_{PIRI+C}(c_k, c_l) = \frac{\omega_1 Sim_{PIRI}(c_k, c_l) + \omega_2 Sim_C(Cont_k, Cont_l)}{\omega_1 + \omega_2} \quad (2)$$

A similarity measure Sim_C of two documents for context comparison can be obtained from the background information provided in the documents as free text. The available documents most often contain brief background information and summaries of document objective, scope, and structure. This information can be exploited for context assessment. These passages can be compared to approximate the documents similarity using textual similarity measures [18, 25]. The specific case-based modeling of the context is in scope of future work.

2.4 Corpus of Nuclear Safety Regulations

For the initial construction of the corpus we annotated 143 documents summing up to about 17.500 pages of nuclear safety regulations as described before. The International Atomic Energy Agency (IAEA) granted their permission to be the source for all these documents [5]. Yet there are many more sources of documents in the same domain as partially summarized in Table 1.

Table 1. Overview of available documents in the domain.

Source	Language	Documents
International Atomic Energy Agency [5]	English	150
Bundesamt für Strahlenschutz [2]	German	80
Consejo de Seguridad Nuclear [4]	Spanish	100
Canadian Nuclear Safety Commission [3]	English	80
U.K. Office for Nuclear Regulation [7]	English	250
U.S. Nuclear Regulatory Commission [8]	English	2000
Autorité de Sûreté Nucléaire [1]	French	40
Ispettorato Nazionale per la Sicurezza Nucleare [6]	Italian	20

The corpus described in this work can be rebuilt following some basic steps. In the future (CRC of this paper), we will provide a ready-to-use download. All resources are available via GitHub [21]. Tools to aid in building and working on the corpus by oneself are explained in the source file or readme files. The corpus consists of three major parts. Scripts written in Java provide functionality, ontologies contain the data, a case base provides case-based similarity assessment, retrieval, and adaptation capacity. The ontology was implemented using the semantic wiki KnowWE [12]. For the case-based implementation we made use of the framework myCBR [11]. For the textual structuring of the regulatory documents an ontology was implemented [22]. Core components make use of the SKOS ontology (Simple Knowledge Organization System) [28] and the PROV ontology [26] as upper ontologies.

To access documents without annotation the present annotation information can be used as seeds for a case-based bootstrapping strategy to mine new cases [16]. This semi-supervised approach for entity and relation extraction supports the user in an active learning scenario. The algorithm compares the unknown corpus \mathcal{B} with all n-grams retrieved from the already annotated corpus \mathcal{A} . It uses different similarity measures for n-grams of different sizes. To compare large text passages a text based similarity measure like tf-idf or sentence embeddings is used, smaller information units are compared using case-based similarities. The result is the set of automatically extracted annotations most similar (to a certain threshold) to manually verified information units existent in the corpus \mathcal{A} . The user reviews the generated annotations, adjusts them if necessary, and adds them to the case base as *manually verified annotations*.

Algorithm 1: Algorithm for semi-supervised case-based bootstrapping.

Data: Annotated corpus \mathcal{A} , not annotated corpus \mathcal{B}
Result: Set of new annotations for corpus \mathcal{B} with high similarity
gram_max=length of largest annotated n-gram in \mathcal{A} ;
v=similarity switch (e.g. 10 words);
Query ontological entity labels to the corpus \mathcal{B} ;
Construct new ontology with retrieved machine annotations;
Get all annotations from the corpus \mathcal{A} in gram-size-order;
while $i < gram_max$ **do**
 if $i < v$ **then**
 | $sim_{gram} = sim_{case-based}$
 else
 | $sim_{gram} = sim_{text-based}$
 Query i-grams to machine annotations of \mathcal{B} with sim_{gram} ;
 Request user review for most similar information units;

3 Use Cases

We exemplify the previous approach by use case scenarios contained in the corpus to verify the hypothesizes stated in the introducing section. In the first part of this section the semantic retrieval is illustrated. Afterwards, we demonstrate how textual passages can be adapted using the hierarchical relation of entities. In the following, we describe the usefulness of the PIRI-approach for the generation of new document sketches.

3.1 Semantic Search and Retrieval of Documents

A semantically annotated corpus allows for semantic search of documents fitting to a given problem description. Therefore, the user query is translated into a structured query to the case base. For instance, the question “Which measures prevent a fire?” can be analyzed in a first step retrieving the contained entities *Measure*, *Fire*, and the relation *isPreventiveMeasureFor*. The most special relational triple according to their hierarchical interrelation, fulfilling the elements extracted out of the user query, is $\langle Measure \text{ isPreventiveMeasureFor } Fire \rangle$. All ontological elements containing entities of the hierarchy, that are equal or more special, should be respected and thus retrieved. For instance:

- *piri:fireWatch piri:isPreventiveMeasureFor piri:fireIncident*
- *piri:fireBarriers piri:isPreventiveMeasureFor piri:fireSpreadingIncident*
- *piri:smokeDetector piri:isPreventiveMeasureFor piri:smolderingIncident*

Following this retrieval, the user can research the particular text passages, that were annotated with these concepts.

3.2 Adaptation of Textual Passages

Text snippets can be adapted by specification or generalization of entities according to the ontological classification hierarchy. Additionally the corpus aims to support the mining of transformation knowledge. For instance, to find a process how measures suitable for certain scenarios are automatically adaptable to new and unknown incidents. An example for adaptational capacity using the hierarchical structuring of incidents and measures bases on the following textual snippet.

Example 1. “In general, the fire containment approach is preferred, since it emphasizes passive protection and thus the protection of safety systems does not depend on the operation of a fixed fire extinguishing system.” [9].

It contains the entities *passive protection*, *safety systems*, and *fixed fire extinguishing system*. The entity *passive protection* has a related entity *active protection*. The entity *fixed fire extinguishing system* has the related entity *mobile fire extinguishing system*. Those two are related in an appropriate manner. We can check the correct adaptation by simply querying for cases, where a *mobile fire fighting system* is used for *active protection*.

3.3 Generation of Document Plots

Usually, documents are created by first sketching a document plot. Here, graph-based and case-based data management complement to each other. We assume that a document about *maintenance of fire safety systems* has to be written, which also means to create a new case for this scenario [14]. In our corpus there exist two documents, one for fire safety in general and one for maintenance of power plants. In a first step, the ontological models associated to the documents are extracted and united using a convenient unification strategy [22]. The new model then contains a relevant set of entities and relations with corresponding information units. In a next step, the PIRI-snippets for the new document are generated automatically. Measures that target the same incident are accumulated into one PIRI-snippet. Afterwards, the document plot is presented to the user. The user now adapts and revises the generated plot. If needed, further user support is available with a case-based query for similar information units. For each component of the maintenance requirements special documents may give deeper advice than available in the two merged documents. We selected an interesting textual passage for fire safety containing advice for maintenance in the following example.

Example 2. “The inspection, maintenance and testing programme should cover the following fire protection measures:

- fire barrier closures such as fire doors and fire dampers;
- fire detection and alarm systems, including flammable gas detectors;
- emergency lighting systems;
- water based fire extinguishing systems;

—a water supply system including a water source and distribution pipe;
—gaseous and dry powder fire extinguishing systems;” [9]

For instance, the maintenance of the *water supply system* is in the focus of the fire safety document. The corpus contains a document subjected to the *radioactive contamination of water* which provides helpful suggestions for the safe maintenance of the water supply system. In this manner, noteworthy measures and incidents can be added to complete the generated document plot.

3.4 Results

The use cases gave a first outline how case-based reasoning can provide a natural representation for covering experience knowledge. They showed that especially the concept of similarity-based retrieval is suitable to handle incomplete input for the formulation and retrieval of experience knowledge. What exceeded the capacity of this work was to show how background knowledge could be integrated into the process in a case-based manner. Nevertheless, it was suggested how strategies basing on the similarity of free text can be intuitively integrated into a structured case-based architecture.

4 Conclusions

This paper introduced an approach for constructing a corpus exploiting graph-based, case-based, and textual information in the domain of nuclear safety regulations. It showed aspects of research work necessary in this field. A proposal for the case-based structuring of a collection of similar documents and text snippets, respectively, was made. Finally selected use cases showed how the corpus can be used practically and will be beneficial for further research work in the domain.

4.1 Related Work

An approach for information retrieval using nuclear safety ontologies was presented by Ogure et. al. [27]. Bouchet and Eichenbaum-Voline [15] presented some early work on case-based search in experience feedback reports from nuclear power plants. Ahmad et al. [10] pick out the issue that frequent changes in daily life induce an increasing amount and heterogeneity of safety-related data. Grabmair et al. [19] presented first results of a feasibility experiment to annotate documents on sub-sentence level with the goal of ranked document retrieval in a certain medical law domain.

The idea to use patterns for relation extraction dates back to Hearst [20] and was evolved by many following authors. A work by Krug et al. [25] using augmented rule-based relation extraction combining active learning with supervised strategies has inspired the presented way of corpus construction. Fundamentals for the use of case-based bootstrapping in document indexing was presented early by Brüninghaus and Ashley [16, 17]. These approaches are well known but came

into focus again meeting state-of-the-art computational and natural language processing capacity.

In this work a simplified data structure is used to assess the content of an entire document. A similar problem statement was handled by Caro-Martinez et al. [18] to use case-based strategies in an environment of incomplete information to find explanatory examples in recommender systems. An approach for the semi-automated proof of correctness of case solving strategies in the domain of German law was presented by Beck et al. [13]. This inspired aspects of this work, such as it showed capacities of a rule-based approach in a similar use case.

4.2 Future Work

For future work the annotation will be extended to more documents of the domain provided by other authorities and countries. We expect benefits from this extension, such as the chance to build a multilingual textual case-based corpus [14]. Basic applications will be evaluated more thoroughly on a broader data basis. The goal is to exploit the tacit corpus knowledge to develop more sophisticated applications. Incorporating more NLP technologies is expected to further facilitate text generation in this and similar knowledge intensive domains. Other researches are welcome to use the presented corpus for their work and to contribute to extensions of it.

4.3 Acknowledgments

We wish to thank the International Atomic Energy Agency (IAEA) for their support and the consent to use their publications as a base for this work [5].

References

1. Autorité de Sûreté Nucléaire: <https://www.asn.fr>
2. Bundesamt für Strahlenschutz: <https://www.bfs.de>
3. Canadian Nuclear Safety Commission: <https://www.cnsccsn.gc.ca>
4. Consejo de Seguridad Nuclear: <https://www.csn.es>
5. International Atomic Energy Agency: <https://www.iaea.org>
6. Ispettorato Nazionale per la Sicurezza Nucleare: <https://www.isinucleare.it>
7. U.K. Office for Nuclear Regulation: <https://http://www.onr.org.uk>
8. U.S. Nuclear Regulatory Commission: <https://www.nrc.gov>
9. Fire safety in the operation of nuclear power plants: safety guide. International Atomic Energy Agency, Vienna (2000)
10. Ahmad, J., Kostov, B., Lalis, A., Kremen, P.: Ontological foundation of hazards and risks in stamp. *Semantic Web Journal* (to appear) (2018)
11. Bach, K., Althoff, K.D.: Developing case-based reasoning applications using my-CBR3. In: Agudo, B.D., Watson, I. (eds.) *Case-Based Reasoning Research and Development*. pp. 17–31. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
12. Baumeister, J., Reutelshoefer, J., Puppe, F.: KnowWE: A semantic wiki for knowledge engineering. *Applied Intelligence* **35**(3), 323–344 (2011)

13. Beck, P.D., Iffland, M., Kronbach, J., Puppe, F., Schenke, R.: Semi-automatische Korrektur von juristischen Lösungsskizzen. *Zeitschrift für Didaktik der Rechtswissenschaft* **3**, 242–251 (01 2016)
14. Bergmann, R.: *Experience Management*. Springer, Berlin, Heidelberg (2002)
15. Bouchet, J.L., Eichenbaum-Voline, C.: Case-based reasoning techniques applied to operation experience feedback in nuclear power plants. In: Smith, I., Faltings, B. (eds.) *Advances in Case-Based Reasoning*. pp. 497–511. Springer Berlin Heidelberg, Berlin, Heidelberg (1996)
16. Brüninghaus, S., Ashley, K.D.: Bootstrapping case base development with annotated case summaries. In: Althoff, K.D., Bergmann, R., Branting, L. (eds.) *Case-Based Reasoning Research and Development*. pp. 59–73. Springer Berlin Heidelberg, Berlin, Heidelberg (1999)
17. Brüninghaus, S., Ashley, K.D.: Reasoning with textual cases. In: Muñoz-Ávila, H., Ricci, F. (eds.) *Case-Based Reasoning Research and Development*. pp. 137–151. Springer Berlin Heidelberg, Berlin, Heidelberg (2005)
18. Caro-Martinez, M., Recio-Garcia, J.A., Jimenez-Diaz, G.: An algorithm independent case-based explanation approach for recommender systems using interaction graphs. In: Bach, K., Marling, C. (eds.) *Case-Based Reasoning Research and Development*. pp. 17–32. Springer International Publishing, Cham (2019)
19. Grabmair, M., Ashley, K.D., Chen, R., Sureshkumar, P., Wang, C., Nyberg, E., Walker, V.R.: Introducing luima: An experiment in legal conceptual retrieval of vaccine injury decisions using a uima type system and tools. In: *Proceedings of the 15th International Conference on Artificial Intelligence and Law*. p. 69–78. ICAIL '15, Association for Computing Machinery, New York, NY, USA (2015)
20. Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora. In: *COLING 1992 Volume 2: The 15th International Conference on Computational Linguistics*. pp. 539–545 (1992)
21. Korger, A., Baumeister, J.: GitHub: <https://github.com/regdoc/piri>
22. Korger, A., Baumeister, J.: The SECCO ontology for the retrieval and generation of security concepts. In: Cox, M.T., Funk, P., Begum, S. (eds.) *ICCB. Lecture Notes in Computer Science*, vol. 11156, pp. 186–201. Springer (2018)
23. Korger, A., Baumeister, J.: Case-based retrieval and adaptation of regulatory documents and their context. In: *LWDA Berlin*. pp. 292–303 (2019)
24. Korger, A., Baumeister, J.: Case-based generation of regulatory documents and their semantic relatedness. In: Arei, K., Kapoor, S., Bhatia, R. (eds.) *Future of Information and Communication Conference San Francisco. Advances in Information and Communication*, vol. 1130, pp. 91–110. Springer (2020)
25. Krug, M., Reger, I., Jannidis, F., Weimer, L., Madarász, N., Puppe, F.: Overcoming data sparsity for relation detection in german novels. In: *DHd 2017 Digital Humanities: multimedial & multimodal*, Montreal. pp. 490–493 (2017)
26. Moreau, L., Groth, P.: *Provenance: An Introduction to PROV*. Synthesis Lectures on the Semantic Web: Theory and Technology, Morgan and Claypool (2013)
27. Ogure, T., Furuta, K.: Information retrieval using ontology for sharing knowledge on safety. In: Spitzer, C., Schmocker, U., Dang, V.N. (eds.) *Probabilistic Safety Assessment and Management*. pp. 531–536. Springer London, London (2004)
28. W3C: SKOS Simple Knowledge Organization System Reference: <http://www.w3.org/TR/skos-reference> (August 2009)

Visualizing the behavior of CBR agents in an FPS Scenario

Philipp Yasrebi-Soppa¹, Jobst-Julius Bartels¹, Sebastian Viefhaus¹, Pascal Reuss^{1,2}, and Klaus-Dieter Althoff^{1,2}

¹ University of Hildesheim
Samelsonplatz 1
31141 Hildesheim

{reusspa, bartelsj, viefhaus, yasrebi}@uni-hildesheim.de

² German Research Center for Artificial Intelligence (DFKI)
Trippstadter Str. 122
67663 Kaiserslautern
kalthoff@dfki.uni-kl.de

Abstract. The analysis and visualization of agent behavior enables a developer to identify unexpected or faulty behaviors and can show room of improvement. Therefore, visualization tools can be helpful to analyze behaviors during or after simulations. This paper presents a visualization tool VISAB developed for analyzing and visualizing the movement and actions of CBR agents in a first-person scenario. We describe the settings of the scenario and in more detail the visualization possibilities to get a better understanding of agent behavior during game-play.

Keywords: Visualization · Case-Based Reasoning · Agent behavior · Multi-Agent System · First-Person Scenario

1 Introduction and motivation

Visualization of agent behaviors can also be helpful for teaching Artificial Intelligence (AI). It can be used to support students and inexperienced AI developers during knowledge modeling and designing decision making processes for agents. The goal of this work was to create a visualization tool to display agent behavior in a first person scenario. This scenario is part of a platform for teaching Case-based Reasoning (CBR), learning software agents and multi-agent systems (MAS) during an advanced programming practical. The basic idea of this platform is to have several modules that represent different scenarios that can be played and solved by software agents. The students design and implement an agent or a team of agents with a CBR system (or other AI technologies that enables decision making) for one of the given scenarios. The implemented agent then plays the scenario and the behavior of the agent will be analyzed and visualized to

Copyright © 2020 by the paper's authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

give feedback to the students. The students should be able to watch their agent while playing and get an evaluation after completing the scenario. In addition to play a scenario solo, it will be also possible to compete with the "home team" of agents in directly competitive scenarios. Figure 1 gives an overview of the desired architecture of the teaching platform.

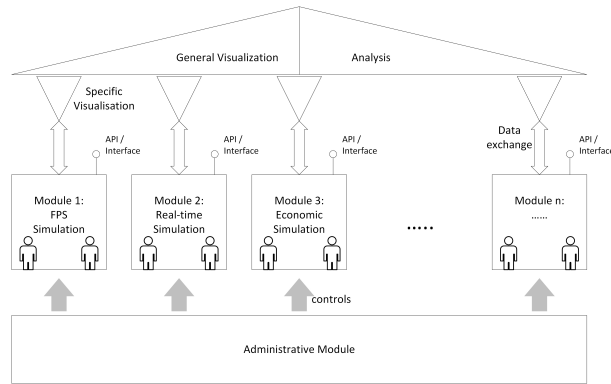


Fig. 1. Overview of the planned platform architecture

Currently, several modules are being implemented and in different states of completion: a first-person game, a real-time strategy game, an economic simulation, and several board game implementations like Settlers of Catan. In addition to these playable scenarios, an administration module and a first visualization module for the first-person game are under development. The visualization component records a game and enables the user to view the game later with several options to configure the displayed information.

The visualization of agent behavior in given environments with defined surrounding conditions enables a developer to identify unexpected or faulty behaviors and can show points of improvement. In addition to visualizing the behavior of the agent itself, background and contextual information can be visualized, too. Using this additional information, agent behaviors and decisions can be analyzed and the decision-making process of an agent will become more comprehensible to the developer and can be better optimized. [1][5][9]

This paper provides an overview of the visualization module and the implemented features. The remaining paper is structured as follows. Section 2 gives an overview of some related work in the field of agent behavior visualization. Section 3 describes the first-person scenario (FPS) and then in detail the visualization module for the participating agents, while Section 4 provides an overview of the performed evaluation. The paper concludes in Section 5 with a short summary and an outlook to future work.

2 Related Work

Over the last decades many research activities and approaches for modeling and visualizing the behavior of software agents in different use cases were realized. To prove the functionality of an artificial intelligence, multi-agent systems became a popular application area. We focus on the FPS domain, a sub-genre of action video games. In a typical FPS game, there are two teams of typically human players trying to overcome the opposing team by either eliminating each member of the opposing team or by successfully complete another objective, such as planting a bomb at a certain place or by preventing the opposing team to do so. While both teams are actively playing at the same time, most FPS are limited by a round-time of approximately five minutes and a limited map size.

Agent modeling frameworks like NetLogo[25], REPAST[11], or Pogamut[5] provide simulation and visualization capabilities beside their core design and modeling features. NetLogo was developed in 1999 and is still an active multi-agent environment today. It enables a developer to simulate complex situations with several thousand of agents in 2D or 3D. In addition, NetLogo offers to visualize several background information and to analyze the agent models with the help of different diagrams. An interesting feature of NetLogo comes with the extension HubNet. It enables the development and execution of participatory simulation for lectures. In these simulations every participating student controls a part of the overall system, for example a traffic light in a traffic simulation. [26][23]

The Recursive Porous Agent Simulation Toolkit (REPAST) is an open source framework for modeling and simulation agents. Similar to NetLogo it offers the modeling and simulation of agents in 2D and 3D environments and several visualization features with diagrams and graphs.[11] REPAST was developed further into two different version, REPAST Symphony and REPAST for high performance computing (REPAST HPC). REPAST Symphony offers additional visualization capabilities with the help of a geographic information system to visualize the movement of agents.[2][16]

Pogamut is another free development environment for modeling and simulation agent behavior in a 3D environment. It is used mainly in combination with the Unreal game series and is designed to support research and education. The current version of Pogamut was released in 2015[18]. During the debugging of implemented agents, Pogamut allows the visualization of several information, for example the position of agents on a map, the view direction of agents, and the planned movement.[6][7] In addition to development frameworks with visualization capabilities, there are several more or less pure visualization tools, that a designed with the main purpose to test and evaluate the behavior of software agents.

GameBots is a virtual testing environment to evaluate software agents in games. The goal was to develop a tool that can be used on computer games to enable the use of these games for research and education in the fields of AI and MAS. The GameBots tool provides three components to visualize background information about agents and their environment: a 3D virtual world, a global

VizClient for analyzing the entire simulation, and a local VizClient for analyzing the behavior of a single agent. With these three components, GameBots is able to visualize positions, view directions and field, movement, points, messages, and decision of the individual agents and agent teams.[1][20]

The Unreal Tournament Semi-Automated Force (UTSAF) is a military-based agent simulation in the 3D environment of Unreal Tournament. In the context of UTSAF, an agent can be a ground or an air unit. UTSAF uses the tool GameBots in combination with special information brokers to visualize agent behavior. These information broker modules can collect the information about individual agents, agent teams, or the entire environment and passes them to different instances of the GameBots tool. This enables the user to act as a spectator for the simulation and get an overview of the entire environment or see the simulation through the eyes of an individual agent.[12][17]

Lithium is a tool that was developed to enable analyses in multiplayer computer games. The tool visualizes information via overlays on top of the running game. Lithium was designed to analyze the entire situation of a computer game, and not specific information about a single agent. The goal is to capture the entire dynamic of the computer game. Lithium can visualize information on a local and global level. Local level information is based on specific positions or parts of the environment or specific events, while the global level is used to visualize trends or behaviors over time. The tool provides information about the position and movement of agents, combat behavior, and agent views on the local level. On the global level, Lithium can display information about the agent density on a specific part of the map, the medicine density and needs, control areas for specific teams, and combat information like fire support ranges.[9]

HeapCraft is a free tool for visualization and data search with a focus on the analysis of agent behavior in interactive virtual worlds. The tool can be used by administrators and players of multiplayer game servers and aims at changing a player's behavior in positive and social way. In addition, problems in the game world and with player activities can be identified and failure diagnoses can be performed.[13] A prominent game HeapCraft is applied to, is the 3D computer game Minecraft, but can be applied to various games with virtual worlds. The tool provides several components that can be integrated into a game as plug-ins. The Epilog Dashboard provides visualizations and analyses about player behavior in real time. In context of Minecraft the dashboard visualizes for example the online time, the covered distance, build blocks, and gathered resources. In addition, the dashboard computes an index for collaboration with other players. With the help of the Map Miner player activities can be tracked, analyzed, and visualized on a 2D map. The plug-in Classify is able to analyze the behavior of one player over a complete day and visualizes the information in form of diagrams. [14][15]

The Visualization Toolkit for Agents (VISTA) is a framework to visualize the internal reasoning processes of software agents. It aims at evaluating the behavior and decision making process of agents and can be used during or after a simulation.[21] VISTA was developed with four purposes: providing a generic

framework capable to be used in as many agent architectures and systems as possible, providing a domain-independent framework, enable the tracking of internal reasoning processes, and provide visualizations of agent behaviors during run-time as well as after the termination of a simulation by recording the behaviors. For the visualization of the internal reasoning processes, VISTA uses a so-called Situational Awareness Panel (SAP) to collect and display all information about the agents, their interactions, and communications. In addition, VISTA is capable of generating explanations for the behavior of agents with focus on objects and situations.[21][22]

3 Visualization of CBR agent behavior

This section describes first briefly the developed game with the FPS scenario. A more detailed description can be found in [8] and [19]. Then we will describe in more detail the conceptual idea of the visualization component called VISAB (Visualization of Agent behavior).

3.1 Settings of the FPS scenario

The FPS scenario was developed as a multi-agent system and consists of three components: the multi-agent system itself, the game logic and visualization component, and a CBR component. The game component was developed with Unity 3D[24], while the multi-agent system was implemented using Boris.Net[4]. The CBR component was modeled and implemented using the open source tool my-CBR[3]. The Unity 3D framework was used to design the environment in which the software agents compete each other and to visualize the actions of the agents to the user. The agents are implemented within the Unity 3D component with the help of Boris.Net. There are three different agents implemented: the player agent, the planning agent, and the communication agent. The *player agent* gets an update of the situation through the Unity framework. With each sensor update, the *player agent* sends the information to the *communication agent*. This agent transforms the received data into a JSON string and passes it to the CBR component. The CBR component performs a retrieval and answers the request of the communication agent by sending the most similar cases back, also in a JSON string. The solution of the cases contain possible plans that can be executed by the *player agent*, but have to be transformed into Unity 3D specific orders to be executable. Therefore, the retrieved solutions are passed to the *planning agent*. This agent evaluates the received solutions and forms a plan that is passed to the *player agent*. The *player agent* then executes the new plan. If no new plan can be formed, the *player agent* continues executing the current plan.

The case structure modeled within the CBR component consists of a situation description based on the sensor input of the *player agent* and associated actions that form an executable plan. The situation description consists of seventeen attributes with various data types. The attributes have integer, symbolic, or Boolean data types. The distance to an entity in the game is represented by four

symbolic values: near, middle, far, and unknown. This way the similarity measure is less complex. Is the distance to an entity less than 15 Unity scale units, the distance is considered near, between 15 and 30 scale units the distance is set to middle, and between 30 and 50 scale units the distance is set to far. If the distance is greater than 50 scale units or the position of an entity is unknown, the distance is set to unknown. The developed FPS scenario was extended with a reinforcement learning (RL) approach to enable the CBR component to learn from different situations and the executed plans. Therefore, a reward function is used to calculate the win probability of a situation based on several important attributes. The results showed that the integration of an RL approach into the system leads to an improvement in the performance of the CBR agent. More details on the multi-agent system, the individual agents, their relationships, and the reinforcement learning approach can be found in [10].

3.2 Visualization tool for the FPS scenario

The analyses of the CBR agents performance were made by logging information about victory and defeat in a CSV file and by observing the live game-play. But this kind of analysis is not very sufficient and incorporates only few available information. Especially during observation in the live game-play, important actions can be missed. Therefore, a visualization component was required that was capable of collecting all relevant information during the game-play, aggregating and evaluating these information, and display them to the developer to understand the behavior of the agents. Many existing visualization tools were reviewed to find a suitable and applicable tool or framework that could be used in our use case. But none of the reviewed tools fit complete for our application and the desired platform. Especially the requirement of visualizing CBR specific information and the planned application of the visualization component to all planned modules of our platform could not be covered by the existing tools from our point of view. Some tools could be used only partially, other are not actively developed anymore. This led to the development of our own solution VISAB.

The architecture of VISAB is a classic three-layered model. The presentation layer is the interface to the user and contains the visual and graphical components to display the information about the agents. The logic layer is arranged under the presentation layer and is responsible for data processing and the preparation of information for the visualization. The components of the presentation layer are controlled and managed by the logic layer. In addition, the logic layer also enables the processing of user interactions and information requests. The undermost layer is the data layer that provides the data streamed from the game during live game-play or from a recorded file. The data layer loads and saves the data from text files with a format based on keys and values. It is the connection to the Unity game platform. Figure 2 gives an overview of the VISAB architecture.

VISAB requires the game data in a JSON similar format. This game data is stored in a text file and contains eighteen different properties. The value of each

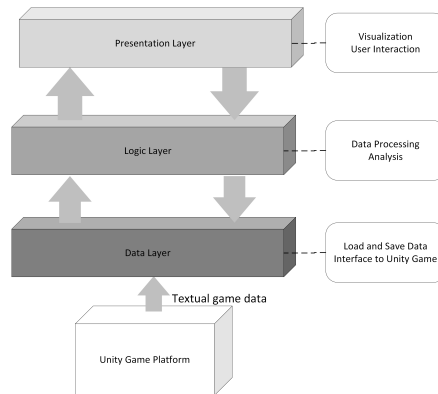


Fig. 2. Overview of the VISAB architecture

property is stored for each frame of a game cycle. The properties are listed in the following:

- **ammoPosition** - the positions of ammunition crates
- **coordinatesCBRBot** - the coordinates of the CBR agent
- **coordinatesScriptBot** - the coordinates of the scripted agent
- **healthCBRBot** - the current health of the CBR agent
- **healthScriptBot** - the current health of the scripted agent
- **healthPosition** - the positions of health containers
- **nameCBRBot** - the name of the CBR agent
- **nameScriptBot** - the name of the scripted agent
- **planCBRBot** - the current plan of the CBR agent
- **planScriptBot** - the current plan of the scripted agent
- **roundCounter** - a counter for the current round
- **statisticCBRBot** - victories and defeats of the CBR agent
- **statisticScriptBot** - victories and defeats of the scripted agent
- **weaponCBRBot** - current weapon of the CBR bot
- **weaponScriptedBot** - current weapon of the scripted agent
- **weaponMagAmmoCBRBot** - current ammunition for the equipped weapon of the CBR agent
- **weaponMagAmmoScriptedBot** - current ammunition for the equipped weapon of the scripted agent
- **weaponPosition** - the positions of weapons

A VISAB file can contain any type of information if it has the format $[key = value]$, for example $[weaponCBRBot = Pistol]$. The generic statistics visualization component of VISAB displays any information in the given format. In addition to a VISAB file, the game data can also be provided during live game-play using a data stream from the Unity game platform.

The presentation layer has two main perspectives to display the visualizations based on the data. The first one is a generic statistics overview of the provided

data. The properties and their values are displayed in a table to give the user overview of the available data. In addition, for each agent a diagram with the executed plans for each agent is displayed using the processed data. The diagram shows how often a specific plan is executed during game-play as well as the total number of all executed plans. Figure 3 shows the statistic perspective of VISAB with some sample data.

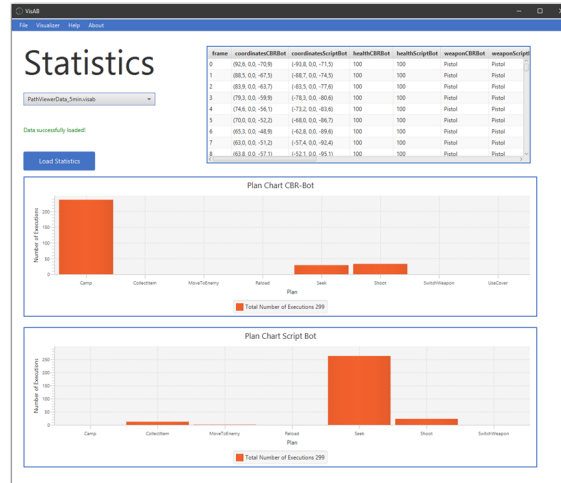


Fig. 3. Statistic perspective of VISAB

The second perspective is the so-called PathViewer and the heart of VISAB. It allows a detailed visualization of a game with focus on different aspects of the game and agent behaviors. The PathViewer consists of several elements: a map of the level, where the game took place, two tables with the detail information of the playing agents, a configuration panel, and a time frame. In the map, all information of the game can be visualized by symbols and paths and therefore displays a replay of the game from a bird's eye perspective. The configuration panel allows the user to enable or disable the visualization of certain information and serves also as a legend for the displayed information. The time frame is used to control the playback of the recorded game. This can be done by playing the complete game or by selecting a specific frame that should be displayed. Figure 4 shows the PathViewer perspective with sample data from a fifteen minute game-play at the beginning of the visualization.

To make use the PathViewer, at first a User has to select a data file or a live stream that should be visualized (1). Then the data is loaded (2) and displayed in the two tables (3). In the next step, the user selects the information to be visualized or accept the default configuration (4). The information will be displayed in the map (5) after starting the visualization (6). During the

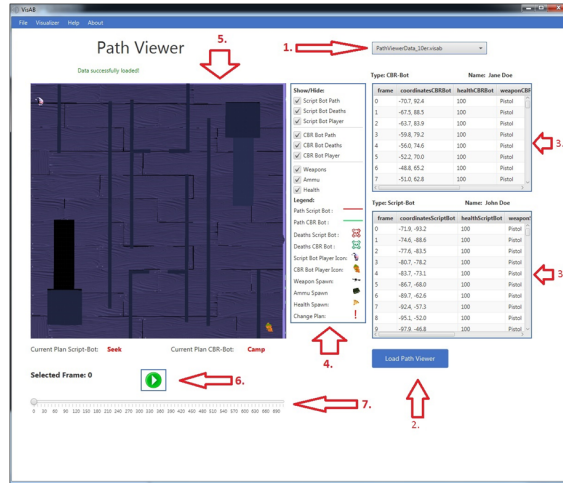


Fig. 4. PathViewer perspective of VISAB at the start of the visualization

visualization, the user can always see the current displayed frame or select a specific frame (7). The single steps are marked in Figure 4.

During the visualization, different information can be found on the map. First the path of the playing agents will be displayed to retrace the routes during a game session. Along the paths several icons can be found. The current position of the agents is also displayed along the routes for every frame. This allows to see the detailed movement on the routes during playback. Every time an agent is defeated a symbol is placed on the map to visualize the situation. In addition, every time a weapon, ammunition, or health is spawning, the corresponding icon is displayed on the map. If it is collected, the icon disappears. At least, another important information can be visualized: the point during a game, when an agent changes his action plan. This is displayed using an exclamation mark to make the specific situation visible. Using the visualized information on the map and the detail information about the agent status and behavior in the two tables, a user can retrace and analyze agent behavior and can identify situations with bad or even wrong agent behavior. This allows a better adaptation of agent behavior than just viewing the live game without the contextual information or just using the logged CSV file. Figure 5 shows the PathViewer with several information visualized in the map.

4 Evaluation

The evaluation of VISAB was performed to test the correct and complete visualization of the analyzed information. In addition, the performance while displaying different kinds of information were tested. To verify the correctness and

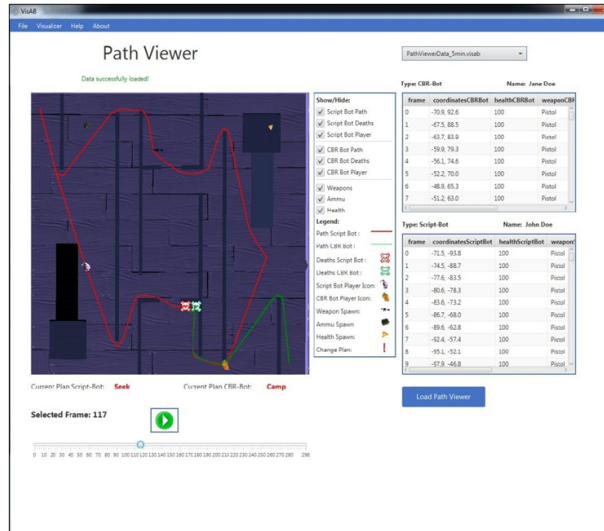


Fig. 5. PathViewer perspective of VISAB with visualized information on the map

completeness of the displayed information, a live game was directly compared with the visualized information in VISAB. The game-play was recorded by using a the ReLive function of a RADEON graphic card. This recorded game-play was then compared with the VISAB visualization, frame by frame. For the verification three game-play with a length of ten, twenty, and thirty minutes were recorded and compared. The result was that every occurred entity and situation during the game-play was recorded complete and correctly. The performance of VISAB was evaluated to test the handling of big data sets recorded during long game-play sessions. This was done with two data sets for a game-play of 90 minutes. The tests showed only minimal delays when selecting a specific frame with the slider. Therefore, the performance is more than sufficient for the planned use cases.

An evaluation within lectures with participating students is planned for the next semester. The goal will be to evaluate the impact of VISAB on the analysis and development process of the students.

5 Conclusion and Outlook

This paper presents a visualization tool for agent behavior in an FPS scenario to analyze and identify bad or wrong behavior of the participating agents. We described the settings of the FPS scenario and then the idea, architecture, and current realization of the visualization tool VISAB. We also have conducted a small successful evaluation on the features of VISAB. The next steps for developing VISAB further, are to evaluate VISAB with students and extend the

PathViewer with more information of the internal reasoning processes, like case similarities. We are also currently re-implementing the FPS scenario for team play and other game modes like capture the flag. Therefore, we will also adapt the PathViewer to display the additional information generated by the new situations. Adding new perspectives to enable the visualization of the other modules are also planned.

References

1. Adobbati, R.; Marshall, A. N.; Scholer, A.; Tejada, S.; Kaminka, G.; Schaffer, S.; Sollitto, C.: GameBots: A 3D Virtual Test-Bed for Multi-Agent Research. In: WAGNER, T. (Ed.); Rana, O.S. (Ed.): Proceedings of the second international workshop on Infrastructure for Agents, MAS, and Scalable MAS, Volume 5, Montreal, Canada, 2001.
2. Argonne National Laboratory, Webseite 2015. - <https://sourceforge.net/projects/repast/>; last verification: 16.06.2020
3. Bach, K.; Sauer, C.; Althoff, K.-D.; Roth-Berghofer, T.: Knowledge Modeling with the Open Source Tool myCBR. In: Nalepa, G.J. (Ed.), Baumeister, J. (Ed.), Kaczor, K. (ed.): Proceedings of the 10th Workshop on Knowledge Engineering and Software Engineering (KESE10), Prague, Czech Republik, 2014
4. Bojarpour, A.: Boris.Net, Website, 2009 - <http://www.alibojar.com/boris-net>, last verification: 16.06.2020.
5. Gemrot, J.; Kadlec, R.; Bida, M.; Burkert, O.; Pibil, R.; Havlicek, J.; Zemak, L.; Simlovic, J.; Vansa, R.; Stolba, M.; Plch, T.; Brom, C.: Pogamut 3 can assist developers in building AI (not only) for their videogame agents. In: Dignum, F. (Ed.); Bradshaw, J. (Ed.); Silverman, B. (Ed.); Doesburg, W. (Ed.): Agents for games and simulations, Springer-Verlag Berlin, Heidelberg, 2009, S. 1-15.
6. Gemrot, J.; Brom, C.; Kadlec, R.; Bida, M.; Burkert, O.; Zemcak, M.; Pibil, R.; Plch, T.: Pogamut 3 – Virtual humans made simple. In: Srinivasan, N. (Ed.); Gupta, A. K. (Ed.); Pandey, J. (Ed.): Advances in Cognitive Science, SAGE Publications Pvt. Ltd, 2010, S. 211-243.
7. Gemrot, J.; Brom, C.; Plch, T.: A periphery of pogamut: From bots to agents and back again. In: Dignum, F. (Ed.): Agents for games and simulations II, Springer-Verlag Berlin, Heidelberg, 2011, S. 19-37.
8. Hillmann, J.: “Konzeption und Entwicklung eines Prototypen für ein lernfähiges Multi-Agenten-System mittels des fallbasierten Schließen im Szenario einer First-Person Perspektive” (Conception and Development of a prototype for a multi-agent-system with learning capabilities using case-based reasoning in the first-person perspective szenario). Hildesheim, University of Hildesheim, 2017.
9. Hoobler, N.; Humphreys, G.; Agrawala, M.: Visualizing competitive behaviors in multi-user virtual environments. In: Visualization, IEEE, 2004, S. 163-170.
10. Kolbe, M.; Reuss, P.; Schoenborn, J.M.; Althoff, K.-D.: Conceptualization and Implementation of a Reinforcement Learning Approach Using a Case-Based Reasoning Agent in a FPS Scenario, In: Jaeschke, R. (Ed.); Weidlich, M. (Ed.): Proceedings of the Conference “Lernen, Wissen, Daten, Analysen”, Berlin, 2019
11. Macal, C. M.; North, M. J.: Tutorial on agent-based modeling and simulation. In: Simulation Conference, Proceedings of the Winter, 2005, S. 73-83.
12. Manojlovich, J.; Prasithsangaree, P.; Hughes, P.; Chen, J.; Lewis, M.: UTSAF: A Multi-Agent based Framework for Supporting Military-based distributed interactive

- simulations in 3D virtual environment. In: Simulation Conference, Proceedings of the Winter, Volume 1, 2003, S. 960-968.
13. Müller, S.; Solenthaler, B.; Kapadia, M.; Frey, S.; Klingler, S.; Mann, R. P.; Summer, R. W.; Gross, M.: HeapCraft: Interactive Data Exploration and Visualization Tools for Understanding and Influencing Player Behavior in Minecraft. In: Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games, 2015, S. 237-241.
 14. Müller, S.; Kapadia, M.; Frey, S.; Klingler, S.; Mann, R. P.; Solenthaler, B.: Statistical Analysis of Player Behavior in Minecraft. In: Zagal, J. (Ed.): Proceedings of Conference on Foundations of Digital Games, 2015.
 15. Müller, S.; Frey, S.; Kapadia, M.; Klingler, S.; Mann, R. P.; Solenthaler, B.; Summer, R. W.; Gross, M.: HeapCraft: Quantifying and Predicting Collaboration in Minecraft. In: Proceedings of the eleventh AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, 2015, S. 156-162.
 16. North, M. J.; Collier, N. T.; Ozik, J.; Macal, C. M.; Tatara, E. R.; Bragen, M.; Sydelko, P.: Complex adaptive systems modeling with Repast Sim-phony. In: Complex adaptive systems modeling, Volume 1, 2013.
 17. Prasithsangaree, P.; Manojlovich, J.; Hughes, S.; Lewis, M.: UTSAF: A Multi-Agent-Based Software Bridge for Interoperability between Distributed Military and Commercial Gaming Simulation. In: The Society for Modeling and Simulation International, 2004, S. 647-657.
 18. Pogamut, Webseite 2015. - http://pogamut.cuni.cz/main/tiki-view_blog.php?blogId=3; last verification: 16.06.2020.
 19. Reuss, P., Hillmann, J., Vieffhaus, S., Althoff, K.-D.: "Case-Based Action Planning in a First Person Scenario Game". In: Rainer Gemulla, Simone Ponzetto, Christian Bizer, Margret Keuper, Heiner Stuckenschmidt (Ed.). LWDA 2018 - Lernen, Wissen, Daten, Analysen - Workshop Proceedings. GI-Workshop-Tage "Lernen, Wissen, Daten, Analysen" (LWDA-2018) August 22-24 Mannheim Germany University of Mannheim 8/2018.
 20. SIOUTIS, C.: Reasoning and Learning for intelligent Agents. Phd Thesis, University of South Australia, 2006.
 21. Taylor, G. E.; Jones, R. M.; Fredriksen, R. A.: VISTA: A Generic Toolkit for Visualizing Agent Behavior. In: Proceedings of the 11th Conference on Computer Generated Forces and Behavioral Representation, 2002, S. 157-167.
 22. Taylor, G. E.; Knudsen, K.; Holt, L. S.: Explaining Agent Behavior. In: Behavior Representation in Modeling and Simulation (BRIMS), 2006.
 23. Tisue, S.; Wilensky, U.: NetLogo: Design and implementation of a Multi-Agent Modeling Environment. In: Proceedings of agent, Volume 2004, S. 7-9.
 24. Unity Technologies: Unity 3D Overview, Website, 2018 - <https://unity3d.com/de/public-relations>, last verification: 16.06.2020.
 25. Wilensky, U.: NetLogo Frogger model. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL, Webseite 2002. - <http://ccl.northwestern.edu/netlogo/models/Frogger>; last verification: 16.06.2020.
 26. Wilensky, U.: NetLogo. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL, Webseite 2017. - <http://ccl.northwestern.edu/netlogo/>; last verification: 16.06.2020.

Development and Implementation of a Case-Based Reasoning Approach to Speed-Up Deep Reinforcement Learning through Case-Injection for AI Gameplay

Marcel Heinz¹, Jakob M. Schoenborn^{1,2} and Klaus-Dieter Althoff^{1,2}

¹ University of Hildesheim

Universitätsplatz 1, 31141 Hildesheim, Germany

{heinzm, schoenb}@uni-hildesheim.de

² German Research Center for Artificial Intelligence (DFKI)

Trippstadter Str. 122, 67663 Kaiserslautern, Germany

kalthoff@dfki.uni-kl.de

Abstract. Game environments offer properties that are useful for researching challenges in artificial intelligence (AI). Gaming enables testing, evaluation, and preparation of new methods for real-world scenarios. Reinforcement learning (RL) has undergone enormous further development in the recent years. The usage of artificial neural networks makes it possible to use reinforcement learning algorithms in complex environments. To learn feasible solutions, RL agents have to interact with the environment and learn based on their experience. Many scenarios require long training times and a vast amount of training data. Reusing previously experience knowledge can be the key to shortened training cycles and improved performance. Case-based reasoning (CBR) is another methodology of artificial intelligence using experiences from previous situations for solving new situations by adapting known solutions. Therefore, CBR appears to be particularly suitable for knowledge transfer in the area of reinforcement learning and is applied to improve the learning process of RL agents within video games. First, this work develops a theoretical approach in order to show in a second step the practical feasibility with the help of a prototypical implementation. The evaluation of the proposed method confirms reduced training time and improved performance.

Keywords: Case-Based Reasoning, Case-Injection, Reinforcement Learning, Transfer Learning, Gaming

1 Introduction

With the development of autonomous vehicles [8], automated diagnoses of diseases [7], and rapidly evolving language assistants, the subject of AI continues

Copyright © 2020 by the paper's authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

to appear in everyday life. For companies, research institutions or countries, AI has become a panacea for many areas and promises solutions to urgent questions of the future. To solve these problems, a variety of different approaches have been steadily researched, tested, and evaluated with techniques such as RL, (Deep) Neural Networks (DNN) and CBR among a multitude of other approaches as well. We developed and implemented intelligent agents that are able to reuse knowledge in a proficient way to overcome some of the obstacles of RL agents, such as accelerating the learning phase. These hurdles are the limitation of the learned task and the costly process of learning. RL agents still apply their knowledge in a rather small operating area. In order to work in other games, it has to be retrained, which is cost-intensive and time-consuming. A combination of CBR, deep learning (DL) and RL together with the concepts of multi-agent systems (MAS) and transfer learning (TL) can lead to a sophisticated solution that overcomes the outlined difficulties.

The core of this work is the development of a general procedure based on the CBR methodology for knowledge transfer within a DRL environment. The underlying work examines the structure of a systematic transfer learning approach, in which, as far as possible, all process steps run automatically. In order to build the bridge from theory to practice, the proposed algorithm is prototypically implemented³ and then evaluated.

2 Related Work

This section reviews research that is related to our new approach. Since the topics CBR and DRL are in any case of great importance in AI research, the main focus of this section is on the area of games.

Bianchi et al. examined the knowledge transfer through CBR components within RL environments [3]. The main focus of the work is a case-based policy inference algorithm. This accelerates the learning process through an intelligent selection of similar, already learned cases within the case base. The publication transfers the existing knowledge within a Q-learning environment. In contrast, this work applies knowledge transfer to a DRL agent. The proposed algorithm shows success in terms of temporal performance and more robust learning behavior. The more similar cases found, the lower ϵ can be chosen and thus influences the exploration-exploitation behavior [3]. The work uses a fixed value for ϵ . A grid-world environment evaluates the proposed algorithm, in which an agent starts at any randomly chosen point and has to navigate to a target point. Based on this situation, the similarity measure consists of the Euclidean distance from the start to the destination. In contrast, the suggested architecture of this work calculates the similarity of visual and numerical inputs. In the experiment phase, initial cases are generated, which are then reused in the later transfer of knowledge. During the learning process, the cases from the case base are evaluated and the best case is used. Bianchi et al. showed that their approach achieves

³ Code available: <https://github.com/marcel-heinz/peng>

the same amount of reward as conventional Q-learning approaches in a shorter time.

In the work of J. Hillman [4], RL improves the learning process with the help of CBR and TL in a first-person shooter scenario. Hillman points at a practical benefit for the use in real-world scenarios in which the state space grows enormously. RL is used within the source domain to generate various cases for the later target domain. The similarity measure for the retrieval is calculated from the distance of the agent to the objects within its environment. In addition, the algorithm accesses the case base within each episode after each step to find a suitable case. If the retrieval is not set optimally, this can lead to performance losses in terms of required time. The algorithm presented here leads to a significant improvement within the realized domain. Kolbe et al. introduce an approach that combines RL and CBR to improve the performance of an first person shooter (FPS) agent [5]. For this purpose, Kolbe et al. build on the knowledge generated by Hillmann [4] and develop a MAS inhabiting three interacting agents in a Unity 3D game environment. To ensure that case retrieval delivers useful cases, a RL agent was also implemented, which delivers the appropriate actions from the case base on the basis of stored sequences. The variable reward function, which is useful for this promising combination, adapts to the saved sequences and the overall win chance. In our approach, the retrieval process is supported by a RL component, which increased the performance in the conducted tests.

3 Past Experience Network for Gameplay (PENG)

This section lays the theoretical basis for the subsequent development of the procedure. Besides, this section presents the Past Experience Network for Gameplay (PENG) process model, which reflects the basic features of an intelligent agent architecture that reuses knowledge in the Atari2600 gaming environment. Atari is well-known for famous arcade games such as Pacman and Breakout. The PENG system builds on the basic principles of the CBR methodology and fulfills the following characteristics: Manual input for game setup, flexible application options with different domains, dealing with visual game environments, MAS approach and evaluation of trained agents to identify a suitable agent. Figure 1 shows the proposed CBR cycle of the PENG architecture.

3.1 Nomenclature

As part of this work, a model repository \mathcal{R} equals the case base. Also, in the PENG process model, a case τ is divided into two parts, problem and solution. Equation 1 formalizes the model repository and equation 2 describes an individual case.

$$\mathcal{R} := \{\tau_0, \tau_1, \dots, \tau_n\} \tag{1}$$

$$\tau := (\Lambda_\tau, \mu_\tau) \tag{2}$$

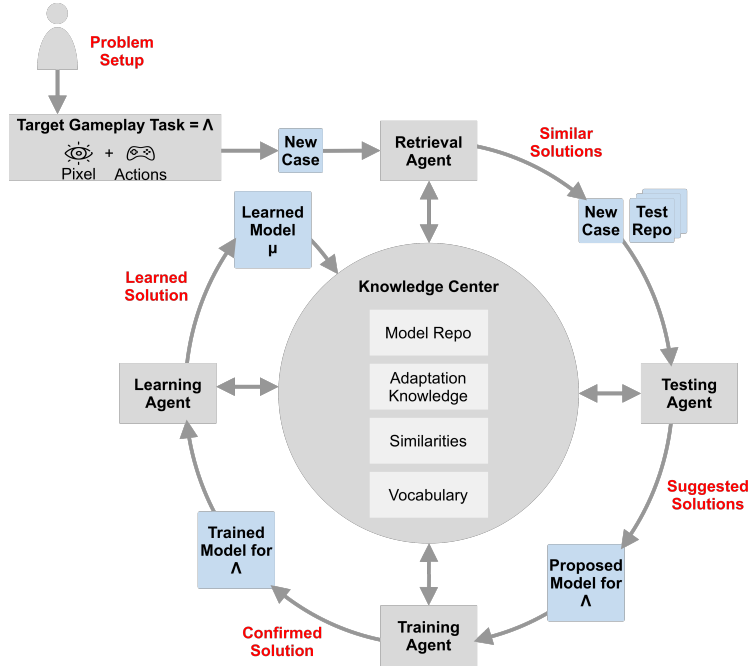


Fig. 1. Proposed CBR cycle of PENG, based on [1]

With the target gameplay task Λ and the model μ as in Equation 3 and 4

$$\Lambda_\tau := (\text{Pixel Values, Number of Actions}) \quad (3)$$

$$\mu_\tau := (\text{NN Architecture, Policy}) \quad (4)$$

The standard CBR methodology retains relevant knowledge within the knowledge base. Within this knowledge base are the containers for general knowledge such as adaptation knowledge or similarity measures. The PENG method stores all available knowledge inside the knowledge center (KC). Figure 2 briefly describes the general process for the PENG architecture. The entire process is described in detail in the following.

3.2 Similarity Measure

The similarity measure is an integral part of the overall PENG architecture. As described above, the PENG component accesses the gameplay and nb_action attributes. The local similarity measures should be designed in such a way that they cover the particular characteristics of the individual attributes. Therefore, the local similarity measure at the attribute level should have the following properties:

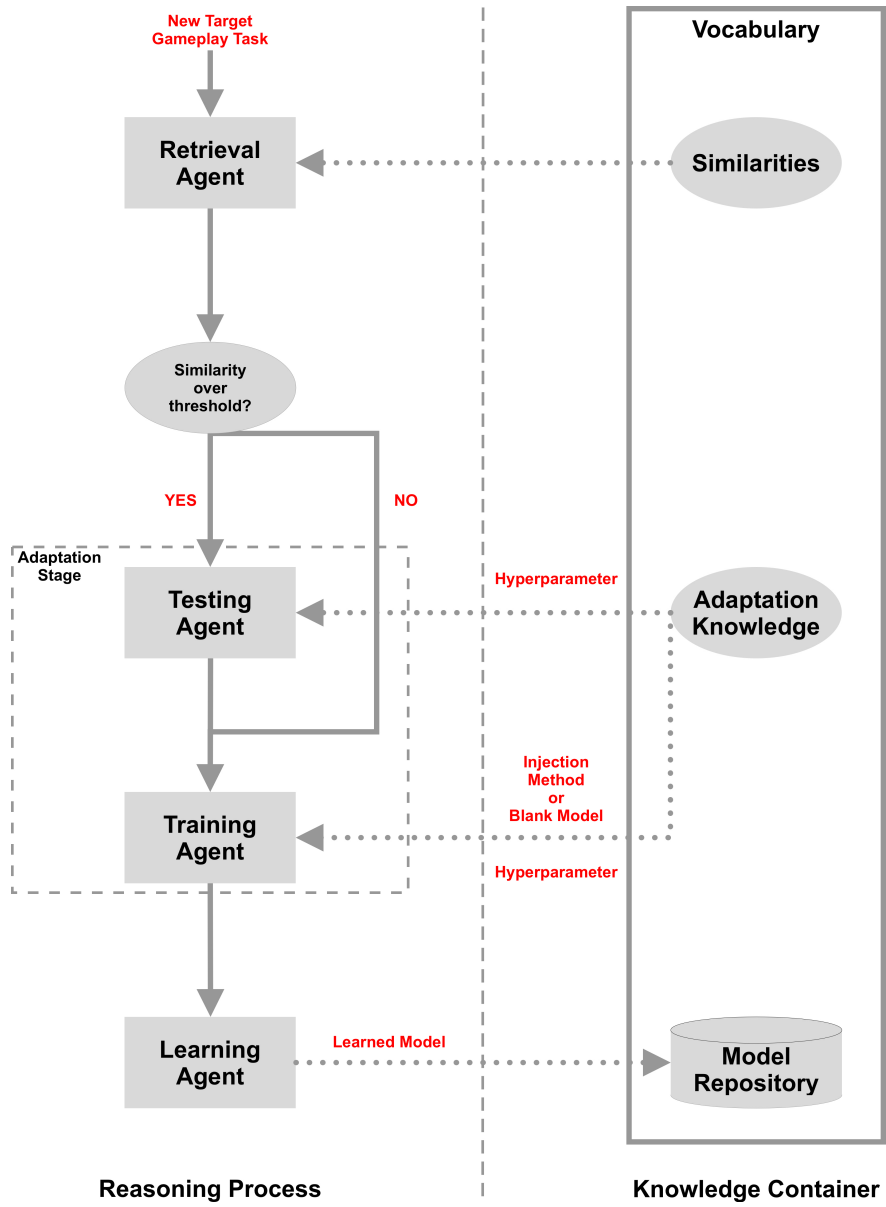


Fig. 2. PENG Process Model based on [2]

- A reliable similarity measure for integer values
- A reliable similarity measure for image data

We employ the Euclidean distance to measure the distance between the `nb_action` attribute. Since the proposed PENG system also works with image data, the retrieval step requires a similarity measure that handles pixel values and interprets the stored data. The image classifier applies two DNNs, which use different prediction methods. On the one hand, the prediction layer uses the softmax function. On the other hand, the last layer uses the sigmoid function. The model with the softmax activation function is stronger in classifying already known data, while the model with the sigmoid activation function has an advantage when showing the network environments that it has not seen before. The sigmoid function shows nonetheless the (estimated) probability that a new environment belongs to a trained class. Equation 5 presents the gameplay similarity, where *SIP* refers to ‘sigmoid prediction’ and *SOP* to ‘softmax prediction’.

$$Sim_{gameplay} = w_{sigmoid}SIP + w_{softmax}SOP \quad (5)$$

Equation 6 shows the amalgamation function for the global similarity. When calculating the global similarity, the attributes receive an individualized weight w to bias the retrieval.

$$Sim_{global} = w_{nb_action}Sim_{nb_action} + w_{gameplay}Sim_{gameplay} \quad (6)$$

3.3 Retrieval Agent

The retrieval agent is the entry point to the PENG system, whenever the system receives a new target gameplay task Λ . The main goal of this agent is to search within the model repository for saved game environments that are similar to the query case in order to train a more powerful DRL agent. Consequently, the system utilizes problems and solutions from the past. Ideally, this agent should output a significantly reduced section of the model repositories in the form of \mathcal{R}_{test} . The retrieval agent’s function is to select appropriate models in \mathcal{R}_{test} , that are as similar as possible to the current task Λ . Whenever the retrieval agent recovers a similar model from \mathcal{R} , the DRL agent has the opportunity to learn from past experience. Moreover, already trained and tested models are reused or recycled, which contributes to the challenges task limitation and the costly learning process. The main steps are:

1. Initialize the new environment,
2. Load the model repository and set the query case,
3. Calculate the similarity between the number of actions,
4. Calculate the similarity for gameplay images,
5. Calculate the global similarity.

3.4 Testing Agent

The testing agent receives the most similar case from the retrieval agent within the \mathcal{R} . The main goal of this agent is to apply the chosen model inside \mathcal{R}_{test}

with the corresponding architecture and policy to the query environment and to select the best performing model. In other words, the agent uses \mathcal{R}_{test} to identify a model that is already performing well for the target gameplay task \mathcal{A} . However, with the preselected models in \mathcal{R}_{test} , it is not guaranteed that the stored models μ are also usable for the current application \mathcal{A} . The agent's output is a suggested model for the current task \mathcal{A} , that the testing agent passes on. The main steps are:

1. Define the most similar model with architecture and policies,
2. Check if the last layer of the DRL agent has to be changed,
3. Build the DRL agent,
4. Test query environment with every policy from the most similar environment,
5. Select the best performing policy.

3.5 Training Agent

The central part of the training agent is to provide a reasonable solution for the query environment. This objective can be achieved by either train the agent with the Q-injection method from scratch, when no similar game environment was found. The second way is to train the new game with the model-injection procedure, which applies a similar solution from the model repository.

Q-Injection The Q-injection method introduces an additional probability ϕ , besides the parameter ϵ that corresponds to the exploration-exploitation strategy. ϕ determines if the agent injects Q-values during the training phase. From the beginning, the Q-values are not definite, therefore random numbers between $[0,1]$ are drawn as a first heuristic in order to inject values into the DRL agent. The range of numbers refers to the minimum and maximum values that are possible for the Q-values in this setting. This procedure takes advantage of the initial instantiated Q-values that are zero. Moreover, Q-values turn into values between $[0,1]$ after training. Based on this, we assume that the injection of Q-values > 0 has a positive influence on learning behavior.

Model-Injection In this case, the DRL agent uses the complete model of the previously trained agent, including architecture and policy, to solve \mathcal{A} . The proposed model serves as a blueprint for the new target gameplay task. Thereupon the training agent injects the experienced model, including the policy into the new agent. Once this transfer process is complete, the training phase starts. The main steps are:

1. Initialize the DRL agent,
2. Check for transfer mode (Q-injection vs. model-injection),
3. Check if the last layer of the DRL agent has to be changed,
4. Build the DRL agent,
5. Config and compile the DRL agent,
6. Train the DRL agent.

3.6 Learning Agent

The central task of the learning agent is to save the newly generated information and knowledge, that the PENG method can reuse for succeeding tasks.

The learning agent consists of two main components, first a learning component and second a maintenance part. CBR systems store and process experience knowledge, ordinarily with different approaches. The agent can build up knowledge or reject the currently learned model τ_A . Consequently, the agent decides whether she retains a model or not. The most obvious way is to compare the recently learned model with the models within the model repository. Only if the newly learned model is significantly different from the previous models within the model repository, the agents saves it. For the implementation stage of the learning agent, we used this comparison method, which learns models that are fundamentally different from the previous ones. The main steps are:

1. Clean the model repository and learn new models,
2. Collect image data of the game for NN,
3. Image augmentation,
4. Train the NN for image classification.

4 Experiment and results

The evaluation of the PENG system is essential to assess the theoretical elaboration and practical implementation more precisely. Therefore, this section describes various experiments that have been carried out in order to determine the performance of the system. The proposed framework consists of several parts, based on the CBR methodology. Nevertheless, the main focus is on the behavior achieved by the DRL agent. For this reason, the core of the experiments will rely on the results of the agent performance, supported by the PENG methodology.

4.1 Experiment Setup

The defined specifications prior to the first run:

- PENG starts with an empty model repository
- the retrieval agent starts after the system trained three games in advance
- similarity threshold of 0.4 as first-fit heuristic
- weights for the amalgamation function are set equal to 0.5 for nb_actions and 0.5 for gameplay
- parameters of the DRL agent correspond to the work of Minh et al. [6]
- a deep Q network (DQN) is used as the baseline

Table 1 shows the used hyperparameters within the DQN and the PENG experiments. The PENG system trained each game for 2,000,000 steps, and the environment was freely selected from the Atari2600 repository.

4.2 Metrics

This subsection implements some metrics that provide sufficient information about the success or failure of the proposed PENG method.

	DQN	PENG
Training Steps	2.000.000	2.000.000
Learning Rate	0.00025	0.00025
Gamma	0.99	0.99
Target Model Update	10.000	10.000
Warmup Steps	50.000	50.000
Epsilon Start	1.0	0.5
Epsilon End	0.1	0.1
Annealing Steps	1.000.000	1.000.000
Replay Limit	1.000.000	1.000.000
Reward	[-1.0, +1.0]	[-1.0, +1.0]

Table 1. Hyperparameters: DQN vs. PENG

Performance Increase/Decrease Algorithms that take less effort to achieve a defined goal are successful in practical implementation. Therefore, it is desirable to investigate how many steps were needed to reach the same reward level as the baseline algorithm after 2,000,000 steps and vice versa. Equation 7 and equation 8 formalizes this metric.

$$PI = 100 \cdot \frac{\text{Total Steps}_{DQN}}{\text{Steps}_{PENG | \text{Reward}_{PENG} = \text{Total Reward}_{DQN}}} \quad (7)$$

$$PD = 100 \cdot \frac{\text{Steps}_{DQN | \text{Reward}_{DQN} = \text{Total Reward}_{PENG}}}{\text{Total Steps}_{PENG}} \quad (8)$$

Episode Reward The achieved reward in the particular episode provides an insight into the learning behavior of the agent over the entire period. Equation 9 defines the Episode Reward.

$$\text{Episode Reward} = \sum_{\text{step}=0}^{\text{Episode End}} \text{Reward}_{\text{step}} \quad (9)$$

Sum Reward RL agents are conventionally measured by how beneficial a proposed policy is or how much reward an agent gets in its environment. One way to demonstrate the performance of an RL agent is to show the total of all rewards received over the entire period. Equation 10 defines the sum reward.

$$\text{Sum Reward} = \sum_{i=0}^{\text{Total Episodes}} \text{Reward}_{\text{Episode}_i} \quad (10)$$

4.3 Results

This section provides the test results of the experiments based on the implemented method. We present the experiment results of the model-injection procedure, and the results of the Q-injection method. The abbreviations from the data tables refer to the following Atari2600 games:

- B - Breakout
- MP - MsPacman
- S - Seaquest
- A - Alien

Model-Injection The results in Table 2 show that the presented method has achieved a definite increase in performance if the PENG system found a similar case within the model repository. The technique achieved the most significant performance increase of 121.54%. In this case, the target gameplay task was ‘Breakout’, and the retrieval agent recovered the most similar case ‘Breakout’ from the model repository.

Approach		B	MP	S	A
DQN	Steps	2000000	2000000	2000000	2000000
PENG	Steps	1645490	1924129	1941309	1821558
PI/PD		121.54%	103.94%	103.02%	109.80%

Table 2. Experiment Results for model-injection - PI/PD

Table 3 shows the measured values for the cumulative reward over the entire time. The average reward and maximum reward at the end of training are significantly higher in all gaming environments. The obtained results indicate that the search for a similar case leads to a not inconsiderable increase in the overall reward.

Approach		B	MP	S	A
DQN	SR	Mean 9161.17	54012.14	8011.16	33211.40
		Max 41640.00	132169.00	21640.00	81105.00
		Std 10959.72	39447.67	6413.29	23882.56
PENG	SR	Mean 17887.57	56945.64	9889.01	42653.81
		Max 51909.00	136346.00	22445.00	89691.00
		Std 15658.34	40482.38	6784.25	26854.50
100% $\frac{\text{PENG}}{\text{DQN}}$	SR	Mean 195.25%	105.43%	123.44%	128.43%
		Max 124.66%	103.16%	103.72%	110.57%
		Std 142.87%	102.62%	105.78%	112.44%

Table 3. Experiment Results for model-injection - Sum Reward(SR)

Q-Injection Table 4 describes whether the method is attributable to a performance increase. The data shows a comparison between the baseline method with the experiment of the Q-injection method. Since all experiments were stopped after 2,000,000 steps, the games Breakout, MsPacman, and Alien achieved fewer rewards in total within the PENG learning cycle. Based on the 2,000,000 steps, there is only a performance increase of 102.99% in the game Seaquest.

Table 5 shows the recorded data points of the accumulated rewards over the entire training period. The data shows that only the game Seaquest achieved

Approach		B	MP	S	A
DQN	Steps	1928880	1958882	2000000	1948749
PENG	Steps	2000000	2000000	1941861	2000000
PI/PD		96.44%	97.94%	102.99%	97.44%

Table 4. Experiment Results for Q-injection - PI/PD

more rewards throughout the 2,000,000 steps, increasing by 104.78%. The lowest value achieved the Breakout environment with 95.34%. Interestingly, the average for the sum of the rewards is higher in the game Alien, but the maximum result is below the baseline method.

Approach			B	MP	S	A
DQN	SR	Mean	9161.17	54012.14	8011.16	33211.40
		Max	41640.00	132169.00	21640.00	81105.00
		Std	10959.72	39446.67	6413.29	23882.56
PENG	SR	Mean	8583.87	51706.76	7430.91	35012.35
		Max	39701.00	128620.00	22675.00	79180.00
		Std	10568.42	36998.50	6351.68	23743.15
100% $\frac{\text{PENG}}{\text{DQN}}$	SR	Mean	93.70%	95.73%	92.76%	105.4%
		Max	95.34%	97.31%	104.78%	97.63%
		Std	96.43%	93.79%	99.04%	99.42%

Table 5. Experiment Results for Q-Injection - Sum Reward(SR)

4.4 Discussion

The results of this paper are two-fold concerning the DRL learning process. We showed that the model-injection method provides a distinct advantage when the model repository stores a similar case. On the opposite, the Q-injection method yields to an equal or even more miserable result, compared to the baseline method. The applied t-Test verified the findings for both methods with a high significance ($p=0.01$). However, our paper examined a limited representation of Atari2600 games that could affect the validity of the data. In future work, we advise conducting the experiments on a high-performance computer system, in order to obtain more results in parallel and scrutinize more games or other game environments. On a side note, the Q-injection method was instantiated with random values between $[0,1]$. Therefore, a better-constructed initialization technique for the corresponding Q-values may produce better results. Furthermore, we measured the achievement of the DRL agent to reason about the acceleration of the learning process. Further analyses should also focus on the required time for the complete PENG cycle since improvements regarding the learning period also correlate with the overall process. For the evaluation of the results, the algorithm operated for 2,000,000 steps. Future investigations should test a different stop criterion for assessing the procedure, such as the number of episodes or the reward to be achieved. Also, we implemented the

PENG framework prototypically with a first-fit heuristic. Consequently, we suggest that especially the testing agent and training agent evolves, in order to inject a more beneficial model with probably more robust results. Despite the mentioned obstacles, the outcomes indicate that PENG can support the DRL learning behavior within the Atari2600 gaming environment and can serve as a benchmark for further developments linked to our procedure.

5 Conclusion

We provided and evaluated the proposed PENG methodology and processed the measurement results from the experiments. The defined metrics were applied to the model-injection and Q-injection methods. The validation showed that the model-injection procedure has a clear advantage and we found indications for a lower performance of the Q-injection method. The statistical analysis shows that the model-injection methodology yields a significant improvement in training behavior, whereas the results from the Q-injection procedure can be put into perspective, since the result unveils no significant difference between DQN as baseline and PENG, except for the Breakout game. The obtained knowledge inside the KC can be further tested for other games, especially in terms of testing the process of adapting knowledge between different games. Furthermore, the parameters for both, the DRL agent and any agent involved in the PENG methodology can be further investigated.

References

1. Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Commun.* **7**(1), 39–59 (1994)
2. Amaief, K., Lu, J.: Ontology-supported case-based reasoning approach for intelligent m-government emergency response services. *Decision Support Systems* **55**(1), pp. 79–97 (2013)
3. Bianchi, R.A.C., et al.: Heuristically accelerated reinforcement learning by means of case-based reasoning and transfer learning. *Journal of Intelligent & Robotic Systems* **91**(2), pp. 301–312 (2018)
4. Hillmann, J.: Conception and development of a prototype for a multi-agent-system with learning capabilities using case-based reasoning in the first-person perspective scenario. (2017), master thesis at University of Hildesheim
5. Marcel Kolbe, Pascal Reuss, J.M.S., Althoff, K.D.: Conceptualization and implementation of a reinforcement learning approach using a case-based reasoning agent in a FPS scenario. *CEUR Workshop Proceedings* **2454** (2019)
6. Mnih, V., et al.: Human-level control through deep reinforcement learning. *Nature* **518**(7540), pp. 529–533 (2015)
7. Newman, T.: Could artificial intelligence be the future of cancer diagnosis? (2019), <https://www.medicalnewstoday.com/articles/325750.php>, last validation: 02/17/2020
8. Saleem, F.: The future of self-driving cars: New generation of transportation (2018), <https://innov8tiv.com/the-future-of-self-driving-cars-new-generation-of-transportation/>, last validation: 02/17/2020

Towards case-based reasoning in real-time strategy environments with SEASALT

Jakob M. Schoenborn^{1,2} and Klaus-Dieter Althoff^{1,2}

¹ University of Hildesheim
Universitätsplatz 1, 31141 Hildesheim, Germany
schoenb@uni-hildesheim.de

² German Research Center for Artificial Intelligence (DFKI)
Trippstadter Str. 122, 67663 Kaiserslautern, Germany
kalthoff@dfki.uni-kl.de

Abstract. Real-time situations provide numerous different problems to solve. Starting with the requirement of finding a solution inside an acceptable time frame, the problem is to find the right balance between performance and precision of the system. One might not be able to wait multiple minutes for a solution, however, a too quickly made decision might entail a certain risk factor. Thus, it has to be decided when methods such as using a rule-based system is sufficient and when it is rather beneficial to take the cost of using methodologies for knowledge management. We are using StarCraft II as an example for decision making in a real-time environment using incomplete information with a finite set of buildings and units to control. We propose using agents to decide the proportion of command authority between using a rule-based and a case-based reasoning agent. Earlier stages of the game seem to be promising for immediate reactions while later stages of the game require more planning due to the increased rate of information, which have to be processed. By reusing past experiences, case-based reasoning may be able to help improving the planning process.

Keywords: Case-based Reasoning · Realtime Strategy · Knowledge Management

1 Introduction

To solve problems in a real-time situation is very difficult, especially with incomplete information. The general problem consists in finding a selection of complex processes in order to decrease the idle time of any given unit as much as possible. This is not only applicable in the gaming area, but similar problems can be found in the production area: using limited resources to obtain the highest possible output. The longer it takes to process the information, the more likely it is to lose value of the made decision, due to the delayed provision of the solution

Copyright © 2020 by the paper's authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).



Fig. 1. Screenshot of StarCraft II during early stages of the game: Building the base, train combat units and manage resources. The panel in the bottom contains multiple information based in the selected unit such as health points, attack damage, unit-specific abilities, and a minimap with three different levels of fog of war (visible, visited but not currently visible (gray), and never visited (black)).

or since the data as a decision foundation has been changed during the process - or a combination of both. This requires the decision making processes to be as effective as possible.

StarCraft II is a real-time strategy game developed and maintained by Activision Blizzard, including hosting and organizing tournaments which since 2010 have awarded \$33,003,549.29 in total prize money from 5839 tournaments, with a majority of the price money granted to players located in south korea [3]. The game, representative for any other games in its genre, evolves around two different aspects and can be seen in Fig. 1: macro- and micromanagement. The former considers the usage of resources (minerals, blue crystals) to build structures and combat units, the latter considers moving units for scouting and fighting the enemy player. Combat contains multiple different aspects to consider, such as differences in unit weapon- and armor types, ground and air units, and additionally in the kind of movement itself, for example, using a “hit- and run”-strategy to deal as much damage as possible while taking as less damage as possible. With these differences, it is important to scout the enemy to know about the enemies chosen strategy to counter it by building corresponding units. These receive a damage bonus based on their weapon type against the armory types of the enemies unit. This has also been tested in a CBR approach by Cadena and Garrido [2].

The usage of reinforcement learning (RL) and artificial neural networks (ANN) is in current research the way to solve seemingly any problem, for example, Vinyals et al. provided 2019 an AI, which is capable of defeating professional StarCraft II players [9] or earlier approaches by Mnih et al. using a similar approach [6]. However, ANN usually implies the unfair advantage of numerous training sessions, i. e., parallelizing games, or taking larger datasets into account than a human could process (971.000 replays of games played by human players have been used as data set) [9]. Since one of the strengths of case-based reasoning (CBR) is to use it even with a smaller set of cases, we investigate the possibilities of implementing CBR into the decision making process.

To consider the possibilities, we overview recent work which has been done in the real-time strategy games area, followed by describing our ideas on micro- and macromanagement agents, and ending with a conclusion including future work.

2 Related Work

Vinyals et al. developed AlphaStar, an AI which combines ANN and RL especially during training [9]. The API on which the AI has been created contains an observation object, which provides necessary information such as visible units, buildings, and the environment in general. Using these observations and the abilities of the owned units, the action state space can be transmitted via the monitoring layer. The monitoring layer processes the received observations by an artificial delay of 80ms and limiting the number of taken actions to approx. 22 per 5 seconds. This decision has been made to prevent the AI to gain an unfair advantage in contrast to the human player who is limited in the number of physical actions per second. The professional player Dario ‘TLO’ Wunsch credits AlphaStar to not be “superhuman”, resulting in an overall fair feeling when playing against the AI [9]. In terms of learning components, the combination of RL and supervised learning (SL), multiple instances of RL agents are spawned by the SL layer, collect experiences, update the policy and value outputs. As a baseline, replays of human games have been used to learn from their behaviours and strategies and apply it to the current player.

Wender and Watson presented 2014 an approach to combine CBR with RL specifically for the micromanagement problem [10, 11]. Two kinds of agents are presented. One agent is observing the overall state of the game by creating so-called influence maps. These are areas, in which the system can take influence by executing actions such as attacking, building, or scouting to further increase the influence. Any other agent represents one unit object of the game, such as a marine soldier selected in Fig. 1. The casebase contains cases, which describe, for example, actions based on the current influence, such as sending certain agents to specific areas to increase the influence. These cases are not changed during the execution of the program. However, using Q-learning in the RL component, the solution of the most similar case may be adjusted to fit into the current situation.

Based on the perceived result after execution, the agent will be rewarded or punished [10, 11].

3 On the granularity level of control

The goal is to defeat the enemy by destroying every building and unit. Depending on the experience of the opponent, especially against beginners, a rule-based agent using only a set of few rules, such as building as soon as possible and collectively attacking after x units have been trained, is sufficient to fulfill this task. There are a few rules of thumb, which generally hold true and can easily be followed by any rule-based agent, such as not being supply blocked (meaning certain buildings have to be built before recruiting more units), using excess resources to build further production buildings for faster recruiting, and gathering combat units before heading towards the enemies base. This does not take the complexity of the game into account, which has been slightly mentioned in the introduction.

For the agent framework, we use the SEASALT of Kerstin Bach, which consists of multiple different layers dealing with knowledge presentation, -provision, -representation, -formalization and knowledge sources [1]. Fig. 2 shows the complete architecture layout. As a brief overview, the “*Shared Experience using a Agent based System Architecture Layout*” uses one coordination agent who controls n topic agents. Each of those agents, including the coordination agent, contain a case factory, which in turn contains multiple agents for knowledge maintenance and formalization. A collector agent gathers information from a community of experts, for example, by using crawling technologies and extracting textual information into knowledge representations. Other kinds of knowledge representations are ontologies, taxonomies, similarity measures, constraints, vocabularies, and rules. These can be accessed by any layer, while knowledge formalization and knowledge sources may also modify these.

For our application, we propose to instantiate a coordination agent, macro- and micro agent and an explanation agent. The coordination agent inhabits templates and a question handler in a knowledge map. One coordination agent can control other agents, which contain an own case factory and casebase. The casebase of these agents (including the coordination agent) are specially targeted for their individual needs, especially in terms of defined similarity measurements for the retrieval. For example, the micro agent might value the unit count for tactical combat decision making higher than the macro agents for prioritizing the reproduction of fallen units. To account for the complexity of the game and to inhabit a learning component, the following models are possible:

Centralized CBR. By only using one methodology, there is no interference and communication overhead between multiple agents to be expected, resulting in an overall faster decision making process. However, it seems questionable whether it is feasible to let each agent query the CBR system - and as such the casebase - based on numerous triggering events. This may create the necessity to repeatedly

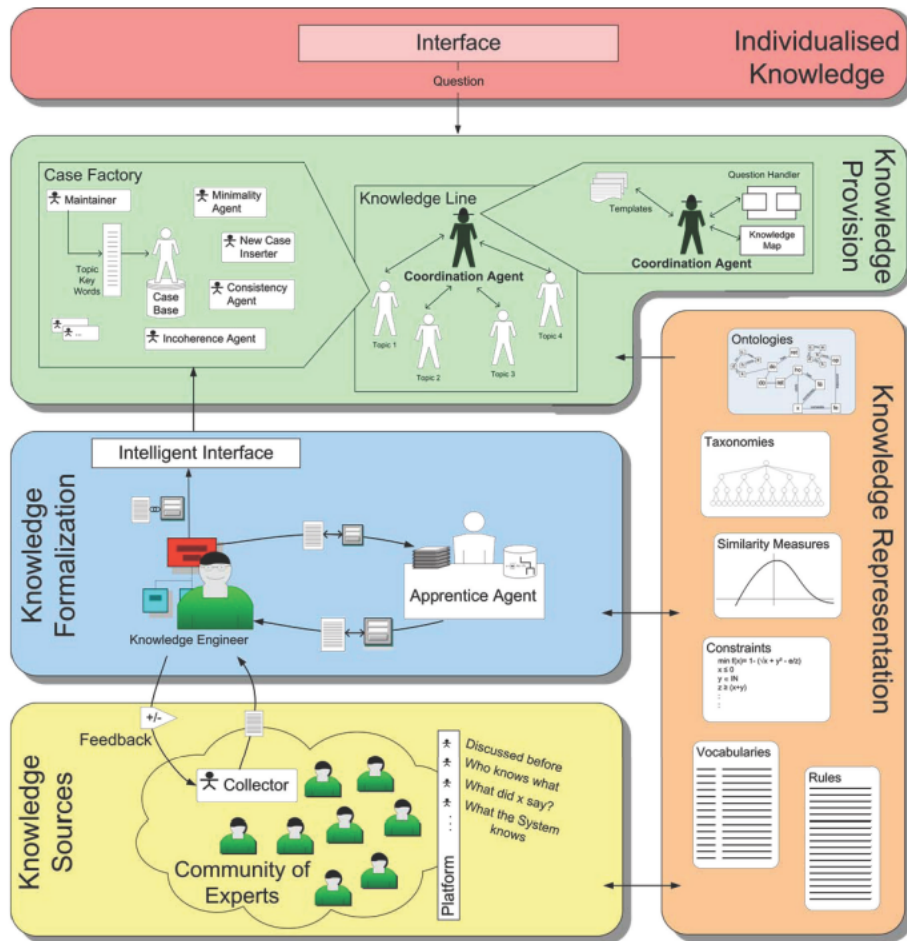


Fig. 2. SEASALT, a domain independent architecture for knowledge management using multiple agents

evaluate the current plan, especially in very information heavy situations such as during fights between multiple armies.

Distributed problem solving. For distributed problem solving, a coordinator agent can be used as a first-level support to handle increasing complexity limits, such as during combat fights. The coordinator agent functions as the centralized CBR system, which has been partially covered by Wender and Watson [10]. By combining the information gathered by the micro- and macro agent, the coordinator agent combines these with the general information of the observation object (such as resources and visible areas) to create a new plan. A plan may consist out of multiple sequences, analogous to the approach of Kolbe et al. used in first-person shooter gaming scenarios [5].

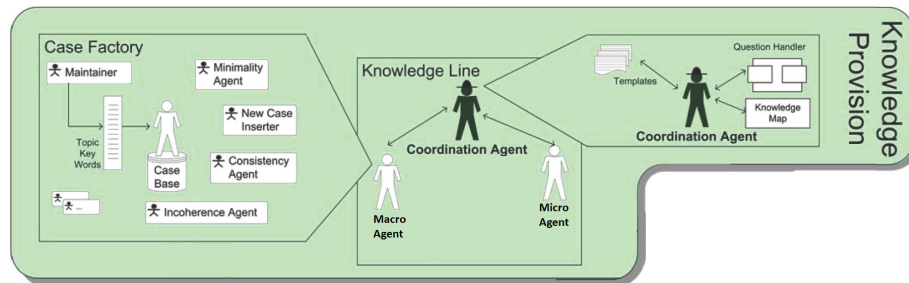


Fig. 3. Instantiation of distributed problem solving using one macro agent, one micro agent, and one coordination agent.

CBR and ANN in combination. As discussed above, Vinyals et al. used ANN in combination with RL to train the agents with new strategies and planning capabilities [9]. For providing explanations for an ANN system, Keane and Kenny defined ANN-CBR twins. The combination of ANN and a CBR technique, which mostly used k-NN in their case, provided better results than considering each method separately [4]. The approach could be used analogously: CBR could provide most similar cases to the current situations while ANN interprets these and takes control over the game state and the overall game plan.

In Fig. 4, an explanation agent has been added to the knowledge provision layer. This agent will receive information of the macro- and micro agent directly whenever the agent queries those, in addition to the coordination agent. The coordination agent in turn may provide the solution to the graphical user interface. The explanation is targeted for the knowledge engineer to further understand why certain actions has been taken. This is helpful to understand the learning process of the micro- and macroagents and support further debugging of those, allowing a more targeted way for knowledge maintenance to increase the learning rate of the agents and to gain a more purposeful search through the casebases. Since it can be assumed that the knowledge engineer inhabits knowledge over the domain, providing a list with most similar cases and their features may serve as

an explanation by itself - using the inherent explainability of CBR. Otherwise, for rather novice users, explanation patterns as introduced by R. Schank can be used for explanations [8]. These patterns can be filled, for example, with the rules that have been used, the applied similarity measures, the used vocabulary, or adaptation rules that have been triggered. These are the components of the knowledge containers as defined by Richter [7].

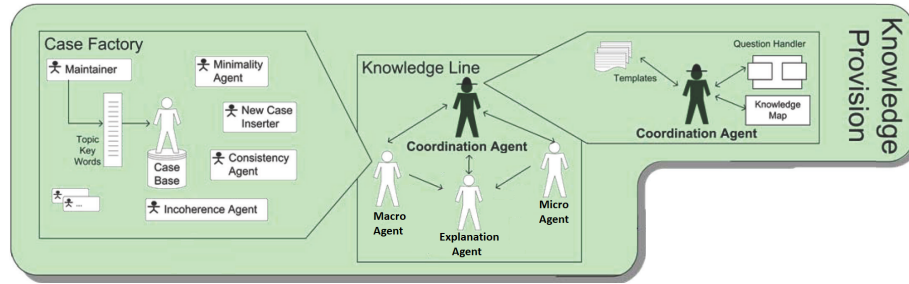


Fig. 4. Instantiation of distributed problem solving extended by adding an explanation agent, which gathers information from the macro- and micro agent and communicates with the coordination agent, transferring explanations to the graphical user interface.

Full multi-agent system. Each unit is treated as an own agent. However, these agents need to willingly work together as a team to defeat the opponent. Typically, in a multi-agent system, each agent may interact with any other agent as well as the coordinator agent who coordinates single agents to fulfill the overall goal (which may be split into separate lower level goals). The architecture is analogous to SEASALT in Fig. 2. From a knowledge management perspective, it may be interesting to evaluate on which granularity the level of control should be settled. As stated before, each unit could be treated as an own agent. A typical army usually consists out of 20-40 units, thus, leading to 20-40 agents in need to communicate and coordinate with each other and the coordination agent. Another possibility may be to designate one CBR agent to a specific unit type, since usually an army only consists out of 2-4 different unit types. This would decrease the level of communication needed drastically. However, experiences are then also limited to a single agent.

4 Conclusion

Developing an artificial intelligence in real-time strategy fields provide multiple challenges. However, there are also chances for increasing the efficiency by including CBR. Here, we shared a few thoughts on the granularity level of control by different approaches. With having positive and negative aspects, the question remains open whether there is an overall “best” approach or whether a hybrid architecture might be feasible as well. Regarding this question, another important question is the granularity of the case structure in terms of choosing

attributes and how to model the local and global similarity without ending up in overfitting problems. Based on the case structure, the case instances themselves are another challenge on their own. For a given sequence of events, a starting point and an end point has to be defined. Especially in terms of learning from single fights, the start and the end of a fight has to be determined and saved inside the case structure. Furthermore, the transferability of knowledge learned by a single agent to another agent using the same (or parts of) casebase remains to be an open research question, which may help to structure the casebases of a single agent. These challenges may be considered in future work.

References

1. Bach, K.: Knowledge Acquisition for Case-Based Reasoning Systems. Ph.D. thesis, University of Hildesheim (2013), <http://www.dr.hut-verlag.de/978-3-8439-1357-7.html>
2. Cadena, P., Garrido, L.: Fuzzy case-based reasoning for managing strategic and tactical reasoning in starcraft. In: Batyrshin, I., Sidorov, G. (eds.) *Advances in Artificial Intelligence*. p. 113–124. Springer Berlin Heidelberg (2011)
3. Earnings, E.: *StarCraft II Top Players & Prize Pools - Esports Tracker :: Esports Earnings* (2020), <https://www.esportsearnings.com/games/151-starcraft-ii>, last validation: 06/14/2020
4. Keane, M.T., Kenny, E.M.: How case-based reasoning explains neural networks: A theoretical analysis of XAI using post-hoc explanation-by-example from a survey of ANN-CBR twin-systems. In: Bach, K., Marling, C. (eds.) *Case-Based Reasoning Research and Development - 27th International Conference, ICCBR 2019, Otzenhausen, Germany, September 8-12, 2019, Proceedings*. Lecture Notes in Computer Science, vol. 11680, pp. 155–171. Springer (2019)
5. Kolbe, M., Reuss, P., Schoenborn, J.M., Althoff, K.D.: Conceptualization and implementation of a reinforcement learning approach using a case-based reasoning agent in a FPS scenario. In: *LWDA 2019, Workshop on Knowledge Management, Berlin* (2019)
6. Mnih, V., et al.: Human-level control through deep reinforcement learning. *Nature* **518**(7540), 529–533 (2015). <https://doi.org/10.1038/nature14236>
7. Richter, M.M.: Fallbasiertes Schließen. Görz, Günther; Rollinger, Claus-Rainer; Schneeberger, Josef (Hrsg.): *Handbuch der Künstlichen Intelligenz* **4**, 407–430 (2003)
8. Schank, R.C.: *Explanation Patterns: Understanding Mechanical and Creatively*. L. Erlbaum Associates Inc., USA (1986)
9. Vinyals, O., et al.: Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* **575**(7782), 350–354 (2019). <https://doi.org/10.1038/s41586-019-1724-z>
10. Wender, S., Watson, I.: Combining case-based reasoning and reinforcement learning for unit navigation in real-time strategy game AI. In: Lamontagne, L., Plaza, E. (eds.) *Case-Based Reasoning Research and Development*. p. 511–525. Springer International Publishing (2014)
11. Wender, S., Watson, I.: Integrating case-based reasoning with reinforcement learning for real-time strategy game micromanagement. In: Pham, D.N., Park, S.B. (eds.) *PRICAI 2014: Trends in Artificial Intelligence*. p. 64–76. Springer International Publishing (2014)

Process Mining for Case Acquisition in Oncology: A Systematic Literature Review

Joscha Grüger¹ , Ralph Bergmann^{1,2} , Yavuz Kazik¹, and Martin Kuhn¹

¹ Business Information Systems II, University of Trier, 54286 Trier, Germany
<http://www.wi2.uni-trier.de>

{grueger, bergmann, s4yakzi, s4makuhn}@uni-trier.de

² German Research Center for Artificial Intelligence (DFKI), Branch University of
Trier, Behringstraße 21, 54296 Trier, Germany
ralph.bergmann@dfki.de

Abstract Process Mining is a technology family for the analysis of business processes based on event logs. The methods are successfully applied in various areas, including medicine. This paper examines, using a systematic literature review, whether Process Mining is suitable for case acquisition from Hospital Information Systems in order to construct a case base for experience-based systems targeted at decision support in oncology. The review investigates whether there are special characteristics of process mining in the oncological field compared to other medical fields and if the development of similarity measures is discussed in the contributions. For this purpose, 2848 papers were reviewed manually, based on title, abstract and full text, resulting in 55 relevant papers. These were analyzed in detail regarding the research questions. The paper can serve as a basis for further research, identify research opportunities in this domain and provide a useful overview of the current work.

Keywords: Process Mining · Oncology · Case Based Reasoning · literature review.

1 Introduction

Medical guidelines are “systematically developed statements designed to assist healthcare professionals and patients in making decisions about appropriate health care in specific clinical circumstances” [28]. These are classified according to the AWMF³ system into four development levels from S1 to S3, with S3 being the highest quality level of the development methodology. The classification of a guideline as S3 means that it has undergone all elements of systematic development and the recommendations given therein have a high level of evidence [11]. In the best case, clinicians can make treatment decisions based on these high-quality S3 guidelines and are thus able to offer evidence-based treatment.

Copyright © 2020 by the paper’s authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

³ Arbeitsgemeinschaft der Wissenschaftlichen Medizinischen Fachgesellschaften

This is usually possible with well understood disease patterns, such as stroke. In other areas, such as oncology or paediatrics, there is in many cases insufficient evidence for a fully evidence-based treatment of patients. This is partly because studies in these areas are difficult (e.g. with children), diseases are rare or disease patterns are not yet sufficiently researched due to their complexity (e.g. uveal melanoma). In addition, the process of developing guidelines is quite slow, i.e., it usually takes at least two years. In view of scientific progress, especially in medicine, the question of the timeliness of guidelines arises.

In the absence of appropriate guidelines and high evidence studies, treatment decisions are made based on personal experience of medical experts. In contrast to evidence-based medicine, we then speak of “eminence-based medicine”, as a treatment decision is based on the comprehensive professional experience of recognized medical experts in the field [19]. In the field of oncology, for example, multidisciplinary experts regularly meet in tumor boards to discuss critical cases and then make decisions, often based on treatment experience with similar patients.

Today, the complexity of such decisions is constantly increasing. The decision-making process is becoming more and more complicated due to the constant development of new therapeutic approaches, an ever-wider range of drugs and their frequently unexplored interaction with given constraints such as comorbidities. In addition, the departure of experienced physicians can have a negative impact on the quality of treatment, as their experience also leaves the clinic.

During the daily treatment of patients, however, physicians systematically record experiential knowledge in hospital information systems (HIS). A HIS is the central information system of a hospital and receives, transmits, processes, stores, and presents information. Date and time of treatments, patient demographics, and examination results are stored in a HIS along with other information [16]. We envision that this information can be used as experience by a Case-Based Reasoning (CBR) system to support eminence-based decision making by the wealth of collected experience available in HIS. For this purpose, treatment processes from a HIS must be captured as a time series of semantically described activities and transferred into semantic case descriptions in order to construct a case base.

In this paper, we therefore investigate based on a literature survey whether process mining, which is an established technology for extracting process knowledge from events logs, can be applied or has been applied already in order to acquire semantic case descriptions from HIS. So far there are only a few literature reviews in the field of process mining in medicine [35,41,13] and only one systematic literature review in the field of process mining in oncology [22]. None of the papers examines the use of process mining for case acquisition for CBR. Processes in the health care sector differ greatly from processes from other domains due to their high complexity, heterogeneity and significant variation over time [17]. This makes it difficult to adapt approaches from other domains. In the present work, a literature study in the medical domain of oncology is performed and used to investigate whether it is possible to generate systematic case descrip-

tions from HIS data using process mining. The paper focuses particularly on the data source from which data is acquired, the process mining methods used, and the data formats and descriptions used, with the aim to provide systematic basis for the topic. By analysing the literature on process mining in oncology, this paper also provides a foundation for future work and helps identifying challenges and research gaps based on the previous research.

The remainder of this paper is organized as follows: in Section 2 we give an overview of the basics of Process Mining and Case Based Reasoning and discuss related work. In Section 3 we present the methodology of the literature review. Then we evaluate the results of the study in Section 4 and summarize them in Section 5 and discuss possible directions for future work.

2 Foundations and Related Work

Case-Based Reasoning [21,3] is an established problem-solving methodology for solving problems based on past experience. Experience is formalized in the form of cases collected in a cases base. A problem (e.g. to determine the best treatment option of a patient) is solved by searching for similar cases in the case base and then reusing the solution contained in the most similar case(s). Unlike black box algorithms such as *deep learning*, the solutions of CBR systems can be easily justified on the basis of similar cases, which can help to strengthen the confidence of healthcare professionals in the AI system, especially in the medical field [26]. The CBR cycle consists of four sequential phases. In the RETRIEVE phase, the most similar cases for a given case are searched for in the case base. Then, in the REUSE phase, the information and knowledge about the most similar cases is used to solve the problem given. Afterwards the solution found in the REVISE phase has to be checked. In the RETAIN phase, those parts of the solution are included in the case base that could be useful for solving later cases [1]. CBR publications in the medical field usually focus exclusively on retrieve and avoid automatic adaptation [8].

Process mining technologies enable the extraction of process knowledge from event logs of information systems. Based on these techniques, process models can be created (discover) and improved (enhancement) and traces can be validated for their conformity with existing models (conformance checking) [37]. Process Mining is already partially used in medicine. The research focuses in particular on the field of oncology and operations. In other areas, such as care giving, cardiology, diabetes, dentistry, medication, intensive care, and radiotherapy, there are considerably fewer publications [13,35]. The focus of most process mining publications in the medical domain is usually on the control flow perspective, based on the discovery of the execution sequence of process activities [35].

3 Methodology

To answer the following research questions, a systematic literature review in the field of process mining in oncology was conducted:

RQ1: What is the state of research in the field of process mining in the domain of oncology?

RQ2: Are there process mining approaches based on oncological data from a HIS?

RQ3: Are there approaches to use process mining for case acquisition for experience-based systems?

RQ4: Are there studies that deal with the similarity of oncological processes?

The search is divided into three main parts: the initial search, the backward snowballing and the forward snowballing [42]. The results of each step are filtered through a three-step application of including- and excluding criteria's (see Fig. 1). Overall, one including, and three excluding criteria were established and

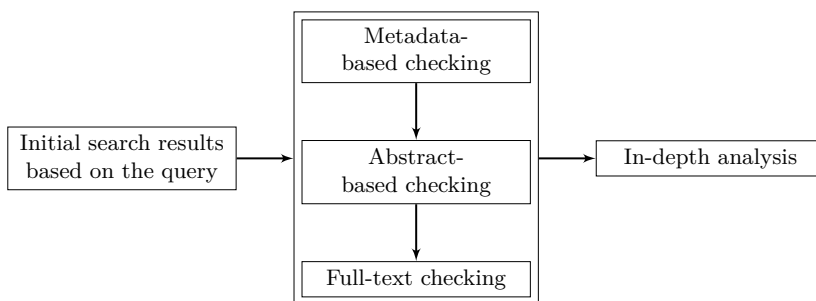


Fig. 1. Applying the including and excluding criteria's.

applied. These ensure that only relevant and accessible documents are included in the analysis:

EC1: Duplicates of the same study are excluded.

EC2: Articles that are not written in English or German are excluded.

EC3: Articles that are not published in a journal or at a conference are excluded.

IC1: Articles written in the field of process mining in oncology or whose authors use oncological data are included.

The first step of the initial search is the database selection. For this purpose, published literature searches in the field of process mining in medicine [35,22,25] were analyzed and the databases used therein were extracted as a basis for database selection. The following sources were identified: ACM DL, CiteSeerX, dblp, Google Scholar, IEEE Explore, PubMed, Science Direct, Scopus, Semantic Scholar, Springer and Web of Science. Based on the databases and a database selection matrix according to Bethel [4,27] the databases Google Scholar and Science Direct were selected.

The search query was created based on the PICOC method (Population, Intervention, Comparison, Outcome, Context) according to Kitchenham [20]. This

approach is intended to ensure that the query is precise and only considers the essential components. To ensure that the approach fits the given research question, the Data field has been added and the Comparison and Outcome fields have been removed. The final query is: (“oncology”) AND (“process mining”) AND (“hospital”) AND (“event log”). The same query was used for both databases.

The initial search took place on 20.12.2019. Google Scholar delivered 174 results and Science Direct 24. After forward and backward snowballing, 60 papers were classified as relevant. After analyzing the papers, five papers were excluded due to a lack of information concerning our research questions. Therefore, 55 papers were considered in the analysis process (see Fig. 2).

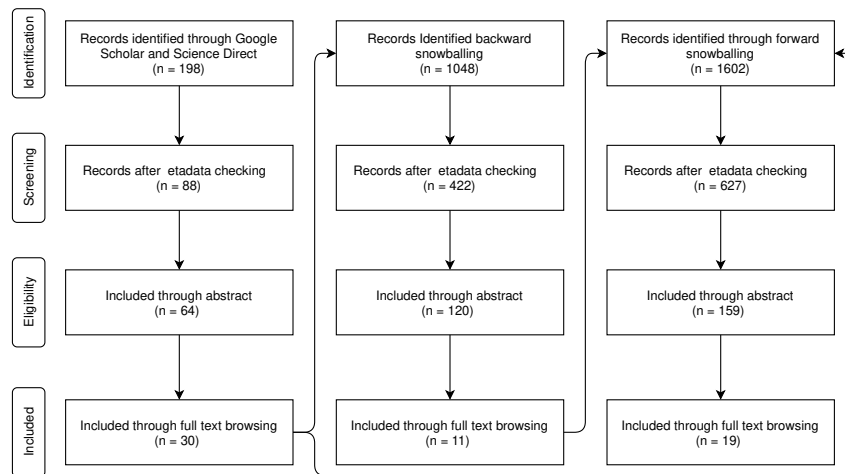


Fig. 2. PRISMA Overview of the results of the different steps of the literature search of the literature review.

To answer the research questions, a data extraction form was developed based on the core features of process mining in oncology and on metadata of the papers.

4 Results

The first papers on process mining in oncology were published in 2008. However, the majority of the papers, 48 out of 55, were published between 2013 and 2019. Most of the papers come from Europe (40 out of 55 papers). With 24 papers the Netherlands is the most important contributor in Europe. This is probably due to the large research group in the field of Process Mining at the University of Eindhoven (TU/e), which was headed by Prof. van der Aalst. From North America and Asia six papers each were found, from South America only two were found.

The papers analyzed address a total of 21 different types of cancer. The majority of the papers referred to gynaecological cancer (19 papers). Other cancers addressed are lung cancer and breast cancer (10 papers each), followed by colorectal cancer (9 papers found), skin cancer (5 papers) and stomach cancer (4 papers). 13 types of cancer were mentioned only once, and in eight contributions the type of tumour was not mentioned.

4.1 Data and Process Mining Perspectives

In order to answer research question RQ2, it was examined on which data the papers work and which data sources were used. After examining the process mining data spectrum, the data used mainly comes from administrative systems (58 %) and from the clinical part of hospital information systems (30 %). Only one paper uses data from medical devices. Most papers, 49 of 55, apply process mining technologies to medical data (diagnosis, prognosis, treatment and prevention of disease activities) and 3 papers use organizational data (management and financial), 3 papers use both medical and organizational data.

Data coming from HIS is described to be very complex, containing heterogeneous structured and unstructured data [10] and sometimes scattered across multiple HIS [5]. Poor data quality and the distribution of data across different HIS can significantly hinder the process extraction [5]. Mans et. al. [36] evaluate data quality issues in the data of a HIS. Among other things, they point out that manual documentation of events leads to the fact that individual events are not documented ("missing events"). In addition, the distribution of the data to different systems leads to imprecise timestamps and executing actors are imprecisely documented (imprecise resource).

Many authors emphasize the complexity of clinical processes (30 papers). They attribute this, among other things, to the high degree of flexibility, the dynamics in treatment processes and in everyday clinical life and a high number of interactions of interdisciplinary actors in a treatment path.

Regarding the process mining perspectives, it can be said that most papers focus on the control flow perspective (48 %). With 23 % follows the time perspective, which was mostly used to identify bottlenecks. The case perspective was only used in 17 % of the papers and the organizational perspective in 11 % of the papers.

The most used process mining technique is process discovery (found in 48 papers). One reason for this is that the other three process mining techniques require a process model, which is often generated via process discovery. Conformance checking was applied in 13 papers and process re-engineering in 6 papers. Operational Support was only used in three papers.

4.2 Process Mining Methodology

The methodology used in the papers clusters the papers according to the tasks to be performed when applying algorithms and techniques for process evaluation. Following [35], the present paper distinguishes between three methodological

approaches. The non-domain-specific ad hoc method is used in 21 papers. The clustering method, consisting of the five phases log preparation; log inspection; control flow analysis; performance analysis; and role analysis [6], is used in two papers. The L* life cycle[37], as the third methodological approach, also consists of 5 phases: Planning and justification; extraction; generating the control flow model and linking the event log; generating the integrated process model; and providing operational support [37]. This method was used in 4 papers. Most papers (29 contributions) do not describe a concrete procedure based on known methods.

4.3 Techniques, Algorithms, Tools and Software

In 30 papers special process mining algorithms are used, 36 % of the papers use data mining and machine learning algorithms and 9 % use algorithms from other areas. The most used algorithms are the process discovery algorithms [40] (10 papers), followed by the fuzzy miner [15] (6 papers).

Nearly half of the papers examined use the ProM⁴ software (42 %), 7 papers use the R programming language and the Process Mining Toolkit Disco [14] is used in 6 papers. Eight papers have not mentioned any software. ProM is probably the most used tool as it comes with many plugins, offers an interface to develop own plugins and the ProM core is open source⁵ [9].

4.4 Clinical Path Similarity

To answer research question RQ4, it was examined which papers cover the similarity of paths. Eight papers deal with the similarity of mined clinical pathways. The main challenge in the application of process mining techniques to medical processes and the subsequent comparison of clinical paths is, in the eyes of 5 out of 8 authors, the flexibility with which the activities are performed. Therefore, many clinical events occur randomly and often without a specified order. Thus, many common similarity measures for processes cannot be applied. Furthermore, it is stated that clinical processes are always time-linked. Therefore, they can change significantly over time and as research progresses [18].

To be able to compare these flexible and heterogeneous clinical pathways, the authors developed and used clustering approaches. The authors used these approaches to cluster activities and then calculated the similarity of the pathways based on the identified clusters of a pathway instead of the specific pathway with treatment activities. Only one approach defines a multidimensional similarity measure and includes besides the pure procedural data also performing actors/resources, and data values to calculate the similarity.

⁴ promtools.org

⁵ ProM 6 core, GNU Public License

4.5 Process Representation

None of the papers examines explicitly the use of process mining for case acquisition for CBR. Most papers use a procedural process modeling language like Petri Nets [32] (9 papers), BPMN⁶ (2 papers) and PWF⁷ [12] (2 papers). However, in most cases the exact representation is not given and the procedural character of the process modeling language can only be inferred from the algorithms used. Another representation was chosen by 7 authors, by using a declarative approach. All seven papers chose the declarative process modeling language [38], based on Linear Temporal Logic (LTL). The frequent use of Declare is due to its integration into ProM. The authors usually justify this approach by the suitability of declarative approaches for very flexible processes.

4.6 Research Gaps

To answer research question RQ3, research gaps were identified based on the papers analyzed. For this purpose, the three-step procedure proposed by Müller-Bloch et. al. [31] for identifying research gaps and the PICOS framework [34] was used. This process consists of the localization and characterization of the gaps in step one, the verification of the gaps in step two and the presentation of these in step three. The following research gaps were identified.

No papers were found in the area of case acquisition using process mining for knowledge-based systems (including CBR) in oncology. Studies on the transferability of process mining-based approaches to case acquisition from other domains to oncology are still missing.

One of the papers explicitly examines data quality issues in the process mining context in data from a Dutch hospital. There is no equivalent study for German oncology clinics. The complexity of the data from HIS is mentioned in the papers, but not examined in detail. However, this is interesting for the more advanced and especially for the multi-perspective process mining approaches. Therefore, further studies could provide a basis for further research in this area.

The cancer best researched with process mining technologies is gynecological cancer due to the BPI Challenge data set. Other data sets, such as the MIMIC III data set or the data sets used in [30,23] are not suitable for performance analysis due to data problems [24]. This indicates the urgent need for other available data sources in this domain.

The next gap describes the need of a data quality indicator [2,5,39]. There should be a method to measure the data quality of event logs. This is necessary for unsupervised learning techniques like Sched-Miner which rely on data quality due to the use of unsupervised learning [2]. The three noise types mentioned in [39] are a good starting point for further research concerning the quality indicators.

Research gaps were also identified in process reengineering. Declarative Process Mining deals well with highly variable processes which are the standard for

⁶ Business Process Model Notation, <https://www.omg.org/spec/BPMN>

⁷ Pseudo-WorkFlow Language

healthcare processes. In particular, there is a need for research in the preparation of a correct declarative constraint set based on guidelines and an adapted real log to be replayed [33,29].

5 Conclusion and Future Work

The analysis of the papers shows that most papers focus on the analysis of data using process mining and less on describing the process and difficulty of exporting and extracting HIS-data and transforming them into event logs. Data from HIS is described as noisy, incomplete, and complex. This results in a complexity of the mining models, which is due to the lack of data quality on the one hand, but also to the high flexibility of the treatment processes in hospitals.

With regard to process representation and semantification, it can be noted that none of the papers examines the use of process mining for case acquisition for CBR. Most approaches rely on a procedural process modeling language, while 7 papers chose a declarative approach. The authors usually justify this approach by the suitability of declarative approaches for very flexible processes.

In applying similarity measures to oncological processes, the authors see particular challenges in the fact that the processes are highly flexible and change over time as research progresses. Specific challenges for oncological data that differ from other medical domains were not mentioned.

The application of process mining in oncology especially focuses on the control flow perspective. This is probably partly due to the fact that the control flow perspective is often used as the basis for the other process mining perspectives [13]. In terms of methodology, the ad hoc approach is followed mostly by the papers. Compared to the other methodology, it can cope with the complexity of real-world clinical processes [7]. In technical terms, the authors used the heuristic miner most often, arguing that the miner is particularly good at handling noisy data. The most widely used software is ProM.

The results provide a basis for future research in the field of case acquisition from oncological procedural data in HIS using process mining. The investigation of approaches to case acquisition using process mining and the answering of the question of the transferability of the approaches to oncology would be of particular interest. Also, the analysis of data and data quality in German oncology departments in the context of process mining would be of interest for further research. It would also be interesting to systematically investigate the potentials of process mining in CBR approaches.

References

1. Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications* **7**(1), 39–59 (1994). <https://doi.org/10.3233/AIC-1994-7104>

2. Arik Senderovich, Kyle E. C. Booth, J. Christopher Beck: Learning scheduling models from event data. *Proceedings of the International Conference on Automated Planning and Scheduling* **29**, 401–409 (2019), <https://www.aaai.org/ojs/index.php/ICAPS/article/view/3504>
3. Bergmann, R.: *Experience Management: Foundations, Development Methodology, and Internet-Based Applications*, Lecture Notes in Artificial Intelligence, vol. 2432. Springer (2002)
4. Bethel, A., Rogers, M.: A checklist to assess database-hosting platforms for designing and running searches for systematic reviews. *Health information and libraries journal* **31**(1), 43–53 (2014). <https://doi.org/10.1111/hir.12054>
5. Bettencourt-Silva, J.H., Clark, J., Cooper, C.S., Mills, R., Rayward-Smith, V.J., de La Iglesia, B.: Building data-driven pathways from routinely collected hospital data: A case study on prostate cancer. *JMIR medical informatics* **3**(3), e26 (2015). <https://doi.org/10.2196/medinform.4221>
6. Caron, F., Vanthienen, J., Baesens, B.: Healthcare analytics: Examining the diagnosis–treatment cycle. *Procedia Technology* **9**, 996–1004 (2013). <https://doi.org/10.1016/j.protcy.2013.12.111>
7. Caron, F., Vanthienen, J., Vanhaecht, K., van Limbergen, E., Deweerdt, J., Baesens, B.: A process mining-based investigation of adverse events in care processes. *Health information management : journal of the Health Information Management Association of Australia* **43**(1), 16–25 (2014). <https://doi.org/10.1177/183335831404300103>
8. Choudhury, N., Ara, S.: A survey on case-based reasoning in medicine. *International Journal of Advanced Computer Science and Applications* **7**(8) (2016). <https://doi.org/10.14569/IJACSA.2016.070820>
9. Claes, J., Poels, G.: Process mining and the prom framework: An exploratory survey. In: La Rosa, M. (ed.) *Business Process Management Workshops*, Lecture Notes in Business Information Processing, vol. 132, pp. 187–198. Springer Berlin Heidelberg, Berlin/Heidelberg (2013). https://doi.org/10.1007/978-3-642-36285-9_19
10. Dagliati, A., Sacchi, L., Zambelli, A., Tibollo, V., Pavesi, L., Holmes, J.H., Bellazzi, R.: Temporal electronic phenotyping by mining careflows of breast cancer patients. *Journal of Biomedical Informatics* **66**, 136–147 (2017). <https://doi.org/10.1016/j.jbi.2016.12.012>
11. Encke, A., Kopp, I., Selbmann, H.K.: Bedeutung der S1-, S2-, S3-Leitlinien. *Allgemein- und Viszeralchirurgie up2date* **3**(04), 257–267 (2009). <https://doi.org/10.1055/s-0029-1185952>
12. Gatta, R., Lenkowicz, J., Vallati, M., Rojas, E., Damiani, A., Sacchi, L., de Bari, B., Dagliati, A., Fernandez-Llatas, C., Montesi, M., Marchetti, A., Castellano, M., Valentini, V.: pminer: An innovative r library for performing process mining in medicine. In: ten Teije, A., Popow, C., Holmes, J.H., Sacchi, L. (eds.) *Artificial intelligence in medicine*, Lecture notes in computer science Lecture notes in artificial intelligence, vol. 10259, pp. 351–355. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59758-4_42
13. Ghasemi, M., Amyot, D.: Process mining in healthcare: a systematised literature review. *International Journal of Electronic Healthcare* **9**(1), 60 (2016). <https://doi.org/10.1504/IJEH.2016.078745>
14. Günther, C.W., Rozinat, A.: Disco: discover your processes. In: Lohmann, N., Moser, S. (eds.) *Proceedings of the Demonstration Track of the 10th International Conference on Business Process Management (BPM 2012)*. pp. 40–44. CEUR Workshop Proceedings, CEUR-WS.org (2012)

15. Günther, C.W., van der Aalst, W.M.P.: Fuzzy mining – adaptive process simplification based on multi-perspective metrics. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) *Business process management*, Lecture Notes in Computer Science, vol. 4714, pp. 328–343. Springer, Berlin (2007). https://doi.org/10.1007/978-3-540-75183-0_24
16. Haux, R.: *Strategic information management in hospitals: An introduction to hospital information systems*. Health informatics series, Springer, New York (2004), <http://www.loc.gov/catdir/enhancements/fy0818/2003059129-d.html>
17. Homayounfar, P.: Process mining challenges in hospital information systems. pp. 1135–1140 (01 2012)
18. Huang, Z., Gan, C., Lu, X., Huan, H.: Mining the changes of medical behaviors for clinical pathways. *Studies in health technology and informatics* **192**, 117–121 (2013)
19. Kelly, A.M.: Evidence-based practice: an introduction and overview. *Seminars in roentgenology* **44**(3), 131–139 (2009). <https://doi.org/10.1053/j.ro.2009.03.010>
20. Kitchenham, B., Charters, S.: *Guidelines for performing systematic literature reviews in software engineering* (2007)
21. Kolodneer, J.L.: Improving human decision making through case-based decision aiding. *AI Magazine* **12**(2), 52 (1991). <https://doi.org/10.1609/aimag.v12i2.895>, <https://www.aaai.org/ojs/index.php/aimagazine/article/view/895>
22. Kurniati, A.P., Johnson, O., Hogg, D., Hall, G.: Process mining in oncology: A literature review. In: *Proceedings of the 6th International Conference on Information Communication and Management ICICM 2016*. pp. 291–297. IEEE Press, Piscataway, NJ (2016). <https://doi.org/10.1109/INFOCOMAN.2016.7784260>
23. Kurniati, A.P., McInerney, C., Zucker, K., Hall, G., Hogg, D., Johnson, O.: A multi-level approach for identifying process change in cancer pathways. In: Di Francescomarino, C., Dijkman, R., Zdun, U. (eds.) *BUSINESS PROCESS MANAGEMENT WORKSHOPS*, Lecture Notes in Business Information Processing, vol. 362, pp. 595–607. Springer, [Place of publication not identified] (2020). https://doi.org/10.1007/978-3-030-37453-2_48
24. Kurniati, A.P., Rojas, E., Hogg, D., Hall, G., Johnson, O.A.: The assessment of data quality issues for process mining in healthcare using medical information mart for intensive care iii, a freely available e-health record database. *Health informatics journal* **25**(4), 1878–1893 (2019). <https://doi.org/10.1177/1460458218810760>
25. Kusuma, G.P., Hall, M., Gale, C.P., Johnson, O.A.: Process mining in cardiology: A literature review. *International Journal of Bioscience, Biochemistry and Bioinformatics* **8**(4), 226–236 (2018). <https://doi.org/10.17706/ijbbb.2018.8.4.226-236>
26. Lamy, J.B., Sekar, B., Guezennec, G., Bouaud, J., Séroussi, B.: Explainable artificial intelligence for breast cancer: A visual case-based reasoning approach. *Artificial Intelligence in Medicine* **94**, 42–53 (2019). <https://doi.org/10.1016/j.artmed.2019.01.001>
27. Levay, P., Craven, J.: *Systematic searching: Practical ideas for improving results*. Facet Publishing (2019)
28. Lohr, K.N., Field, M.J. (eds.): *Clinical practice guidelines: Directions for a new program*, Publication IOM, vol. 90-08. National Academy Press, Washington, D.C (1990). <https://doi.org/10.17226/1626>
29. Maggi, F.M., Bose, R.P.J.C., van der Aalst, W.M.P.: A knowledge-based integrated approach for discovering and repairing declare maps. In: Salinesi, C., Norrie, M.C., Pastor, O. (eds.) *Advanced Information Systems Engineering*, Lecture Notes in Computer Science, vol. 7908, pp. 433–448. Springer Berlin Heidelberg, Berlin/Heidelberg (2013). https://doi.org/10.1007/978-3-642-38709-8_28

30. Meng, W., Ou, W., Chandwani, S., Chen, X., Black, W., Cai, Z.: Temporal phenotyping by mining healthcare data to derive lines of therapy for cancer. *Journal of Biomedical Informatics* **100**, 103335 (2019). <https://doi.org/10.1016/j.jbi.2019.103335>
31. Müller-Bloch, C., Kranz, J.: A framework for rigorously identifying research gaps in qualitative literature reviews. In: *ICIS (2015)*
32. Peterson, J.L.: Petri nets. *ACM Computing Surveys (CSUR)* **9**(3), 223–252 (1977). <https://doi.org/10.1145/356698.356702>
33. Rinner, C., Helm, E., Dunkl, R., Kittler, H., Rinderle-Ma, S.: An application of process mining in the context of melanoma surveillance using time boxing. In: Daniel, F., Sheng, Q.Z., Motahari, H. (eds.) *Business Process Management Workshops, Lecture Notes in Business Information Processing*, vol. 342, pp. 175–186. Springer International Publishing, Cham (2019). https://doi.org/10.1007/978-3-030-11641-5_14
34. Robinson, K.A., Saldanha, I.J., Mckoy, N.A.: *Frameworks for Determining Research Gaps During Systematic Reviews*. Rockville (MD) (2011)
35. Rojas, E., Munoz-Gama, J., Sepúlveda, M., Capurro, D.: Process mining in healthcare: A literature review. *Journal of Biomedical Informatics* **61**, 224–236 (2016). <https://doi.org/10.1016/j.jbi.2016.04.007>
36. Rs Ronny Mans, Van der Aalst, Rjb Rob Vanwersch: *Process mining in healthcare : opportunities beyond the ordinary*. Computer Science (2013)
37. van der Aalst: *Process mining manifesto*. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) *Business Process Management Workshops, Lecture Notes in Business Information Processing*, vol. 99, pp. 169–194. Springer Berlin Heidelberg, Berlin, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28108-2_19
38. van der Aalst, W.M.P., Pesic, M., Schonenberg, H.: Declarative workflows: Balancing between flexibility and support. *Computer Science - Research and Development* **23**(2), 99–113 (2009). <https://doi.org/10.1007/s00450-009-0057-9>
39. van der Spoel, S., van Keulen, M., Amrit, C.: Process prediction in noisy data sets: A case study in a dutch hospital. In: Mylopoulos, J., Rosemann, M. (eds.) *Data-Driven Process Discovery and Analysis, Lecture Notes in Business Information Processing*, vol. 162, pp. 60–83. Springer Berlin Heidelberg, Berlin/Heidelberg (2013). https://doi.org/10.1007/978-3-642-40919-6_4
40. Weijters, A.J.M.M., Aalst, van der, W.M.P., Alves De Medeiros, A.K.: *Process mining with the HeuristicsMiner algorithm*. BETA publicatie : working papers, Technische Universiteit Eindhoven (2006)
41. Williams, R., Rojas, E., Peek, N., Johnson, O.A.: Process mining in primary care: A literature review. *Studies in health technology and informatics* **247**, 376–380 (2018)
42. Wohlin, C.: Guidelines for snowballing in systematic literature studies and a replication in software engineering. In: Shepperd, M., Hall, T., Myrtveit, I. (eds.) *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*. pp. 1–10. ACM, New York, NY (2014). <https://doi.org/10.1145/2601248.2601268>

Student Graduation Projects in the Context of Framework for AI-Based Support of Early Conceptual Phases in Architecture [★]

Viktor Eisenstadt^{1,2}, Christoph Langenhan³,
Klaus-Dieter Althoff^{1,2}, Andreas Dengel²

¹University of Hildesheim, Institute of Computer Science
Samelsonplatz 1, 31141 Hildesheim, Germany

²German Research Center for Artificial Intelligence (DFKI)
Trippstadter Strasse 122, 67663 Kaiserslautern, Germany
{viktor.eisenstadt, klaus-dieter.althoff}@dfki.de

³Chair of Architectural Informatics, Technical University of Munich
Arcisstrasse 21, 80333 Munich, Germany
langenhan@tum.de

Abstract In this paper, current, past, and planned student graduation projects in the context of MetisCBR, the distributed AI framework for intelligent support of the early room configuration process in architectural design, will be presented. During the last years, a number of such projects were initiated to achieve a master’s or bachelor’s degree. All these projects have in common that they intend to extend the currently available functionalities of the framework with new features using the modern AI techniques and trends, such as explainable AI or generative adversarial nets, in order to keep up with the recent AI developments. For each project, a summary of the concept(s), results of the experiments (if any), and the current status (e.g., defended or ongoing) will be presented. The main goal of this paper is to reward the student contributions to the MetisCBR framework by making them visible to the research community.

1 Introduction

MetisCBR¹ is a framework for AI-based support of early conceptual phases in architectural design and was initially created 2015 as a master thesis project (during the research project Metis²) in the form of a retrieval engine for similar building designs based on established artificial intelligence technologies *case-based reasoning* (CBR) and multi-agent systems (MAS). In the next years, the framework was gradually extended with additional functionalities, and possesses currently (2020), the following features to support the conceptual creation of building designs in the form of *abstract graph-based room configurations*:

[★] Copyright © 2020 by the paper’s authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

¹ <http://veisen.de/metiscbr>

² <https://www.ar.tum.de/en/ai/research/ksd-research-group/funded-projects/>

1. *Retrieval* – The system looks for similar room configurations for the given query using attribute-value-based or graph-matching-based search strategies. The former is used for a broad general search for inspiring designs, the latter to find exact or almost identical structure of the query within other graphs, for example to examine how the current structure is used in another context. For each search process, a group of agents, placed in a container, is responsible.
2. *Suggestion* – This functionality was developed to recommend the possible next design steps to the designer based on her previous design steps. A step is an action, e.g. *Add*, *Delete*, or *Change type* of the room. Using recurrent neural networks (RNN) and the design steps record (the *process chain*), the system suggests the next step using the most similar previous process chains.
3. *Adaptation* – The framework applies convolutional neural networks (CNN) in the form of the currently popular GAN structure (generative adversarial nets), to produce a number of possible evolutions of the current room configuration to show the designer how it might look in the future merging its feature matrix with the matrices of the most similar previously saved designs and letting the system decide which evolution can be considered real.
4. *Explanation* – Using the methods of explainable AI (XAI) with explanation patterns *Transparency*, *Justification*, and *Relevance*, the system can enrich the results of the previous three modules with *explanations* that contain the contextual insights into the currently executed process. For example, such explanations can provide information on search patterns used for retrieval or which steps of the current session were used to produce a step suggestion.

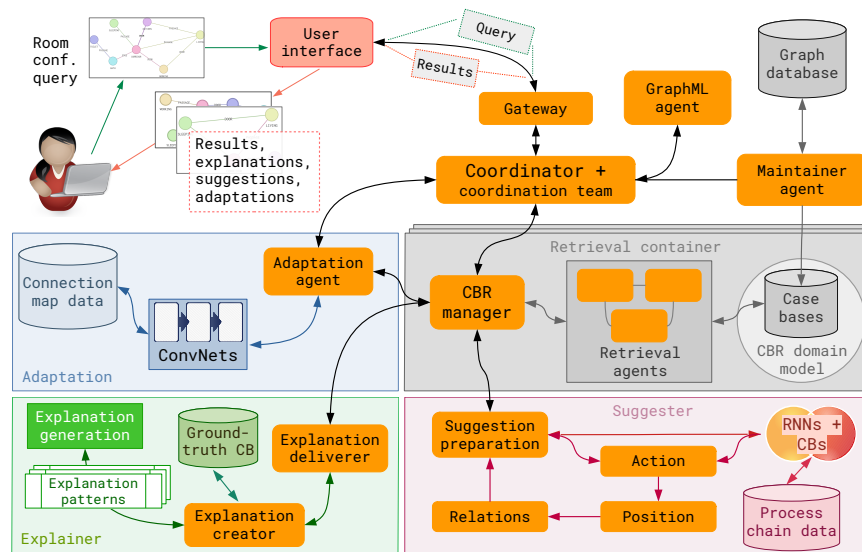


Figure 1. Overview of the current system architecture of MetisCBR.

Being now part of a PhD thesis, MetisCBR was specifically conceptualized for use in student graduation projects in order to conceptualize new functionalities based on the already existing ones and test and implement them if they prove or seem promising. In the next sections, a selection of the most notable graduation projects that had influence on the further development of the framework will be presented. Subsequently, we provide a summary of other relevant projects that relate to the framework but do not directly extend or evaluate it (e.g., surveys).

2 BDI-Based Explainable AI Component

The first graduation projects that will be described in this paper extend MetisCBR with an additional explanation module, the *BDI-Explainer*, that is based on the established multi-agent systems paradigm **B**elief, **D**esire, **I**ntention. This new explainer was inspired by the research work by Broekens et al. [1] and based on its complexity it was divided into two separate graduation projects: an already defended master thesis [4] dedicated to conceptualization and future-proof of the concept by comprehensive evaluation among the targeted user group of MetisCBR (architects), and the currently ongoing bachelor thesis that aims at implementation and quantitative evaluation of this BDI-based explanation module.

The BDI-based explainer (see Figure 2) makes use of different types of knowledge in the form of Beliefs (architectural knowledge of the system), Desires (explanation goals), and Intentions (current action to generate an explanation). The architectural knowledge is represented by the commonly used architectural technical terms and vocabularies in the form of typologies or taxonomies. Identically to the other explainers [2], the BDI-Explainer makes use of explanation patterns described in Section 1. The patterns are used as the current explanation goals (e.g., *justify the suitability of the suggested design step*). The *explanation generation* component provides the user with the explanation expression.

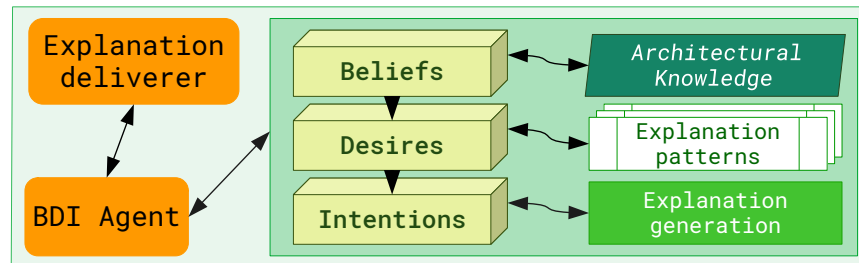


Figure 2. The planned implementation of the BDI-Explainer (figure adapted from [4]).

The evaluation of the concept of the BDI-Explainer revealed that 75% of the participated architects find the explanations in architecture modeling software useful and helpful for understanding of its functionality, however, this does not stimulate the creativity. Currently, the BDI-Explainer is being implemented using the multi-agent systems framework JADE and its BDI extension BDI4JADE [5].

3 Construction of Spatial Layouts with Game Theory

In this currently ongoing master thesis, it is planned to apply game theory, the well-known business negotiation technique, to construct an optimal room configuration (an early representation of a floor plan) based on predefined optimum criteria and the negotiation strategy. Game theory is one of the core features of cooperative multi-agent systems, it provides the relevant agents with a means to achieve an optimal agreement for distribution of the currently planned tasks among them. If executed properly as planned, the game-theory based cooperation strategy usually results in an optimal outcome for each of the collaborating agents.

Game theory was already applied for a multitude of domains, architectural design is among them. However, for construction of an optimal spatial layout of a building, to the best of our knowledge, this technique was not applied before. The task of the master thesis is to explore the possibilities of game theory for early phases of architectural design and create a concept for this application. In general, it is planned to elaborate a number of specific negotiation strategies that the agents can use to find an optimal configuration for the current design task, e.g., an apartment for an elderly married couple or a standalone multi-functional bungalow. Two general agent setups are possible to apply a strategy:

1. *Holistic* – Every agent is able to propose a solution for the configuration of rooms available in the layout, other agents might or might not suggest the improvements and are able to justify their suggestions using utility functions.
2. *Distributed* – In this setup, each agent is responsible for one room (or room type) only and has a task of placing this room in the best possible position in the configuration. For example, the agent responsible for living rooms will claim the central position for them and easy access from other rooms.

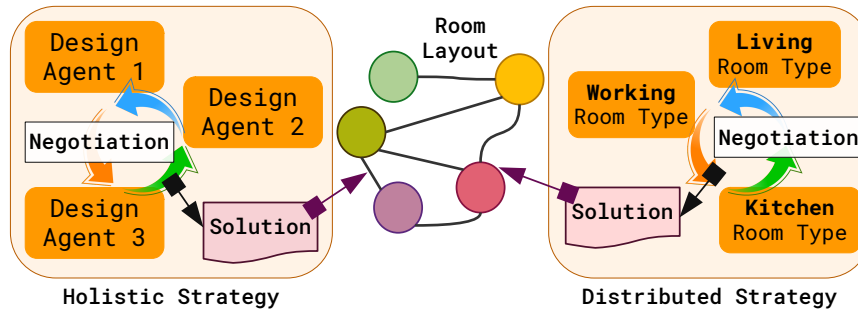


Figure 3. Strategies used for game theory-based spatial layout.

In Figure 3, an overview of both types of the agent setups is shown. After conceptualization, the game-theory-based construction strategies should be prototypically implemented using the aforementioned MAS frameworks JADE and/or BDI4JADE and evaluated by a representative of the architecture domain.

4 Speech UI-Supported Room Configuration Design

Similarly to the BDI-Explainer, this project is also a combination of two graduation projects, where a master thesis contains the research and the detailed concept with implementation instructions, complemented by a practical bachelor thesis which will contain the implementation and quantitative evaluation.

In this combined project a human speech-controlled component for early phases of architectural design should be conceptualized and implemented in MetisCBR and its web-based user interface (UI) RoomConf Editor³. Inspired by the modern natural language generation (NLG) assistant systems, such as Apple's Siri and Amazon's Alexa, the collaboration between the system and the architect will be enriched by a human dialog-based module that should listen to the architect's voice commands and questions via the specific web browser API (application programming interface), forward them to the backend of the system for parsing and producing the NLG-based answer that accompanies the achieved results and reproduce it using a human voice imitation in the UI.

In general, all four main functionalities of MetisCBR described in Section 1 should be covered by the speech-based dialog with the system. All functions that can be executed with the non-voice interaction methods, such as requesting the next design action suggestion should be also possible to execute with a voice command. In specific situations, a real dialog with the system should be possible as well, for example to ask for improvement if certain conditions in the layout look doubtful. Another example is providing the step-by-step guidance to the user during a specific design task. Two different methods should cover both situations described above: dialog trees and artificial neural networks (ANN). A gateway selects the most suitable parsing method for the current user expression. In Figure 4, an overview of the voice-controlled design support module is shown.

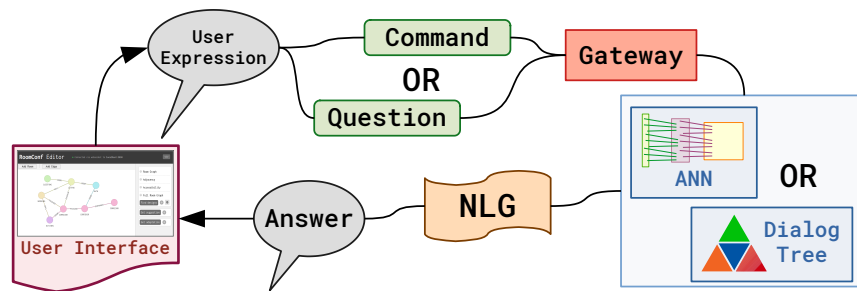


Figure 4. Overview of the voice-controlled design component.

An example of a dialog tree and instructions for implementation of ANNs were already described in the aforementioned defended master thesis [7]. In the bachelor thesis, it is planned to extend them, implement a proof of concept, and comparatively evaluate using different automatically applied design scenarios.

³ <http://veisen.de/metiscbr/roomconf/>

5 Surveys and Reviews

Besides the conceptual and practical projects that directly extend the MetisCBR framework, a number of thematically related approach surveys were assigned as graduation projects as well. Their goal is to collect the knowledge available in the research domain and so determine the position of the framework among the related approaches. The following list describes the most notable reviews:

- *MAS in Architecture* [6] – This already defended master thesis examined major monographies, journals, and conference proceedings to identify and classify the existing and past MAS approaches that aim at supporting the design phases in architecture. Using a specific self-defined classification, each approach was assigned into a category and feature-wise compared with the systems from this category. As a quintessence, the work suggests a meta-model for MAS systems in Architecture, including the MetisCBR framework.
- *Data Augmentation and Augmented Reality in Architecture* – In this currently ongoing master thesis literature review, systems and approaches that enrich architectural design with methods that introduce augmented reality and/or make use of data augmentation for datasets are examined and classified. The goal is to identify the relevant approaches that might provide a benchmark for MetisCBR, as it is planned to extend the framework with augmented reality features, e.g., to map the room layout onto different building contexts.
- *Survey of Open Source CAD Systems* [3] – In this defended master thesis, the currently available open source tools for computer-aided design (CAD), were examined. The goal was to estimate the acceptance of such tools in comparison to proprietary tools and propose a holistic marketing strategy that MetisCBR (if it will be decided to open source it completely) and other approaches can use to find a proper position and user group on the market.

References

1. Broekens, J., Harbers, M., Hindriks, K., Van Den Bosch, K., Jonker, C., Meyer, J.J.: Do you get it? user-evaluated explainable bdi agents. In: German Conference on Multiagent System Technologies. pp. 28–39. Springer (2010)
2. Espinoza-Stapelfeld, C., Eisenstadt, V., Althoff, K.D.: Comparative quantitative evaluation of distributed methods for explanation generation and validation of floor plan recommendations. In: ICAART-2018. pp. 46–63. Springer (2018)
3. Kromm, E.: "Analysis of Status Quo of Open Source CAD Tools and Criteria for Improvement of their Market Position". University of Hildesheim (2019)
4. Mikyas, A.: "Concept for Development of an Explanation Component for BDI Agents to Support the Design Phases in Architecture". University of Hildesheim (2018)
5. Nunes, I.: Improving the design and modularity of bdi agents with capability relationships. In: Dalpiaz, F., Dix, J., van Riemsdijk, M.B. (eds.) Engineering Multi-Agent Systems. pp. 58–80. Springer International Publishing, Cham (2014)
6. Trabelsi, G.: "Multi-Agent Systems in Architecture – Classification and Evaluation of Approaches for Distributed Agent-Based Support of Design Phases". University of Hildesheim (2020)
7. Younis, B.: "Concept of a Dialog-Based Human-Machine-Interaction for the Domain of Architecture Based on Modern AI Methods". University of Hildesheim (2020)

A Concept for the Automated Reconfiguration of Quadcopters

Kaja Balzereit¹[0000-0001-9203-5902], Marta Fullen¹, and Oliver Niggemann²

¹ Fraunhofer IOSB, Industrial Automation Branch (INA), Fraunhofer Center for Machine Learning, Lemgo, Germany
`{name.surname}@iosb-ina.fraunhofer.de`

² Faculty of Machine Construction, Helmut-Schmidt-University, Hamburg, Germany
`oliver.niggemann@hsu-hh.de`

Abstract. Quadcopters are susceptible to internal and external influences, many of which may lead to faults. To ensure a safe and reliable flight, the quadcopter needs to recover autonomously from faults. However, existing approaches mainly rely on parametrical faults or require a predefinition of possible faults which is not realistic for a complex real-world scenario. The recovery from unforeseen faults and structural faults like a failing engine is still an open research gap.

Hence, in this paper, a concept for the automated reconfiguration, i.e. the automated recovery from a fault, which only uses information about non-faulty system behavior and is able to handle structural changes is presented. From the information about non-faulty behavior a non-faulty system model is created using established machine learning methods. Thus, faults are detected by learned model and no pre-definition of faults is needed. The system structure is modeled using a logical calculus which allows for modeling available system parts and the causal coherences between these.

The approach is applied to a simulation of a quadcopter which underlies a structural fault. It is shown that the approach extends the capabilities of a quadcopter to handle faults autonomously and ensure stability and reliability.

Keywords: Automated Reconfiguration · Symptom Generation · Fault Recovery · Quadcopter

1 Introduction

Unmanned Aerial Vehicles (UAVs) are an emerging technology of great interest in military and civil applications [21,11]. The market for UAVs has emerged especially in the last years and is expected to rise continuously [11,7]. One type of UAVs, the quadcopters which consist of four rotors that can be controlled independently from each other, is in the scope of most research studies [21]. Since

Copyright © 2020 by the paper's authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

quadcopters operate in an open world, the requirements towards reliability and safety are very high. Today, due to many safety concerns, the usage of quadcopters is subject to massive restrictions [11]. It stems from a fact that minor UAV fault can lead to major consequences, including huge damage to humans or environment. Hence, quadcopters need to be designed to be robust to environmental disturbances and tolerant towards internal faults. Currently, controllers used for Cyber-Physical Systems, including quadcopters, are static and at most applicable to a predefined set of faults [24]. An automated reconfiguration of the control is needed to maintain a stable flight in presence of faults [5]. Reconfiguration is the task of recovering a valid system state after a fault has occurred [3]. However, the automated reconfiguration for quadcopters is still an open research gap due to some unanswered research questions, two of which follow.

RQ1: How can automated reconfiguration handle unforeseen faults? Existing approaches on fault-tolerant control are mostly based on an enumeration of known faults and the storage of control instructions specific to these faults [21]. However, when it comes to unknown faults, these approaches can no longer guarantee stable control. Hence, in this paper, a concept which needs no information about known faults but only works on information about non-faulty behavior is presented. Quadcopters contain a multitude of sensors, continuously logging data during the flight. This huge amount of data can be analyzed in an intelligent way using Machine Learning (ML) methods. These methods enable learning a model of the system from historical data and can then be used to detect anomalous behavior, which might indicate the presence of faults [18]. The data used for training contains only measurements from non-faulty flights, thus deviations from non-faulty behavior are handled as candidates for faults.

RQ2: Which formalism is able to handle structural faults? Control theory in general is concerned with a static system model. However, when major faults like an engine failure or a rotor ripping off occur, the system is no longer representative, leading to the control becoming invalid. Logical reasoning can be used here to draw conclusions about the impact of a fault as well as still available and functional components and actions [3]. Thus, new control instructions can be determined even in the presence of major faults.

The contribution of this paper is twofold: (1) A concept for the automated reconfiguration of quadcopters that handles unforeseen faults is presented. Therefore, only information and data about non-faulty flights is used. Thus, faults are detected as deviations from non-faulty behavior and do not have to be known a-priori. (2) An encoding of the reconfiguration problem into first-order logic (FOL) is presented that allows for analyzing the extent of structural faults is presented. Thus, also major faults like component failure (e.g. an engine) can be handled.

Please note that the scope of this paper is not to handle one given fault in an optimal way but to present a concept that restores a safe flight in the presence of unforeseen faults and disturbances. The paper is structured as follows: First, in section 2 the related work is presented and discussed. Then, the solution concept

is presented in section 3. In section 4, using a simulation of a quadcopter, the applicability of the solution concept is evaluated.

2 Related Work

Most research on quadcopter control is concerned with fault-tolerant control (FTC) [22,17]. The goal of FTC is to maintain a stable flight in the presence of wind as well as actuator and sensor faults. Therefore, the quadcopter is described in quantitative terms [26] using the equations

$$\begin{aligned}x(t+1) &= f(x(t), u(t)) \\ y(t) &= g(x(t), u(t))\end{aligned}\tag{1}$$

where x describes the system states, u describes the input (e.g. the rotor velocities) and y describes the output (e.g. the altitude and attitude of the quadcopter). f, g are functions describing the properties of the quadcopter. Using this equation system, the optimal input to change the attitude or altitude of the quadcopter can be determined, and thus, the quadcopter can adapt to environmental changes of the wind speed or sensor faults. However, major disturbances like a rotor ripping off or the battery failing cannot be represented by a static model (1) and require a new type of model. These disturbances lead to changed dynamics such that the functions f, g are no longer valid and need to be adapted. However, this adaptation cannot be done online but requires expert knowledge. To handle these changes, a reconfiguration of the controller is needed [5].

Lunze [14] presented a concept towards reconfigurable control for UAVs using overdetermined sets of equations. However, this approach requires explicit modeling of the quadcopter behavior and thus a large amount of expert knowledge. Wang et al. [27] presented a combination of classical control and constraint satisfaction. The scope of this work is slightly different: no faults are handled but an optimal control for a given path is searched. Chen et al. [6] presented an approach to reconfiguration of actuator faults by an advanced estimation procedure. However, the faults need to be modeled explicitly. Thus, no unknown faults can be handled. Adaptive control algorithms, as presented by Huynh et al. [10] among others, are concerned with continuous disturbances like wind or varying parameters. Unknown or structural faults cannot be handled [21]. Robust control algorithms as presented by Thanh et al. [23] and Ton et al. [25] handle parametric uncertainties and are even adaptable to nonlinear disturbances. However, no structural faults can be handled [21].

The reconfiguration concept presented here can be seen as an extension to classical control theory. The goal is not to determine control instructions for an optimal flight, but to identify the necessary actions to recover a stable flight in the presence of faults.

3 Solution Concept

The goal of automatic reconfiguration is to manipulate the system inputs to restore valid system behavior [3]. To perform automatic reconfiguration, some kind of redundancy is necessary [5]. This can be either physical redundancy, e.g. duplicate components or sensors, or analytical redundancy, i.e. information about the relation between different values measured by the system. Quadcopters in general contain numerous redundancies to ensure safety and reliability requirements are fulfilled. Typical examples for physical redundancy are multiple engines, multiple batteries or multiple sensors. Analytical redundancy can be asserted through knowledge about coherences and relations between sensor and actuator values.

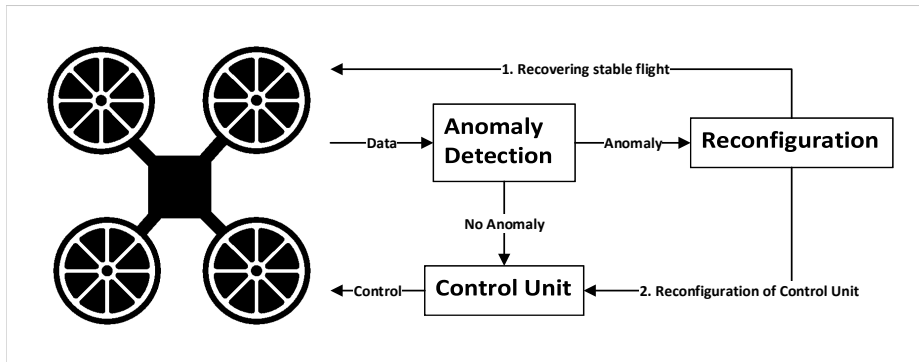


Fig. 1. Concept for the automated reconfiguration of quadcopters. First, a symptom generation is performed on the data. If at least on symptom is present, reconfiguration is performed in two steps.

Faults can be differentiated using the component they concern. Thus, we divide actuator faults which are usually modeled as loss of effectiveness of one rotor, sensor faults which are modeled as sensors returning wrong values and structural/system faults which affect system components like engines or batteries. Actuator faults can be handled using robust control methods, sensor faults usually are handled using Kalman filters [15]. In general, not every possible fault and its consequences is known a-priori because quadcopters operate in a non-deterministic environment, many different factors like wind speed, rain and air humidity have an impact on the behavior of the quadcopter. Thus, foreseeing every possible consequence of environmental influences on the quadcopters behavior and enumerate every possible fault is impossible. The goal of reconfiguration is to adapt the rotor speeds to recover a stable flight in case of every unforeseen fault or at least to perform a safe emergency landing.

The basic concept of automatic reconfiguration for quadcopters is shown in Figure 1. The reconfiguration system operates while the quadcopter is flying and continuously checks for deviations. First, the data delivered by the sensors

of the quadcopter is compared to the learned models, checking for deviations and the presence of symptoms. If no symptoms are present, no further actions are needed and neither the control unit nor the system goal are changed. In case of at least one symptom, the reconfiguration unit first estimates the extent of the deviation. Then, reconfiguration is performed in two steps [5]:

1. A set of actions that moves the quadcopter back into a valid flight is searched and applied directly.
2. A controller that stabilizes the quadcopter in its valid flight in the presence of faults (e.g. a different control structure due to a failure of one sensor) is determined using well known controller design methods.

In some cases, the reconfiguration to a valid flight is no longer possible since the damage is too high. Then, during the first reconfiguration step, actions to perform an emergency landing or a return to launch, if possible, are applied until the quadcopter has landed. The control unit is not reconfigured, since a stable flight cannot be recovered.

3.1 Modeling the System Structure

To enable the reconfiguration unit to handle structural faults, it needs to reason about the consequences of a fault. Therefore, information about the causal coherences, i.e. the impact of a change in one component to other components is needed. Logic is used widely in Artificial Intelligence since it allows for modeling causal coherences and drawing logical conclusions about the system [20]. Basic physical and mathematical knowledge can be modeled in logic to support the reconfiguration unit in its decision making [13]. Using the logical calculus *Satisfiability Modulo the Theory of Linear Arithmetic* [4], also continuous variables (e.g. rotor velocity, wind speed, ...) can be modeled.

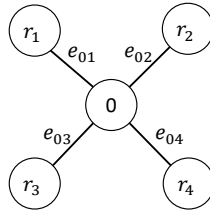


Fig. 2. Topology of a quadcopter.

To model the causal coherences of the quadcopter, first, the system topology is analyzed. Thus, the coherences between the components can be described in terms of a logical calculus. Therefore, the fuselage 0 and each rotor r_1, r_2, r_3, r_4 are modeled as nodes, the set of all nodes is represented by N . Since every rotor is connected to the fuselage, the edges are represented by $E = \{e_{01}, e_{02}, e_{03}, e_{04}\}$. Fig. 2 shows the resulting graph for a quadcopter.

3.2 Symptom Generation

The behavior of every component, i.e. the rotors and the fuselage, is monitored using component models [8]. Today, due to sophisticated learning methods, these models no longer need to be created manually but can be learned [19,2]. Comparing the current behavior of a component to the model, deviations indicating faults are detected. This information is encoded by a binary assignment $\omega : N \rightarrow \{\top, \perp\}$ which is true if the component behaves non-faulty and false otherwise. A connection between two components may only be used if the components are non-faulty, i.e. $b_{e_{ij}}^+ \Rightarrow \omega(i) \wedge \omega(j) \forall e_{ij} \in E$.

Thus, component faults are directly taken into account by the reconfiguration unit. Structural changes due to a component fault (e.g. a failing engine) can be represented by the component model of the corresponding rotor showing a deviation to the current behavior. Hence, the reconfiguration unit identifies the necessary changes under consideration of the impact of the fault.

Therefore, every components behavior is monitored using sophisticated machine learning methods utilizing a normal behavior system model. Here, a modeling formalism which transforms the data into a black- or gray-box representation is used. Such a structure does not explicitly model each observation but creates a new representation based on the normal behavior data. Depending on the formalism used, various measures describe how well the current status fits into the model. An example well-known algorithm is Self-Organizing Map [12], a type of neural network. The measure of fitting in this case is the quantization error, which is the difference between the current status location mapping and the best-matching neuron neighborhood in the model. If this quantization error is high, the component is assumed to behave anomalously, so a symptom is reported to the reconfiguration unit to trigger further actions.

For the creation of these models only data about non-faulty system behavior is required. Thus, no fault modes or an enumeration of known faults has to be given.

3.3 Reconfiguration

As mentioned above, the reconfiguration is performed in two steps: One step to regain a stable flight and one step to maintain this stable flight. Whilst the second step can be done using well-known controller design methods, the first step is still an open research gap [5].

For each rotor, the impact of acceleration and deceleration on the attitude and altitude of the quadcopter is modeled in terms of logical constraints, for example

If rotor r_1 is accelerated, the pitch angle decreases.

or

If all rotors are accelerated proportionally, the height increases.

Thus, the impact of changing rotor velocities on the behavior of the quadcopter can be modeled. Based on this, the reconfiguration unit is able to choose an intelligent combination of rotor accelerations and decelerations to recover a

stable flight, if possible. Otherwise, the reconfiguration unit tries to bring the quadcopter to a safe state (e.g. by a return to launch or an emergency landing).

To enable the reconfiguration method to change the velocity of the rotors, every connection is assigned with two binary variables that lead to an increase or decrease of the current velocity of the corresponding rotor. Therefore, for each edge $e \in E$ two binary variables $b_{e_{ij}}^+, b_{e_{ij}}^-$ that trigger an increase or decrease of the corresponding rotor are introduced. Thus, $b_{e_{ij}}^+ \rightarrow inc(r_j), b_{e_{ij}}^- \rightarrow dec(r_j)$. The predicates *inc*, *dec* indicate that the velocity of the corresponding rotor needs to be increased or decreased. How this is realized in detail needs to be defined by an expert, e.g. that an increase is always realized by increase the velocity given a fixed difference or a percentage amount. To avoid that both variables are set to true at the same time (which would require a simultaneous increase and decrease of the velocity of one rotor) the constraint $b_{e_{ij}}^+ \oplus b_{e_{ij}}^- \forall e_{ij} \in E$ is needed.

Thus, the reconfiguration problem is modeled as a first-order logic formula. If at least one symptom occurs, i.e. one component behaves anomalously, the reconfiguration unit determines for every connection, if the velocity of each rotor has to be increased, decreased, or does not have to be changed by setting the corresponding binary variable to true. Thus, the changes for recovering a stable flight are identified by a combination of acceleration and deceleration of rotor velocities.

4 Results

This section presents the results of symptom generation and reconfiguration experiments. The symptom generation approach has been tested on real quadcopter data, to validate the approach as feasible in real-life scenarios. Reconfiguration experiments utilize a simulation to verify the outcome of reconfigured and non-reconfigured fault scenario. The used simulation of the quadcopter is described in the appendix. The free variables in the logical formula created as described above are assigned with the current values of the sensors. Then, the formula is checked for satisfiability to determine which rotors needs to be accelerated and which need to be decelerated to recover a stable flight in the presence of faults using the Z3 solver [16].

4.1 Symptom Generation

To evaluate the concept, we show that it is indeed possible to detect anomalies in quadcopter behavior using machine learning-based modeling formalisms. Self-Organizing Map (SOM) model formalism is used to perform a preliminary analysis and investigate whether the methods are feasible to detect anomalies in quadcopter flight. We consider an approach feasible for symptom generation if it is possible, at least partially, to differentiate between normal behavior and anomalous behavior using the model.

The symptom generation is performed on data from an industrial drone, to ensure the first step of the concept is viable in real-life applications. Quadcopters are operated using the PX4, an open source flight control software for

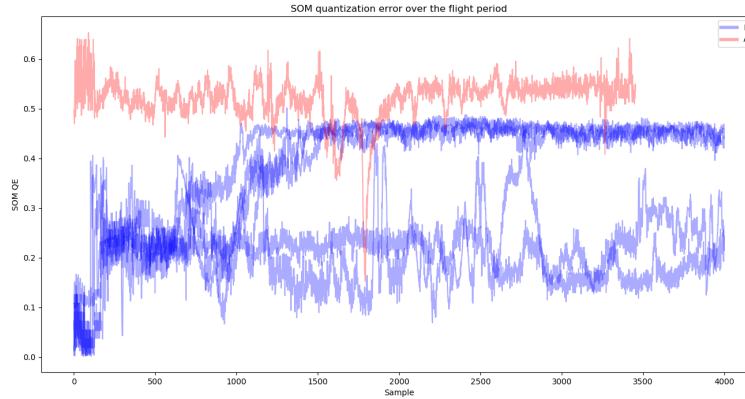


Fig. 3. SOM quantization error (QE) over time for anomalous (red) and non-anomalous (blue) flights. QE generated by a 10x10 SOM from a select combination of sensors and learned on non-anomalous data.

quadcopters and other unmanned vehicles. It allows logging device inputs (sensors etc.), internal states (CPU load, attitude etc.) and log messages. The log structure requires that sensors are organized within predefined sensor groups, however the sampling rates, and therefore timestamps, of different sensor groups are independent from each other. It is therefore only possible to match the values from one sensor group at a time. The SOM model of normal behavior has been learned from chosen sensor logs of quadcopter flights where no faults occurred and the drone was considered to behave entirely correct. This model has then been used to detect anomalies in faulty flight logs.

It is expected that the anomalous behavior data at least partially overlaps the normal behavior in terms of quantization error, however, the results have shown that some sensor combinations generate a quantization error much higher in the case of faulty behavior than normal behavior. This outcome creates a perfect opportunity for symptom generation, where the maximum quantization error of normal behavior is used as the error threshold for a symptom, and a symptom is reported as soon as the error crosses the threshold. Figure 3 illustrates the difference in quantization error over the initial flight period of flight: the error generated by SOM from the anomalous behavior data is decidedly higher than for normal behavior data.

4.2 Fault Scenario: Engine Failure

The Engine Failure fault scenario focuses on a quadcopter flight, where one of the four engines that provide acceleration to the rotors fails mid-flight. The flight begins with a stable flight at the height of 10 meters, until one engine fails and its corresponding rotors velocity decreases to 0. Only the three remaining rotors can

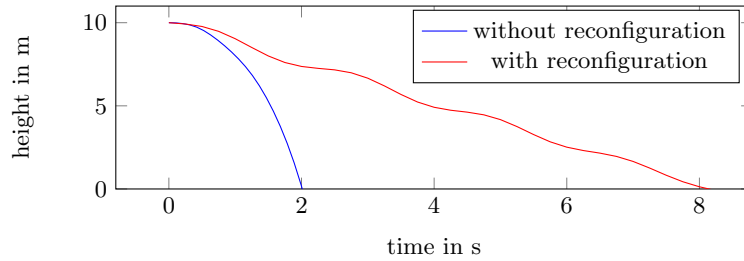


Fig. 4. Height of quadcopter with one engine failing at time $t_f = 0$ s.

be used to control the flight and the goal is to adjust their corresponding motor parameters in such a way that the drone does not suffer a catastrophic failure. Classical control is not capable of handling the new situation with one less rotor: the control unit only recognizes a deviation in the attitude of the quadcopter and tries to adapt the velocity of the quadcopter rotor such that stability is regained. It is not taken into consideration, that one rotor is non-functional. Even in the presence of a fault, all of the engines are controlled similarly. No distinction between available and crashed engine can be made. The control unit uses all four rotors to adapt to the deviation, even though only three rotors are still available. Thus, no stable flight can be regained leading to a crash of the quadcopter. The reconfiguration method based on logical calculus enables the quadcopter to handle the fault and avoid a crash. First, the faulty component, in this case the failing engine is identified using the component models. Then, the reconfiguration unit calculates new control instructions to stabilize the flight by increasing the velocities of the rotors which are still available. However, a stable flight cannot be regained since the disturbance of a failing engine is too severe. Thus, an emergency landing is performed.

The simulated behavior of quadcopters equipped with classical control approach and reconfiguration approach is shown in Figure 4. Using the classical control, which does not adapt to the changed structure of the quadcopter, the quadcopter crashes within approximately 2 seconds. Since the quadcopter accelerates while falling, the velocity when touching the ground is around 13.4 meters per second which can lead to massive damages of the quadcopter and the surroundings. When the reconfiguration is enabled, the fall of the quadcopter is decelerated. The quadcopter touches the ground after approximately 8.4 seconds with a velocity of approximately 1.3 meters per second. Thus, damage to the quadcopter and the surrounding can be reduced significantly.

5 Summary and Outlook

Quadcopter flights are susceptible to internal disturbances, like failing components, as well as environmental disturbances like winds. Today, FTC is commonly used to enable a quadcopter to maintain a stable flight even in the presence of

faults. However, FTC is focused on handling numerical faults, which, in addition, often have to be predefined. Thus, major faults which cause structural changes of the quadcopter cannot be handled by classical FTC. Therefore, this paper presents a concept for the automated reconfiguration to enable quadcopters to handle structural and unforeseen faults. The concept is based on the combination of a logic-based reconfiguration method. Faults are detected as deviations from models which are learned from non-anomalous behavior. Based on this information, reconfiguration is initiated – if necessary. During reconfiguration, stable flight of a quadcopter is described in terms of a logical calculus which allows for modeling condition of a stable flight and requirements to recover a stable flight, if possible. The approach is evaluated using an interactive simulation of a quadcopter. One of the four engines failing represents the structural fault. It is shown that without reconfiguration, the quadcopter crashes within 2 seconds and the velocity when touching the ground is high. With reconfiguration, the quadcopter touches the ground after 8 seconds with a far lower velocity.

Future work will focus on further fault scenarios like failures leading to a reduced engine performance or failing battery cells to prove the applicability of the concept. Then, also the scalability of the approach will be examined by using more detailed models.

Acknowledgments

This work was founded by the Fraunhofer Cluster of Excellence "Cognitive Internet Technologies".

A Appendix

Evaluation is performed using a simulation implemented in Modelica [9] is used. The flight of the quadcopter is described as a state space model (referring to [1])

$$\ddot{\phi} = \dot{\theta}\dot{\psi}a_1 + \dot{\theta}a_2\Omega_r + b_1U_2 \quad (2)$$

$$\ddot{\theta} = \dot{\phi}\dot{\psi}a_3 - \dot{\phi}a_4\Omega_r + b_2U_3 \quad (3)$$

$$\ddot{\psi} = \dot{\phi}\dot{\theta}a_5 + b_3U_4 \quad (4)$$

$$\ddot{x} = (\cos(\phi)\sin(\theta)\cos(\psi) + \sin(\phi)\sin(\psi))U_1/m \quad (5)$$

$$\ddot{y} = (\cos(\phi)\sin(\theta)\sin(\psi) - \sin(\phi)\cos(\psi))U_1/m \quad (6)$$

$$\ddot{z} = -g + (\cos(\phi)\cos(\theta))U_1/m \quad (7)$$

with

$$U_1 = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2), U_2 = b(\Omega_2^2 - \Omega_4^2), U_3 = d(-\Omega_1^2 + \Omega_3^2), \quad (8)$$

$$U_4 = d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2), \Omega_r = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4, \quad (9)$$

$$a_1 = \frac{I_{yy} - I_{zz}}{I_{xx}}, a_2 = \frac{J_x}{I_{xx}}, a_3 = \frac{I_{zz} - I_{xx}}{I_{yy}}, a_4 = \frac{J_r}{I_{yy}}, a_5 = \frac{I_{xx} - I_{yy}}{I_{zz}}, \quad (10)$$

$$b_1 = \frac{l}{I_{xx}}, b_2 = \frac{l}{I_{yy}}, b_3 = \frac{1}{I_{zz}}. \quad (11)$$

Variable	Value	Unit	Description
m	1.1	kg	mass of drone
J_r	$8.5 \cdot 10^{-4}$	$\text{kg} \cdot \text{m}^2$	rotor inertia
$I_{xx} = I_{yy}$	$1.96 \cdot 10^{-2}$	$\text{kg} \cdot \text{m}^2$	quadcopter inertia around x/y-axis
I_{zz}	$2.62 \cdot 10^{-2}$	$\text{kg} \cdot \text{m}^2$	quadcopter inertia around z-axis
l	0.21	m	length of arms
b	$9.29 \cdot 10^{-5}$	$\text{N} \cdot \text{s}^2$	thrust coefficient
d	$1.1 \cdot 10^{-6}$	$\text{N} \cdot \text{m} \cdot \text{s}^2$	drag coefficient

Table 1. Values for the parameters of the quadcopter


The values of the parameters are shown in Table 1 (also referring to [1]). The inputs of the system are represented by the velocities of the rotors $\Omega_1, \Omega_2, \Omega_3, \Omega_4$.

References

1. Alexis, K., Nikolakopoulos, G., Tzes, A.: Model predictive quadrotor control: attitude, altitude and position experimental studies. *IET Control Theory & Applications* **6**(12), 1812–1827 (2012)
2. Balzereit, K., Maier, A., Barig, B., Hutschenreuther, T., Niggemann, O.: Data-driven identification of causal dependencies in cyber-physical production systems. In: 11th International Conference on Agents and Artificial Intelligence (02 2019)
3. Balzereit, K., Niggemann, O.: Automated reconfiguration of cyber-physical production systems using satisfiability modulo theories. In: 3rd IEEE International Conference on Industrial Cyber-Physical Systems (2020)
4. Barrett, C., Tinelli, C.: Satisfiability modulo theories. In: *Handbook of Model Checking*, pp. 305–343. Springer (2018)
5. Blanke, M., Kinnaert, M., Lunze, J., Staroswiecki, M.: *Diagnosis and fault-tolerant control*, vol. 2. Springer (2006)
6. Chen, F., Lei, W., Tao, G., Jiang, B.: Actuator fault estimation and reconfiguration control for the quad-rotor helicopter. *International Journal of Advanced Robotic Systems* **13**(1), 33 (2016)
7. Cohn, P., Green, A., Langstaff, M., Roller, M.: Commercial drones are here: The future of unmanned aerial systems. McKinsey & Company (2017), <https://www.mckinsey.com/industries/capital-projects-and-infrastructure/our-insights/commercial-drones-are-here-the-future-of-unmanned-aerial-systems#>, called 22.04.2020
8. De Kleer, J., Brown, J.S.: Mental models of physical mechanisms and their acquisition. *Cognitive skills and their acquisition* pp. 285–309 (1981)
9. Elmqvist, H., Mattsson, S.E., Otter, M.: Modelica: The new object-oriented modeling language. In: 12th European Simulation Multiconference, Manchester, UK (1998)

10. Huynh, M.Q., Zhao, W., Xie, L.: L 1 adaptive control for quadcopter: Design and implementation. In: 2014 13th International Conference on Control Automation Robotics & Vision (ICARCV). pp. 1496–1501. IEEE (2014)
11. Joshi, D.: Drone technology uses and applications for commercial, industrial and military drones in 2020 and the future. *Business Insider* (2019), <https://www.businessinsider.com/drone-technology-uses-applications?r=DE&IR=T>, called 22.04.2020
12. Kohonen, T.: *Self-Organizing Maps*, Springer Series in Information Sciences, vol. 30. Springer-Verlag, Berlin Heidelberg, 3 edn. (2001)
13. Lake, B.M., Ullman, T.D., Tenenbaum, J.B., Gershman, S.J.: Building machines that learn and think like people. *Behavioral and brain sciences* **40** (2017)
14. Lunze, J.: From fault diagnosis to reconfigurable control: A unified concept. In: 2016 3rd Conference on Control and Fault-Tolerant Systems (SysTol). pp. 413–421. IEEE (2016)
15. Moghadam, M., Caliskan, F.: Actuator and sensor fault detection and diagnosis of quadrotor based on two-stage kalman filter. In: 2015 5th Australian Control Conference (AUCC). pp. 182–187. IEEE (2015)
16. de Moura, L., Bjørner, N.: Z3: An efficient SMT solver. In: International Conference on Tools and Algorithms for the Construction and Analysis of Systems. pp. 337–340. Springer (2008)
17. Nguyen, N.P., Hong, S.K.: Fault diagnosis and fault-tolerant control scheme for quadcopter uavs with a total loss of actuator. *Energies* **12**(6), 1139 (2019)
18. Niggemann, O., Frey, C.: Data-driven anomaly detection in cyber-physical production systems. *at-Automatisierungstechnik* **63**(10), 821–832 (2015)
19. Niggemann, O., Lohweg, V.: On the Diagnosis of Cyber-Physical Production Systems - State-of-the-Art and Research Agenda. In: Proceedings of the Twenty-Ninth Conference on Artificial Intelligence (2015)
20. Nilsson, N.J.: Logic and artificial intelligence. *Artificial intelligence* **47**(1-3), 31–56 (1991)
21. Shraim, H., Awada, A., Youness, R.: A survey on quadrotors: Configurations, modeling and identification, control, collision avoidance, fault diagnosis and tolerant control. *IEEE Aerospace and Electronic Systems Magazine* **33**(7), 14–33 (2018)
22. Tabata, A., Satoh, Y., Nakamura, H., Kato, K.: Adaptive fault tolerant control of quadcopter by using minimum projection method. In: IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society. pp. 2201–2206. IEEE (2018)
23. Thanh, H.L.N.N., Hong, S.K.: Quadcopter robust adaptive second order sliding mode control based on pid sliding surface. *IEEE Access* **6**, 66850–66860 (2018)
24. Tomiyama, T., Moyen, F.: Resilient architecture for cyber-physical production systems. *CIRP Annals* **67**(1), 161–164 (2018)
25. Ton, C.T., Mackunis, W.: Robust attitude tracking control of a quadrotor helicopter in the presence of uncertainty. In: 2012 IEEE 51st IEEE Conference on Decision and Control (CDC). pp. 937–942. IEEE (2012)
26. Venkatasubramanian, V., Rengaswamy, R., Yin, K., Kavuri, S.N.: A review of process fault detection and diagnosis: Part i: Quantitative model-based methods. *Computers & chemical engineering* **27**(3), 293–311 (2003)
27. Wang, Y., Ramirez-Jaime, A., Xu, F., Puig, V.: Nonlinear model predictive control with constraint satisfactions for a quadcopter. In: *Journal of Physics: Conference Series*. vol. 783, p. 012025. IOP Publishing (2017)

INWEND: Using CBR to automate legal assessment in the context of the EU General Data Protection Regulation

Clarissa Dietrich¹, Sebastian Schriml², Ralph Bergmann^{1,3} , Benjamin Raue²

¹ Business Information Systems II, University of Trier, 54286 Trier, Germany
<http://www.wi2.uni-trier.de>

² Institut für Recht und Digitalisierung Trier, University of Trier, 54296 Trier, Germany
<https://irdt.uni-trier.de/>

{[dietrich](mailto:dietrich@uni-trier.de),[schriml](mailto:schriml@uni-trier.de),[bergmann](mailto:bergmann@uni-trier.de),[raue](mailto:raue@uni-trier.de)}@uni-trier.de
³ German Research Center for Artificial Intelligence (DFKI), Branch University of Trier, Behringstraße 21, 54296 Trier, Germany
ralph.bergmann@dfki.de

Abstract The European General Data Protection Regulation (GDPR), which governs the processing of personal data in all EU Member States, contains an exemption for “purely personal or household activities”. Whether this so-called household exemption covers the setup of an online communication forum, particularly on a social media or chat platform, is a question of the individual case. We present a case-based reasoning approach to automatically assessing a scenario provided by the user and generating a tailored legal recommendation.

Keywords: Knowledge Engineering · Legal Tech · Case-Based Reasoning · General Data Protection Regulation

1 Introduction

The European General Data Protection Regulation (GDPR)⁴ entered into force on 25 May 2018, establishing a new data protection law in all EU Member States. Despite a two-year transition period, the advent of this novel legal framework has caused considerable legal uncertainty: Companies, associations and other institutions have sought legal advice in order to ensure compliance with the regulation and to avoid its potentially harsh sanctions: Article 83(5) GDPR allows for administrative fines up to EUR 20 million, or in the case of an undertaking,

Copyright © 2020 by the paper’s authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

⁴ Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC, OJ L 119/1.

up to 4 per cent of the total worldwide annual turnover of the preceding financial year, whichever is higher.

The material scope of application of the GDPR, however, is not restricted to enterprises and organizations: Virtually every individual handles personal data of others incessantly and can therefore be subject to data protection law. Seemingly innocuous examples of data processing include private blogging, the use of social media accounts, taking and sharing pictures of others and a variety of similarly common activities.

Legal uncertainty entails negative consequences beyond mere lack of compliance, specifically so-called “chilling effects”⁵: Individuals may be deterred from socially desirable activities even though these could have been implemented in conformity with data protection laws. The need for high-quality, low-threshold legal expertise is therefore a worthy area of application for artificially intelligent systems.

In the interdisciplinary INWEND⁶ project at the University of Trier, we are developing a software prototype that can elicit relevant factual information from non-lawyers and automatically generate a (preliminary) legal assessment using case-based reasoning (CBR) techniques. The focus of this project is the so-called household exemption in Article 2(2)(c) GDPR which excludes from the scope of the regulation the processing of personal data “by a natural person in the course of a purely personal or household activity”.

This project has a pilot character for the research at the intersection of artificial intelligence and European law. In an interdisciplinary cooperation between legal experts and computer scientists the understanding of the respective other discipline is expected to be improved, which should also enable the expansion of research activities to other common interdisciplinary fields of research. The scientific relevance of the development of a structured legal case base which facilitates automated legal reasoning lies in the possibility to work with consistent and structured case data in the future in order to develop artificially intelligent systems.

The remainder of the paper is organized as follows. In section 2 we outline related work on the use of CBR systems in a legal context as well as recent but distinct research in legal information technology. In section 3 we provide an overview of the relevant legal domain and the scope of the prototype (in the interest of brevity called INWEND hereinafter). In section 4 we proceed to describe our approach to modelling this knowledge domain by using an initial case situation and approximately 200 case variations generated using a set of

⁵ The term is defined by the Oxford Dictionary as “a discouraging or deterring effect on the behaviour of an individual or group, especially the inhibition of the exercise of a constitutional right, such as freedom of speech, through fear of legal action.”, https://www.lexico.com/definition/chilling_effect.

⁶ INWEND stands for “Intelligente Wissensbasierte Entscheidungsunterstützung für juristische Fragestellungen am Beispiel des Datenschutzes”, which translates to “Intelligent knowledge-based decision support for legal questions on the example of the data protection law”.

parameters. We conclude with some remarks on the feasibility of our approach in civil law jurisdictions.

2 Foundations and Related Work

The automation of legal tasks in the broadest sense is currently a topic of intense economic and scientific interest, with a variety of projects and products being discussed under the broad umbrella term of “Legal Tech” [6,8]. While the upsurge in public attention afforded to this field is a more recent phenomenon, the roots of legal informatics are much older: Many influential works date back to the second half of the twentieth century and predate the development and accessibility of powerful computers, large data storage capacities and the Internet.

The general suitability of CBR techniques for handling legal tasks was discovered early, partly because the CBR approach exhibits structural similarities to legal reasoning that are particularly apparent in the common law with its doctrine of precedent and the principle of *stare decisis*⁷. We will first outline some pioneering research work conducted in the 1980s and 1990s and then contrast the INWEND project to this and other contemporary research with a different outlook.

2.1 CBR in the Legal Domain

The possibility to associate the CBR approach with legal reasoning and argumentation was discovered in as early as the 1980s [17]. The pioneering legal reasoning system HYPO [16] used CBR techniques to generate arguments by comparing and contrasting cases with the use of a case base [3]. HYPO then inspired a multitude of further approaches and systems: The successor system CABARET [18] (which stands for “Case-Based Reasoning Tool”) integrated CBR with a rule-based approach. The system CATO [2] (“Case Argument Tutorial”) was designed to teach argumentation skills to students, whereas IBP [7] (“Issue-Based Prediction”) was aimed at the prediction of case outcomes. While these approaches were all rooted in the common law legal system, the INWEND project studies, inter alia, the applicability of these approaches and findings to German and European law.

2.2 Other Related Work

More recently, the ARGUMENTUM [11] project at the German Research Center for Artificial Intelligence has worked on argument extraction from German Federal Constitutional Court decisions. The relevant text passages are extracted from a corpus of decisions to justify or refute assertions in order to construct a convincing argument. While in the German legal system statutory law is regarded as a main source of reasonable legal arguments, decisions makers are to a large extent guided by the reasoning brought forward in court decisions.

⁷ Latin for: “Let the decision stand.”

The INWEND project takes a different approach, setting aside argument mining and starting from highly structured legal case information provided by domain experts.

3 Legal Background and Project Scope

We begin our account with a brief introduction to the legal domain of our projected CBR system, in order to both provide some context for our development strategy and highlight the practical need for the solution we propose.

3.1 Significance of the Household Exemption in Art. 2(2)(c) GDPR

According to Article 2(1) of the GDPR, the regulation applies, inter alia, to the “processing of personal data wholly or partly by automated means”, comprising practically any use of data processing systems [14]⁸. The broad material scope of application of the GDPR is counterbalanced by exemptions listed in Article 2(2) GDPR, in particular the so-called household exemption in Article 2(2)(c) GDPR.

This clause excludes from the scope of application the processing of personal data “by a natural person in the course of a purely personal or household activity”. This means that any such activity may be carried out without having to observe any of the obligations under the GDPR. These comprise a variety of legal duties, such as the principles relating to the processing of personal data set forth in Article 5, the need to justify the processing according to Article 6 and to obey the rights of the data subject set forth in Articles 12 et seqq. Being an exception to the entire data protection regime, there is general agreement that the clause must be interpreted restrictively [12]⁹.

3.2 Scope of the INWEND System

A general problem of legal interpretation, including the construction of statutory law, is the open-textured nature of legal concepts [1] leading to ambiguity. For instance, under which circumstances does the household exemption cover (private, not business-related) communication in a chat or online forum? As the exchange of information via technical communication devices generally entails the processing of personal data, the applicability of the GDPR hinges upon the question whether such communication is to be deemed a “purely personal or household” activity.

The INWEND system is designed to address the need for an automated legal assessment of the question: Does a natural person opening a communication forum on an online platform comply with the household exemption? The practical implication of this system is that a user may enter the circumstances of their

⁸ Paal/Pauly-*Ernst*, Art. 2 GDPR margin number 5.

⁹ Kühling/Buchner-*Kühling/Raab*, Art. 2 GDPR margin number 21.

(intended) use of such a forum and be immediately provided with (preliminary) legal feedback. The availability of such a tool may, as a secondary effect, incentivise platform providers to make available privacy-friendly options to attract more users. In the following section we will describe our strategy for modelling this legal problem and the first results of our research.

4 Approach to Modelling the Household Exemption with CBR

In case-based reasoning the task of solving problems is based on previous experience, which is stored in the form of cases in a case base [4]. This method of experience storage must be reconciled with the experience knowledge of the household exemption. A legal question should be represented in a way a CBR system can interpret as a case. We select a structural approach, thus a case is represented in attribute-value form using a structured vocabulary [4]. This chapter describes how we determine suitable attributes and create a functional case base in order to appropriately represent legal thinking for CBR.

4.1 Case Base Construction from an Initial Case Situation

The CBR approach relies on the existence of a case base, from which relevant reference experiences can be retrieved. With the GDPR being a novel legal framework, the availability of pertinent court decisions – particularly by the authoritative European Court of Justice (ECJ) – is low. Thus, there is little documented experience available that could be turned into a case base.

We therefore took a different approach in acquiring knowledge from the knowledge domain: Based on an interpretation of the household exemption, we developed an exemplary initial case situation. This is a short factual description of the scenario in which the question concerning the applicability of the household exemption arises. In particular, we consider the situation in which a user wants to open a forum on a communication platform and invite others to join the discussion.

This allowed us to identify a number of parameters which pertain to the applicability of the household exemption: Who is the intended circle of participants (such as family members, close friends, acquaintances), and how many people will be accessing the forum? How much personal information of other participants in the discussion will the user gain? Are there technical asymmetries by which the user will have more insight into other participants' personal data than vice versa?

Since legal ontologies are commonly used in legal informatics as a formal knowledge model [13], we considered using an existing legal ontology as a means of knowledge representation. Existing ontologies include LKIF Legal Ontology [9], Criminal Law Ontology [19] and PrOnto [15]. While the LKIF Legal Ontology and the Criminal Law Ontology are not focussed on the EU General Data Protection Regulation, PrOnto models the GDPR main concepts but does not

provide a sufficiently sophisticated representation of the household exemption. As none of these ontologies seemed apt to model our situation adequately, we decided to define a custom set of parameters in order to represent cases with a basic ontology.

4.2 Impact of Legal Reasoning on Case Base Structure

With the help of domain experts we created a documentation of the selected parameters and their features, providing strategies for the assessment of problematic cases as well as sets of positive and negative examples. In the next step, we used these parameters to synthetically generate a case base of approximately 200 cases. Since not all combinations of the parameter features were meaningful from a legal standpoint, the case base was slightly smaller than the product of all features: With five parameters, each of which ranging between two and five features, a total of 240 combinations would have resulted; since 48 of these combinations involved irreconcilable case details, there were only 192 legally meaningful cases.

We then assessed the applicability of the household exemption in the cases of the case base. It is worth noting that, rather than judging cases on an individual basis, legal domain experts employ two strategies to determine the outcome of cases by groups: Firstly, they try to identify so-called “edge cases”, these being parts of the case base where the legal assessment reaches a “tipping point” between two contrary results. Secondly, they extensively employ reasoning *a fortiori*¹⁰, arguing for instance that a case cannot qualify for the household exemption where a comparable case with a more restricted circle of participants has already been rejected: Since a wider circle of participants is an argument against the applicability of the household exemption, such a decision would be inconsistent.

The consistency of a legal case base is a notion which was discussed earlier in the context of the common law, specifically with regard to the doctrine of precedent [10]: In a legal system where court decisions can have a binding effect, it is of evident importance that courts are able to determine whether their envisaged decision is commensurate with the already existing legal framework. With our approach being rooted in a different jurisdiction, it is an important question of our research how this notion of consistency is to be construed in civil law and attained in our system. This is true in both a conceptual sense, which refers to the structure of the case base, and in a pragmatic sense, which is concerned with actual decision making. It is therefore yet to be explored how a consistent case base can be generated from expert knowledge in an expedient and robust way.

4.3 Implementation of the Model in ProCAKE

Drawing on the aforementioned domain knowledge, we then implemented the case representation using the CBR system ProCAKE [5]. This is a framework

¹⁰ Latin for: “from the stronger”.

for structural and process-oriented CBR applications developed by the Department of Business Information Systems II at the University of Trier. The case representation comprises the parameters as well as the range of acceptable values corresponding to the contents of the case base. Additionally, the user is provided with an option to make no specification in order to allow for factual uncertainty.

Considering the argumentation pattern mentioned in 4.2, whereby the applicability of the household exemption can be determined by *a fortiori* reasoning, we chose to model the parameter *circle of participants* as a polyvalent attribute. Cases that differ merely with regard to this parameter can be summarized as one case by representing the values of the respective cases for this parameter in a set. This step reduces the case base by 30 cases. At the time of writing, further modifications of the model are still in progress.

Based on the case model and the case base, a first similarity model was created. For calculating the global similarity the individual parameters are weighted by their impact on the assessment of the case. For example, the aforementioned parameter *circle of participants* has a particularly strong bearing on the outcome of the assessment.

The similarity for the parameter *circle of participants* is computed with a set mapping. If the value of the query matches a value in the set of a case of the case base, the similarity is high, otherwise low. The other local similarities are implemented with simple similarity measures, with equal values being assigned a high similarity score and different values being assigned a low similarity score. Again, at the time of writing, further modifications of the model are in progress.

4.4 Realization of a Graphical User Interface

We developed a prototypical Graphical User Interface (an excerpt is shown in Figure 1), to visualize the intended interaction with the program. The user has to respond to questions by choosing the answer most fitting to the situation at hand. As mentioned under 4.3, the questions also include the option to abstain from answering any individual question.

Each question corresponds to an attribute in our case model, with the answers determining their possible values. As can be seen in Figure 1, we guide the user through the interaction by providing additional information on questions and answers as needed. For additional reference, the factual situation described by the selected parameters is summarized for the user, which can be seen in Figure 2. Specifically, the text seen underneath the headline “Sachverhalt” (facts of the case) is a summary of the factual situation automatically generated from the answers given by the user; the text underneath the headline “Einschätzung” (legal opinion) is an easily comprehensible assessment of this factual situation, pertaining to the plain result – whether the household exemption is applicable or not – as well as its legal certainty.

On the basis of the answers selected by the user a query case is created. When the user clicks the submit button, this query is transmitted to the back end as a JSON object and then relayed as a query to a ProCake server, where it



Fig. 1. Excerpt of the GUI

is used for a retrieval on the case base. The retrieval result containing the most similar case is then returned to the web interface. The outcome of this retrieved case is displayed on the GUI (see Figure 2) and suggested to the user as the closest match to their query.



Fig. 2. Exemplary assessment of a submitted case.

In summary, this first version of a GUI generates a legal assessment in the context of the household exemption for a user case, based on the presented parameters, case base and similarity model.

5 Conclusion and Future Work

This paper presents a first CBR approach to modelling the legal assessment of a factual scenario in the context of the GDPR household exemption. The prototype system developed in the INWEND project is designed to elicit relevant facts from the user and to automatically generate a tailored legal recommendation. In the current stage of development, we have generated a substantial case base to reflect the structure of the GDPR household exemption. The system is controlled via a web interface and is able to retrieve reference cases from its case base.

In a next step, we plan to refine our ontology which represents our concept of the case representation. The ontology serves as a translation help between users and domain experts.

We judge the scalability of this approach to a wider field of law as rather difficult, because the expert knowledge required is expensive and the acquisition effort is high. Thus, a further interesting task to work on is to generate new cases semi-automatically: An algorithm suggests a scenario based on the case parameters and an expert decides the applicability of the household exemption. This algorithm should determine the potential edge cases and disregard cases that do not improve the ability to decide a case.

Further work needs to be undertaken to identify the most relevant cases in the current case base and to adapt the domain and similarity model. We would also like to evaluate our approach with test users in order to verify its reliability and practical utility.

We conclude that CBR systems are generally capable of providing initial legal assessments for non-lawyers in civil and EU law jurisdictions. We could imagine combining our approach with text mining strategies to extract case features from a plain text entered by the user instead of generating the query from a user's questions and answers.

Acknowledgements. This work is funded by the German Federal Ministry of Education and Research (BMBF) under grant number 01UG1920.

References

1. Aikenhead, M.: The uses and abuses of neural networks in law. *Santa Clara High Technology Law Journal* **12**, 31–70 (1996)
2. Aleven, V.: Teaching Case-Based Argumentation through a Model and Examples. Ph.D. thesis, University of Pittsburgh (1997)
3. Ashley, K.D.: Reasoning with cases and hypotheticals in hypo. *International Journal of Man-Machine Studies* **34**(6), 753 – 796 (1991)
4. Bergmann, R.: Experience Management: Foundations, Development Methodology, and Internet-Based Applications, LNCS, vol. 2432. Springer (2002)
5. Bergmann, R., Grumbach, L., Malburg, L., Zeyen, C.: Procake: A process-oriented case-based reasoning framework. In: Proceedings of International Conference on Case-Based Reasoning (ICCBR) (2019)

6. Breidenbach, S., Glatz, F.: Rechtshandbuch Legal Tech. Beck C. H., München (2020)
7. Brüninghaus, S., Ashley, K.D.: Predicting outcomes of case based legal arguments. In: Proceedings of the 9th International Conference on Artificial Intelligence and Law. p. 233–242. ICAIL '03 (2003)
8. Hartung, M., Bues, M.M., Halbleib, G.: Legal Tech - A Practitioner's Guide. Bloomsbury Academic, London (2018)
9. Hoekstra, R., Breuker, J., Di Bello, M., Boer, A.: The lkif core ontology of basic legal concepts. In: Proceedings of the 2nd Workshop on Legal Ontologies and Artificial Intelligence Techniques. CEUR Workshop Proceedings, vol. 321, pp. 43–63 (2007)
10. Horty, J., Bench-Capon, T.: A factor-based definition of precedential constraint. *Artificial Intelligence and Law* **20**, 181–214 (2012)
11. Houy, C., Niesen, T., Calvillo, J., Fettke, P., Loos, P., Krämer, A., Schmidt, K., Herberger, M., Speiser, I., Gass, A., Schneider, L., Philippi, T.: Konzeption und Implementierung eines Werkzeuges zur automatisierten Identifikation und Analyse von Argumentationsstrukturen anhand der Entscheidungen des Bundesverfassungsgerichts im Digital-Humanities-Projekt ARGUMENTUM. *Datenbank-Spektrum* **15**(1), 15–23 (2015)
12. Kühling, J., Buchner, B.: Datenschutz-Grundverordnung. No. 2, C.H.Beck (2018)
13. Liebwald, D.: Auf dem Weg zum Begriff: Vom Rechtswort zur Rechtsontologie – Automatisierte Verfahren zur semantischen Erschließung von Texten. *Wort - Bild - Zeichen: Beiträge zur Semiotik im Recht* **13**, 203–223 (2012)
14. Paal, B., Pauly, D.: Datenschutz-Grundverordnung. No. 2, C.H.Beck (2018)
15. Palmirani, M., Martoni, M., Rossi, A., Bartolini, C., Robaldo, L.: Legal ontology for modelling gdpr concepts and norms. In: Proceedings of the 31st International Conference on Legal Knowledge and Information Systems (JURIX). pp. 91–100 (2018)
16. Rissland, E., Ashley, K.: Hypo: A precedent-based legal reasoner. *Advanced Topics in Law and Information Technology* pp. 213–234 (1989)
17. Rissland, E., Ashley, K., Branting, L.: Case-based reasoning and law. *Knowledge Engineering Review* **20**, 293–298 (2005)
18. Skalak, D., Rissland, E.: Arguments and cases: An inevitable intertwining. *Artificial Intelligence and Law* **1**, 3–44 (1992)
19. Soh, C., Lim, S., Hong, K., Rhim, Y.Y.: Ontology modeling for criminal law. In: Pagallo, U., Palmirani, M., Casanovas, P., Sartor, G., Villata, S. (eds.) *AI Approaches to the Complexity of Legal Systems*. pp. 365–379. Springer (2018)

FGWI-BIA Workshop

Ein agiles Vorgehensmodell zur Einführung von Predictive Analytics in Unternehmen

Jule Aßmann, Joachim Sauer und Michael Schulz

NORDAKADEMIE Hochschule der Wirtschaft, Elmshorn, Deutschland
assmann.jule@guj.de,
{joachim.sauer, michael.schulz}@nordakademie.de

Abstract. Predictive Analytics dient dazu, unter Zuhilfenahme von statistischen Verfahren und auf Basis historischer und aktueller Daten Vorhersagen zu zukünftigen Ereignissen zu treffen. Anwendungsfälle für eine solche Form der Analyse sind in einigen Branchen bzw. Unternehmen zurzeit noch schwer zu finden. In diesem Artikel wird ein agiles Modell vorgestellt, das die Einführung von Predictive Analytics unterstützt und dabei die Unsicherheit minimiert. Damit wird eine Lücke zu etablierten Vorgehensmodellen geschlossen, die erst ansetzen, wenn die Entscheidung, eine Datenanalyse auf einer spezifischen Problemstellung durchzuführen, bereits getroffen wurde.

Der Entstehung des Modells liegen Erfahrungen zugrunde, die bei der Gruner + Jahr GmbH in einem Geschäftsbereich gesammelt wurden, der dem Bereich des Digital Marketings & Advertisings zuzuordnen ist. Eine Übertragbarkeit auf andere Branchen wird berücksichtigt.

Keywords: *Predictive Analytics, Anwendungsfälle, Digital Marketing & Advertising*

1 Einleitung

Predictive Analytics (PA) bietet die Möglichkeit, historische und aktuelle Daten zur Vorhersage zukünftiger Ereignisse zu nutzen. Obwohl PA bereits seit vielen Jahren als relevante Disziplin für Unternehmen allgemein akzeptiert ist, ist dies in einigen domänenspezifischen Anwendungen nicht immer der Fall. Im Gartner Hype Cycle für das Themengebiet „Business Intelligence & Analytics“ ist PA seit 2011 vertreten und hat den Gipfel der überzogenen Erwartungen bereits passiert [1]. Im Hype Cycle für das Themengebiet „Digital Marketing & Advertising“ (DMA) wird PA dagegen erst seit 2015 geführt und aktuell an der Spitze der Aufmerksamkeit eingeordnet; die Phasen der Erleuchtung und anschließender Produktivität haben noch nicht stattgefunden [2]. Die Entscheidung für den Einsatz neuer Technologien wie PA erfordert von Unternehmen Innovationsbereitschaft. Weniger risikoaffine Firmen suchen daher vor einer Durchführungsentscheidung häufig nach Use Cases (im Deutschen: Anwendungsfälle) mit einem erkennbaren wirtschaftlichen Nutzen (vgl. z. B. [3]). Diese sind unter anderem im Bereich des DMA jedoch noch nicht ohne größeren Aufwand zu identifizieren.

In dieser Arbeit wird ein Vorgehensmodell vorgestellt, das Firmen bei der Einführung von PA unterstützt und die Unsicherheit in dieser Anfangsphase minimiert. Dem Modell liegen dabei Erfahrungen zugrunde, die in einem Geschäftsbereich der Gruner + Jahr GmbH, der dem DMA-Bereich zuzuordnen ist und zu Beginn der Untersuchung noch keine Anwendung für PA besaß, gesammelt wurden. Damit besteht ein spezieller Branchenfokus; eine Übertragbarkeit des erarbeiteten Modells auf weitere Branchen und Unternehmen, die den Einsatz von PA prüfen, wurde jedoch berücksichtigt.

Es wurden möglichst viele Verfahren gewählt, um für den Einsatz von PA relevante Einsatzszenarien im betrachteten Unternehmen zu identifizieren: Nach einer Betrachtung theoretischer Grundlagen wird aufbauend auf einer strukturierten Literaturanalyse eine empirische Untersuchung in Form qualitativer Befragungen mit ausgewählten Personen aus der Vermarktung, sowohl innerhalb der Gruner + Jahr GmbH, als auch mit unternehmensfremden Vermarktungsakteuren, beschrieben. Diese Interviews zielten darauf ab, (Teil-) Automatisierungspotenziale und Ineffizienzen in den Tätigkeiten der Befragten zu identifizieren. Basierend auf den erzielten Ergebnissen konnten Anwendungsfälle für PA-Vorhaben erarbeitet werden, die im Kontext der Vermarktungsorganisation Einsatz finden können. Für diese Use Cases wurden Faktoren erhoben, die Anhaltspunkte zum Aufwand einer Implementierung darstellen. Ergänzt wurde die empirische Untersuchung durch eine Gruppendiskussion mit den Führungskräften aus der Gruner + Jahr GmbH, die darauf abzielte, ein im Sinne des Unternehmens übergeordnetes Verständnis zu den aus den qualitativen Befragungen hervorgehenden Problemen zu gewinnen, sowie die Relevanz der Use Cases einzuordnen und ihren Nutzen zu bewerten.

Aus den gewonnenen Erkenntnissen wurde anschließend ein generelles Vorgehensmodell für die Einführung von PA in Unternehmen abgeleitet, das in Abschnitt 4 vorgestellt und abstrakt beschrieben wird. Danach werden Erfahrungen bei der Umsetzung dargestellt und diskutiert. Die Arbeit schließt mit einer Bewertung der Ergebnisse und einem Ausblick ab.

2 Theoretische Grundlagen

In diesem Abschnitt werden Grundlagen von Predictive Analytics und etablierte Vorgehensmodelle zu deren Anwendung erläutert.

2.1 Predictive Analytics

Die Aufbereitung und Verwendung von Daten zur Entscheidungsunterstützung ist bis in die 1970er Jahre zurückzuführen [4]. Durch immer weiter zunehmende Datenmengen und die Verbesserung der zugrundeliegenden Technologien hat die Datenanalyse im letzten Jahrzehnt einen deutlich höheren Stellenwert erhalten als dies zuvor der Fall war [5]. In der Folge entstanden bzw. verbreiteten sich zahlreiche Methoden und Disziplinen, zu denen auch Predictive Analytics zählt.

Der PA-Begriff ist nicht einheitlich definiert. Da diese Herausforderung jedoch nicht den Fokus der vorliegenden Arbeit darstellt, wird an dieser Stelle auf die Literatur verwiesen (vgl. dazu z. B. [5–11]). In Anlehnung an bestehende Definitionen soll in der vorliegenden Arbeit unter Predictive Analytics eine Form der Datenanalyse verstanden werden, mit der unter Zuhilfenahme statistischer Verfahren auf Basis historischer und aktueller Daten Vorhersagen zu zukünftigen Ereignissen getroffen werden können.

PA basiert auf unterschiedlichen Methoden zur Aufdeckung von Ursache-Wirkungs-Beziehungen, deren Ursprung im Data Mining zu verorten ist [11]. In Abbildung 1 sind Methodengruppen und deren Eigenschaften dargestellt, in die ein Großteil der verwendeten Analyseverfahren eingeordnet werden können. Charakterisierende Merkmale einzelner Methoden, wie beispielsweise Einfachheit und Nachvollziehbarkeit, werden in der Matrix bewusst nicht als Kriterien erfasst. Die Ursache hierfür liegt in der Vielfalt statistischer Methoden, die je Gruppierung angewendet werden können und eine übergreifende Bewertung unmöglich macht.

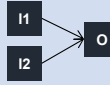
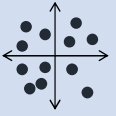
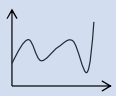

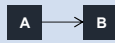
Eigenschaft	Prognose	Segmentierung	Zeitreihenanalyse	Klassifikation	Assoziation
Visualisierung					
Art	überwacht	unüberwacht	überwacht	überwacht	unüberwacht
Anwendungsargument	Vorhersage Output-Variable über Ursache-Wirkungs-Beziehung	Unterteilung Objekte notwendig, aber Klassen nicht bekannt	Zeitabhängigkeit der Daten (z.B. saisonale Schwankung)	Zuordnung Objekt zu festen Klassen erforderlich	(Unbekannte) Wenn-Dann-Beziehungen identifizieren
Schwächen	Vielzahl Input-Variablen → Kenntnis erforderlich	Einfluss von Ausreißern	Nur bei Zeitabhängigkeiten nutzbar	Klassifikationsfehler (falsch positiv oder falsch negativ)	Offensichtliche Assoziationen
Beispiel der Anwendung (marketing-spezifisch)	Vorhersage von Angebotsannahme / -ablehnung	Marktanalysen Kundenclustering	Umsatzforecasts	Identifikation Kunden / Nicht-Kunden	Identifikation Cross-Selling-Potenziale

Abbildung 1. Predictive-Analytics-Methodengruppen, in Anlehnung an [8, 11, 12]

In der wissenschaftlichen Literatur existieren Veröffentlichungen zu PA unter anderem in den Bereichen Luftfahrt, Gesundheitswesen, Einzelhandel, bei der Produkteinführung und Preisgestaltung [5, 12–15]. Darüber hinaus sind Ansätze zur Nutzung für marketingtreibende Unternehmen vorhanden [6]. Einsatzpotenziale speziell für die Medienbranche / Vermarktung sind in der wissenschaftlichen Diskussion dagegen bisher kaum thematisiert. Die Herausforderungen durch Unsicherheiten in der Entscheidung für oder gegen den PA-Einsatz sind für Unternehmen aus diesem Bereich daher besonders hoch, weshalb sich der Bereich gut für eine beispielhafte Betrachtung eignet.

2.2 Vorgehensmodelle zur Anwendung von Predictive Analytics

Predictive Analytics basiert in großen Teilen auf Techniken des Data Minings [7], weshalb sich in diesen beiden Bereichen verwendete Vorgehensmodelle nicht unterscheiden. Genannt werden können hier, bezogen auf die weite Verbreitung, vor allem das Modell *Knowledge Discovery in Databases* (KDD) [16], der *Cross Industry Standard Process for Data Mining* (CRISP-DM) [17] und der Prozess des *Sampling, Exploring, Modifying* und *Assessing* (SEMMA) [18]. All diese Vorgehensmodelle sind bewusst einfach gestaltet, sodass sie nicht nur von PA-Spezialisten, sondern auch von Domänenexperten verstanden werden und so ein Bearbeiten von Problemstellungen über unterschiedliche Interessengruppen und verschiedene Phasen, wie etwa der Datenaufbereitung und -analyse, hinweg ermöglichen.

Im Gegensatz zu KDD und SEMMA beinhaltet das CRISP-DM (vgl. Abbildung 2) mit dem *Business Understanding* eine Phase, die dazu dient, das Ziel und die Anforderungen eines Vorhabens aus Geschäftsperspektive zu verstehen, in eine für die Analyse geeignete Fragestellung zu übersetzen und einen Projektplan zur Umsetzung abzuleiten [17]. Ein umfangreiches Geschäftsverständnis ist essenziell, um eine Datenanalyse erfolgreich durchführen zu können [19]. Ebenfalls wird im CRISP-DM die Agilität der Datenanalyse deutlicher hervorgehoben, als dies in den anderen beiden genannten Vorgehensmodellen der Fall ist.

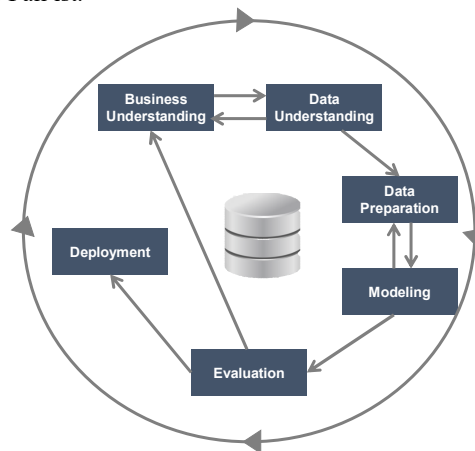


Abbildung 2. CRISP-DM Vorgehensmodell, aus [17]

Auch wenn alle genannten Vorgehensmodelle und vor allem das CRISP-DM viele relevante Aspekte eines PA-Projektes adressieren, setzen sie doch erst dort an, wo die Entscheidung für den Einsatz einer Datenanalyse auf eine spezifische Problemstellung bereits getroffen wurde. In dieser Arbeit soll ein Modell entwickelt werden, das früher, nämlich bei der Auswahl der am besten geeigneten Fragestellung für PA, ansetzt. Dabei sollen die wichtigsten Eigenschaften des CRISP-DM, wie Einfachheit, Agilität und die Möglichkeit des Einbezugs sämtlicher Interessengruppen, als Vorbild für die Entwicklung dienen. Dadurch wird auch eine kombinierte Anwendung dieser beiden Modelle möglich.

3 Forschungsdesign

Das Vorgehensmodell wurde iterativ bei der Gruner + Jahr GmbH entwickelt, da eine einfache Übertragbarkeit aus anderen Unternehmen nicht möglich ist und deshalb die Expertise der Mitarbeiterinnen und Mitarbeiter des spezifischen Unternehmens besonders berücksichtigt werden muss. Dabei wurden ein umfangreiches Literatur-Review, eine qualitative Befragung sowie eine abschließende Gruppendiskussion mit Experten genutzt (vgl. Abbildung 3). Dieses Vorgehen ermöglicht die umfangreiche Berücksichtigung verschiedener Perspektiven auf die Tauglichkeit von PA im spezifischen Fall und sichert durch die Einbeziehung von Interessengruppen zusätzlich die Akzeptanz von Entscheidungen.

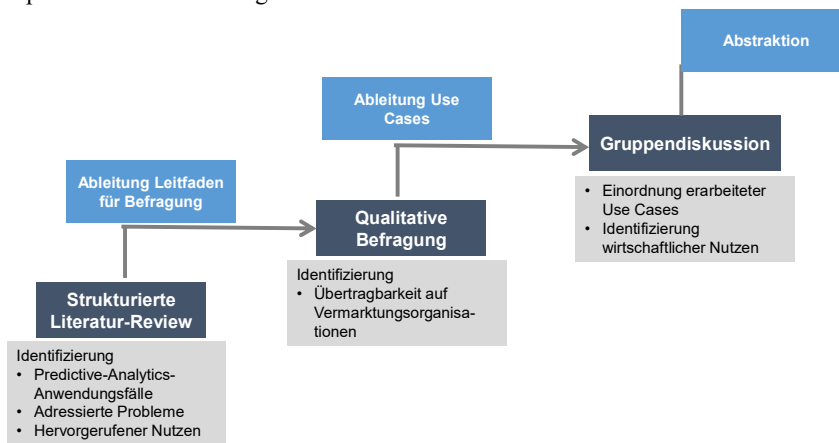


Abbildung 3. Forschungsvorgehen

Mit einem Literatur-Review wurden über die Datenbanken ScienceDirect, Beluga (Katalog der Hamburger Bibliothek), Springer-Link, EBSCO und ACM Digital Library Use Cases mithilfe der in Abbildung 4 dargestellten, aus jeweils drei Elementen bestehenden Suchbegriffe identifiziert.

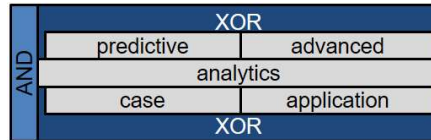


Abbildung 4. Muster der bei der Literaturrecherche verwendeten Suchbegriffe

Im ersten Prozessschritt konnten 30 Use Cases aus unterschiedlichen Branchen erkannt und auf ihren Ursprung und hervorgerufenen Nutzen durch eine PA-Anwendung hin untersucht werden. Eine komprimierte Übersicht der Use Cases befindet sich im Anhang dieses Artikels. Die identifizierten Effekte konnten zu den Kategorien Prozessoptimierung, Verkaufsunterstützung, Wissen, Sicherheit und Kundenerlebnis gruppiert werden, die verschiedene Tätigkeiten eines Jobprofils ansprechen können.

Die anschließende qualitative Befragung wurde unter sechs Vermarktungsmitarbeitern durchgeführt, welche basierend auf den zuvor gebildeten Kategorien den Anteil der manuellen Arbeit und des Bauchgefühls für ihren Aufgabenbereich bewerteten. Als Resultat konnten fünf potenzielle PA-Use Cases für die Vermarktung ermittelt werden: Die Erstellung von Umsatzprognosen für verschiedene Vermarktungsblickwinkel, die optimale Bestimmung von Konditionen, die Identifikation von (Neu-)Kundenpotenzialen, die Identifikation von Cross-Selling-Potenzialen sowie die Unterstützung einer gezielten Kundenansprache.

Eine abschließende Gruppendiskussion unter Führungskräften der Vermarktung ordnete die erarbeiteten Use Cases übergreifend ein. Mithilfe einer Nutzwertanalyse wurden Aufwandstreiber (Anzahl Datenquellen, Art und Unabhängigkeit der Daten, Reifegrad, notwendige Kombination analytisches und fachliches Know-how) und Nutzenindikatoren (Effizienzsteigerung, Kostenreduktion, Erlössteigerung, Wettbewerbsvorteil, Risikominimierung, Entscheidungsrelevanz, Zukunftsrelevanz und Investitionsbereitschaft) für die Use Cases gegenübergestellt und diese vergleichbar gemacht. Für die erste Umsetzung von PA in DMA-Unternehmen wurden daraus resultierend die Use Cases Umsatzprognosen, Cross-Selling-Potenziale und eine Anwendung zur optimalen Bestimmung von Konditionen empfohlen.

Im Anschluss erfolgte für das Unternehmen Gruner + Jahr GmbH die Umsetzung zweier PA-Use Cases. Konform der Empfehlung der vorausgehenden Erarbeitung wurden zum einen Umsatzforecasts für verschiedene Vermarktungsblickwinkel erstellt. Zum anderen wurde die Identifizierung von Neukundenpotenzialen angewandt, die in der Gesamtbewertung aus Aufwand und Nutzen zwar nicht das höchste Ranking erhielt, jedoch mit einer zentralen Datenquelle auskommen kann und daher für die schnelle Ergebniserzielung besonders geeignet erschien. Auf Basis der realen Datengrundlage des Unternehmens konnten innerhalb eines Entwicklungszyklus aus wenigen Tagen schnelle Ergebnisse erzielt werden, die eine Evaluation der Machbarkeit der Use Cases zuließen. Es zeigte sich, dass für beide Use Cases zwar Ergebnisse erzielt werden konnten, das Vorhandensein von Daten in ausreichender Menge jedoch nicht für alle betrachteten Vermarktungsblickwinkel gegeben war. Dies wird als ein allgemeines Hindernis von PA-Vorhaben gesehen [5].

Nach einer kritischen Prüfung der Ergebnisse wurde über den weiteren Umgang entschieden. Use Cases können verworfen, in einem weiteren Zyklus angepasst, oder erweitert oder sofort in den Betrieb überführt werden. Für das Unternehmen Gruner + Jahr GmbH wurde entschieden, die Use Cases zunächst anzupassen und zu erweitern. Dafür wurden zum einen eine weitere Datenquellen angebunden, welche die Fehlermetriken der Prognose verringern konnte, sowie zum anderen das Prognoseergebnis um zusätzliche Informationen verfeinert. Das Ziel bestand darin, die Akzeptanz der Predictive-Analytics-Lösung durch inhaltliche Verbesserungen für eine nachfolgende Überführung in den Betrieb zu gewährleisten. Diese Überführung ist als Folgeaktivität geplant, jedoch zu diesem Zeitpunkt noch ausstehend. Vor der Weiterentwicklung der Use Cases wurde diese Überführung gegen die Umsetzung weiterer PA-Use Cases in Hinblick auf den zu erzielenden Nutzen und Aufwand erneut bewertet und gegeneinander abgewogen.

4 Beschreibung des agilen Modells

Aus den Erfahrungen des Einsatzes im Unternehmen wurde ein allgemeines Vorgehensmodell zur Einführung von Predictive Analytics in Unternehmen abstrahiert, das in Abbildung 5 dargestellt ist. Es ist nicht nur iterativ-inkrementell, sondern auch agil, da aus Erfahrungen aus der Nutzung gelernt werden und das Modell flexibel angepasst werden kann.

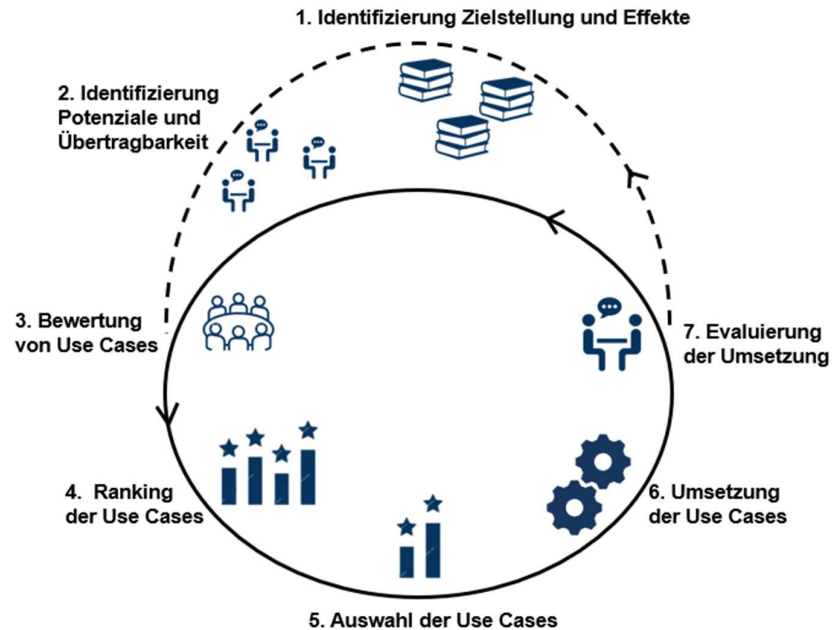


Abbildung 5. Vorgehensmodell zur Einführung von Predictive Analytics

Das Modell besteht aus sieben Schritten, die in zwei Zyklen angelegt sind. Anfangs werden die Schritte eins bis sieben durchlaufen. Abhängig von der Evaluierung im siebten Schritt wird der vollständige Zyklus erneut durchlaufen oder die vorhandenen Use Cases werden neu bewertet, was den Zyklus um zwei Schritte kürzt.

Schritt 1: Identifizierung Zielstellung und Effekte

Predictive Analytics sollte erst dann eingesetzt werden, wenn sinnvolle Einsatzpotenziale für das Unternehmen bzw. die Branche vorliegen. Um diese Beurteilung zu erreichen, sollten Verantwortliche zunächst die direkten Effekte und Zielstellungen von PA betrachten.

Schritt 2: Identifizierung Potenziale und Prüfung der Übertragbarkeit

Im zweiten Schritt sollte eine Prüfung der Übertragbarkeit der Zielstellungen auf das eigene Unternehmen bzw. die Branche hin erfolgen. Sofern es im Unternehmen Aufgaben und Prozessschritte gibt, welche die Effekte der PA optimieren können, sollten die Effekte mit Hilfe der folgenden Leitfragen konkretisiert werden. Das Potenzial des PA-Einsatzes kann dann, aufgeteilt auf zwei Dimensionen, ermittelt werden. Die erste Dimension betrifft die manuelle Arbeit: Ist die Aufgabe wiederkehrend, erfordert aber trotzdem viele manuelle Prozessschritte? Fehlen Informationen (= Wissen / Daten), welche die Prozesse beschleunigen könnten? Die zweite Dimension der Potenzial-

mittlung betrachtet das Bauchgefühl: Erfordern die Aufgaben viel Bauchgefühl, da Informationen (Wissen oder Daten) den Mitarbeitern nicht vorliegen? Ist die subjektive Einschätzung ein elementarer Bestandteil der Tätigkeit?

Die identifizierten Effekte können als Orientierungshilfe zur Ableitung von Handlungsempfehlungen in eine Matrix eingeordnet werden. Ein Beispiel, das bei Gruner + Jahr GmbH Anwendung findet und keinen Anspruch auf Übertragbarkeit besitzt, zeigt Abbildung 6.

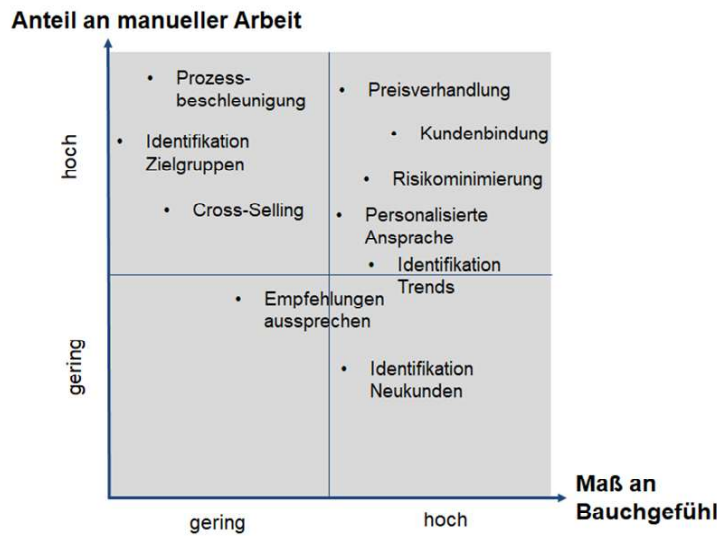


Abbildung 6. Einordnung Predictive-Analytics-Potenziale

In der Matrix spannen das Maß an Bauchgefühl und der Anteil an manueller Arbeit die Dimensionen auf. Wenn beides hoch ist, kann der Einsatz von PA besonders großen Nutzen schaffen (Beispiel Preisverhandlung). Für diesen Bereich sollten als erstes Use Cases erstellt werden. Wenn beides gering ist, kann auf PA verzichtet werden. Wenn nur der Anteil manueller Arbeit hoch ist, kann PA zur Effizienzsteigerung verwendet werden (Beispiel Prozessbeschleunigung); wenn nur das Bauchgefühl hoch ist, kann mit PA die Erfolgswahrscheinlichkeit erhöht werden (Beispiel Identifikation neuer Kunden). Die letzten beiden Bereiche sollten in Betracht gezogen werden, wenn Kapazität und Unternehmensfokus es zulassen.

Schritt 3: Bewertung von Use Cases

Die entstandenen Use Cases sind im Anschluss zu bewerten. Die Kriterien zur Bewertung von Nutzen und Aufwandstreibern sollten unternehmensindividuell identifiziert und gewichtet werden. Mögliche Nutzentreiber sind z. B. Erzielung von Effizienzsteigerung, Ermöglichung von Kostenreduktionen oder Minimierung von Risiken. Mögliche Aufwandstreiber sind z. B. die Anzahl Datenquellen, die Unabhängigkeit der

Daten (intern / extern) sowie der Reifegrad des Use Cases. Auch Chancen (neue Erlösmöglichkeiten, verbesserte Planungssicherheit, Steigerung der Profitabilität u. Ä.) und Risiken (Limitationen durch Datenschutz, Verfügbarkeit der notwendigen Fachkräfte, Auswirkung von Fehlern u. Ä.) sollten Kriterien zur Bewertung bilden.

Schritt 4: Ranking der Use Cases

Abschließend sind die entstandenen Use Cases auf Basis dieser Bewertung miteinander zu vergleichen. Dazu kann das Instrument der Nutzwertanalyse hilfreich sein, wobei die Bewertungskriterien in diesem Fall unternehmensindividuell gewichtet werden müssen. Je konkreter die Umsetzung und Rahmenbedingungen durch ein Unternehmen bekannt sind, desto eher kann zudem eine Quantifizierung von Aufwand und Nutzen erfolgen.

Schritt 5: Auswahl der Use Cases

Im fünften Schritt werden die Use Cases ausgewählt, die umgesetzt werden sollen. Da Abhängigkeiten zwischen den Use Cases bestehen können, müssen das nicht die am besten bewerteten Use Cases sein. Eine wichtige Abhängigkeit besteht beispielsweise im Personalbedarf zur Umsetzung.

Je nach Aufwand der einzelnen Use Cases muss auch geplant werden, wie viele Use Cases in diesem Zyklus umgesetzt werden sollen. Das kann insbesondere dann schwierig sein, wenn das Unternehmen noch nicht über ausreichende Erfahrungen mit der Implementierung von Use Cases verfügt.

Schritt 6: Umsetzung der Use Cases

Die ausgewählten Use Cases werden umgesetzt. Für Details sei hier auf etablierte Vorgehensmodelle wie das CRISP-DM verwiesen.

Schritt 7: Evaluierung der Use Cases

Im letzten Schritt werden die Use Cases unter Berücksichtigung der ursprünglich erstellten Bewertung kritisch evaluiert. Dabei sollten nicht nur alle Use Cases einzeln betrachtet werden; auch ein Vergleich verschiedener Use Cases ist notwendig, um daraus Schlüsse für das weitere Vorgehen zu ziehen.

Wenn die Umsetzung vom Unternehmen als erfolgreich angesehen wird und noch nicht alle identifizierten Potenziale genutzt wurden, erfolgt ein Übergang zu Schritt 3, bei dem die verbleibenden Use Cases im Licht der gesammelten Erfahrungen neu bewertet werden. Es ist auch möglich, schon (teilweise) implementierte Use Cases erneut aufzugreifen und zu verbessern.

Ansonsten beginnt die Bearbeitung wieder bei Schritt 1 mit der Identifizierung weiterer Zielstellungen und Effekte. Die Bearbeitung endet, wenn keine neuen Zielstellungen identifiziert werden können.

5 Erfahrungen mit der Umsetzung

Durch den Einsatz des Vorgehensmodells bei der Gruner + Jahr GmbH konnten fünf nutzenstiftende Use Cases identifiziert werden. Der mehrstufige Prozess und das Einbeziehen verschiedener Expertengruppen aus externer Literatur, internen Mitarbeitern und internen Führungskräften erhöhte dabei die Relevanz und Akzeptanz der entstandenen Use Cases deutlich und vermied so eine PA-Umsetzung, die hohen Aufwand bei geringem Nutzen bedeutet hätte. In Kombination mit der prototypischen Umsetzung wird diese Gefahr auf ein Minimum reduziert.

Die Anwendung des Modells im Unternehmenskontext zeigte, dass sich für Schritt 6 (Umsetzung der Use Cases) die Erstellung eines Prototyps anbietet, wenn die Unsicherheit hoch ist, ob die Daten ausreichend und in geeigneter Form vorhanden sind. In diesem Fall erlaubt eine prototypische Umsetzung auf Basis der Echt-Daten eine Einschätzung darüber. Im Anschluss kann eine finale Umsetzung, die höheren Aufwand bedeutet, mit einer Integration in das Business-Intelligence-System sowie einer regelmäßigen und automatischen Neu-Berechnung des Modells erfolgen.

Bereits bei der Umsetzung sowie Bewertung der Prototypen konnte die Relevanz einer Kombination aus analytischem und Domänenwissen deutlich erkannt werden, welche als eine der größten Herausforderungen von PA-Vorhaben gilt [20]. Ebenso zeigte sich, dass typische PA-Probleme, wie zu geringe Datenmengen oder siloartige Daten, die bisher keine Verknüpfung zulassen [5, 20, 21], frühzeitig identifiziert werden konnten.

Des Weiteren führt die praktische Umsetzung des Modells bei der Gruner + Jahr GmbH zu der Empfehlung, die verwendeten Datenquellen für den PA-Prototypen iterativ zu steigern, um schneller Zwischenergebnisse zu generieren. Nach Einbeziehen erster Datenquellen sollte ein weiterer Zyklus durchlaufen und eine Erweiterung um zusätzliche Datenquellen gegen die Umsetzung anderer Use Cases erwogen werden. In jedem Zyklus werden die Zwischenergebnisse zusammen mit Fachexperten ausgewertet, um ganz im agilen Sinne direktes Feedback zu sammeln und aus den Erfahrungen zu lernen.

Ein weiterer Vorteil, den die praktischen Erfahrungen aufdecken konnten, sind die verschiedenen Berührungspunkte, welche die relevanten Führungskräfte vor einer Durchführungsentscheidung mit PA haben. Die für die Unternehmensvertreter neuartige Thematik wurde so greifbarer – Unsicherheiten konnten reduziert und die Investitionsbereitschaft erhöht werden.

Gleichzeitig sollte ein besonderer Fokus darauf gelegt werden, eine Überführung in den Betrieb zeitnah zu realisieren – ebenfalls den agilen Prinzipien entsprechend. Andernfalls besteht die Gefahr vieler gute Prototypen ohne eines nutzbaren PA-Produktes. Für die Überführung in den Betrieb muss insbesondere dem Prozess und Change Management eine besondere Bedeutung zugeschrieben werden, da die praktische Umsetzung zeigt, dass einer PA-Lösung, die mit Datenwissen ein Bauchgefühl ersetzen oder ergänzen soll, Skepsis entgegengebracht wird.

6 Fazit

Die erste Erfahrung mit dem praktischen Einsatz des Modells zeigt, dass es geeignet ist, um mit den Herausforderungen eines Prozesses zur Einführung von Predictive Analytics in geeigneter Form umzugehen. Die Lücke etablierter Vorgehensmodelle wie KDD, SEMMA und CRISP-DM, die erst ansetzen, wenn die Entscheidung einer Datenanalyse auf eine spezifische Problemstellung bereits getroffen wurde, kann somit geschlossen werden. Zukünftig sollten weitere Umsetzungen auch in anderen Unternehmen bzw. mit anderen Analyseproblemstellungen erfolgen, um das Modell zu überprüfen und weiterzuentwickeln.

Das Modell gibt keine Aussage darüber, ob bzw. nach wie vielen Jahren der Nutzen der ermittelten PA-Use Cases für DMA-Unternehmen den Aufwand übersteigen wird. Diese Aussage ist nur unternehmensindividuell zu treffen, da organisatorische und technische Gegebenheiten Basisfaktoren für den Aufwand darstellen.

Für zukünftige Forschungen bietet es sich daher an, auf Basis einer Implementierung den tatsächlichen Nutzen messbar zu machen sowie einen Vergleich zum erwarteten Nutzen herzustellen. Darauf aufbauend kann schließlich eine valide Aussage zur Wirtschaftlichkeit getroffen werden, da der Aufwand konkret beziffert werden kann. Auch können Risikofaktoren, wie beispielsweise die Akzeptanz unter potenziellen Anwendern, differenzierter bewertet werden.

Das Vorgehen zur Einführung von PA, das in dieser Arbeit praktisch für die Gruner + Jahr GmbH angewendet wurde, enthält ein Abstraktionsniveau, das es zulässt, dieses Vorgehen auch auf andere Branchenbetrachtungen zu übertragen. Ebenso kann das Vorgehen als Orientierungshilfe für die Prüfung der Einsatzpotenziale weiterer Disziplinen neben Predictive Analytics dienen. Ein denkbare Szenario ist die Prüfung der Einsetzbarkeit von Künstlicher Intelligenz, da dies aktuell für immer mehr Unternehmen relevant wird.

Insbesondere für DMA-Unternehmen wird PA voraussichtlich in Zukunft ein entscheidender Baustein werden, um langfristig wettbewerbsfähig zu bleiben. Die gezielte Anwendung eines Vorgehensmodells zur erfolgreichen Einführung von PA wird daher in Zukunft essenziell sein, um die relevanten Use Cases zu identifizieren und zu bewerten sowie die Entscheidung für aber auch gegen den Einsatz von PA zu unterstützen. Nach erfolgreicher Durchführung erster Analysevorhaben muss das Vorgehensmodell allerdings auch um eine projektübergreifende, strategische Struktur ergänzt werden, die eine dauerhafte Etablierung der Disziplin in die Organisation ermöglicht.

References

1. Gartner, I.: Hype Cycle for Business Intelligence, 2011, <https://www.gartner.com/doc/1766215/hype-cycle-business-intelligence->
2. Gartner, I.: Hype Cycle for Digital Marketing and Advertising, 2018, <https://www.gartner.com/doc/3884103/hype-cycle-digital-marketing-advertising>
3. Next Generation Predictive Analytics. Using Forward-Looking Insights to Gain Competitive Advantage (2015)

4. Gluchowski, P.: Business Analytics – Grundlagen, Methoden und Einsatzpotenziale. HMD 53, 273–286 (2016)
5. Attaran, M., Attaran, S.: Opportunities and Challenges of Implementing Predictive Analytics for Competitive Advantage. *International Journal of Business Intelligence Research* 9, 1–26 (2018)
6. Leventhal, B.: Predictive analytics for marketers. Using data mining for business advantage. KoganPage, London, New York, NY, New Delhi (2018)
7. Kridel, D., Dolk, D.: Automated self-service modeling: predictive analytics as a service. *Inf Syst E-Bus Manage* 11, 119–140 (2013)
8. Halper, F.: Predictive Analytics for Business Advantage. Best Practices Report (2014)
9. Gandomi, A., Haider, M.: Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management* 35, 137–144 (2015)
10. Eckerson, W.: Predictive Analytics, <https://tdwi.org/Articles/2007/05/10/Predictive-Analytics.aspx>
11. Chamoni, P.: Advanced Analytics: Eine Annäherung. *BI Spektrum* 12., 8–10 (2017)
12. Siegel, E.: Predictive analytics. The power to predict who will click, buy, lie, or die. John Wiley & Sons, Hoboken, New Jersey (2016)
13. Janke, A.T., Overbeek, D.L., Kocher, K.E., Levy, P.D.: Exploring the Potential of Predictive Analytics and Big Data in Emergency Care. *Annals of emergency medicine* 67, 227–236 (2016)
14. Granovsky, L., Kamienchick, R., Yacovzada, N., Viswanathan, P., Cao, S., Alevras, D., Tamir, R., Grossman, I., Ferro, T., Chary, D., et al.: Using Predictive Analytics to Identify Risk of Clinical Asthma Exacerbations. *Journal of Allergy and Clinical Immunology* 141, AB222 (2018)
15. Bradlow, E.T., Gangwar, M., Kopalle, P., Voleti, S.: The Role of Big Data and Predictive Analytics in Retailing. *Journal of Retailing* 93, 79–95 (2017)
16. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P.: From data mining to knowledge discovery in databases. *AI magazine*, 17(3), 37-37, 37 (1996)
17. Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C. and Wirth, R.: CRISP-DM 1.0, <https://www.the-modeling-agency.com/crisp-dm.pdf>
18. SAS: Introduction to SEMMA, <https://documentation.sas.com/?docsetId=emref&docsetTarget=n061bzurmej4j3n1jnj8bbj1a2.htm&docsetVersion=14.3&locale=en>
19. Shearer, C.: The CRISP-DM Model: The New Blueprint for Data Mining. *Journal of Data Warehousing*, 13–22 (2000)
20. Iffert, L., Bange, C., Mack, M., Vitsenko, J.: Advanced & Predictive Analytics. Schlüssel zur zukünftigen Wettbewerbsfähigkeit (2016)
21. TDWI (ed.): Ten Mistakes to Avoid in Predictive Analytics Efforts (2015)

Anhang 1: Auswertung Literatur-Review

Quelle	Art	Branche	Ursprung	Effekt	Cluster	Use Case	Nutzen	Methoden	Details	Praktisc
Altabar et al. 2018	Journal	Medien	Hoher Konkurrenzdruck führt zu gesteigerter Interesse, die Performance von journalistischem Content zu verbessern und Vertriebskanäle zu kennen	Rechtweilenoptimierung	Prozessoptimierung	Vorhersage der Reichweite eines Artikels vor Veröffentlichung	Entscheidungsunterstützung Prüfen von bestimmten Artikeln Anpassung der Reichweite Zielgruppenanalyse	Clustering Zeitreihenanalyse		x
Altaran und Altaran 2018, S.22f	Journal	Einzelhandel	Markt- und Kundenstruktur unklar	Optimierung Targeting	Verkaufunterstützung	Markt- und Kundensegmentierung	Angebote bedarfsgerechter steuern durch Positionierung im Markt			
Altaran und Altaran 2018, S.22f	Journal	Einzelhandel	Es existieren Produkte, die häufig zusammengekauft werden	Crossselling zur Umsatzsteigerung	Verkaufunterstützung	Warenkorbanalyse	Crosssellingmöglichkeiten erkennen und anbieten	Assoziation		
Altaran und Altaran 2018, S.22f	Journal	Einzelhandel	Bestimmte Faktoren (z.B. Wetter) beeinflussen das Einkaufsverhalten	Identifikation von Trends	Wissen	Trends im Einkaufsverhalten erkennen	Neuproduktempfehlungen steuern Cross- und Upsellingmöglichkeiten vorbereiten und steuern			
Altaran und Altaran 2018, S.22f	Journal	Luftfahrt	Hohe Auslastung in Flugzeugen Kontinuität, niedrige Kosten Hoher Konkurrenzdruck der Branche	Preisfindung	Verkaufunterstützung	Optimalen Verkaufspunkt zu verschiedenen Zeitpunkten ermitteln Erkennen von Bedürfnissen der Kunden und effizienten Kunden-Loyaltäts-Angeboten	Erfolg des höchstmöglichen Umsatz durch Auslastung und Preisgestaltung Erfolgreiche Kunden-Loyaltätsprogramme führen zu Umsatzsteigerung und Kundenbindung			
Altaran und Altaran 2018, S.22f	Journal	Luftfahrt	Lieferverzögerungen führen zur Unzufriedenheit und geringen Verkaufseffekten	Beschleunigung	Prozessoptimierung	Erkennung der optimalen Route und Lieferzeitpunkte	Optimale Lieferwege reduzieren Kosten und beschleunigen den Transport			
Altaran und Altaran 2018, S.22f	Journal	Logistik	Schwierigkeiten und Verzögerungen	Effizienzsteigerung	Prozessoptimierung	Erfolgreiche Auswahl neuer Mitarbeiter	Optimale Ressourcennutzung führt zu Umsatz- und Effizienzsteigerungen			
Altaran und Altaran 2018, S.22f	Journal	Human Resources	Auswahl geeigneter neuer Mitarbeiter meist schwierig	Ressourcennutzung	Prozessoptimierung	Mitarbeiter optimal steuern	Genaue Abschätzung der Bonität dem Unternehmen Nutzerkosten Je genauer die vorgeschlagenen Personen, desto höher die Bindung des Users mit dem sozialen Netzwerk			
Altaran und Altaran 2018, S.22f	Journal	Human Resources	Mitarbeiterboom durch Digitalisierung In sozialen Netzwerken werden vorgeschlagen	Mitarbeiterbindung	Verkaufunterstützung	Optimierung der vorgeschlagenen Personen zum Vernetzen				
Altaran und Altaran 2018, S.22f	Journal	Online	Teils Vergabe von Krediten an kreditunwürdige Kunden Fehlentscheidung ruft hohen Schaden hervor	Empfehlungsoptimierung	Kundenerlebnis	Rekommendation durch verbesserte Einschätzung der Kreditwürdigkeit eines Kunden	Rekommendation durch verbesserte Einschätzung der Kreditwürdigkeit eines Kunden			
Altaran und Altaran 2018, S.22f	Journal	Finanzwesen	Fehlentscheidung ruft hohen Schaden hervor	Rekommendation	Sicherheit	Vorhersage der Kreditwürdigkeit eines Kunden	Rekommendation durch verbesserte Einschätzung der Kreditwürdigkeit eines Kunden			
Bauckhage et al. 2017	Journal	Medien	Vollständige saisonale Schere zwischen Newsblogs Schwierigkeit, die Reichweite zu steigern	Rechtweilenoptimierung	Prozessoptimierung	Forecasting-Modelle zu User-Interaktionen bei Newsblogs	gezielte Marketingstrategien Einschätzung der Bloggenität an Einschätzung der Bloggenität an Entscheidungsunterstützung	Zeitreihenanalyse		x
Bhaduri 2016, S. 47ff	Monographie	Medien	Ziel von Werbetreibenden ist das Erreichen einer Zielgruppe über verschiedene Kanäle zu geringen Kosten	Rechtweilenoptimierung	Prozessoptimierung	Optimierung des Media-Mix aus TV, Print und Internet für Werbausspielung	Gezielte Werbausspielung reduziert Kosten bei gleichzeitiger Erhöhung der Reichweite und somit Werbewirksamkeit			
Bradlow et al. 2017	Journal	Einzelhandel	Preise der Produkte bestimmen Profitabilität eines Stores	Ressourcennutzung	Prozessoptimierung	Anpassung des Preises von Einzelhandelsprodukten	Optimale Preissetzung			x
Ganovskiy 2018	Journal	Medizin	Geringe Überwachung von Asthma-Patienten und somit spät erkannte Verschlechterungen des Krankheitsbildes	Preisfindung	Verkaufunterstützung	Friszeitliche Erkennung von Asthma-Patienten mit Risiko einer deutlichen Verschlechterung des Krankheitsbildes	Friszeitliche Erkennung von Asthma-Patienten mit Risiko einer deutlichen Verschlechterung des Krankheitsbildes			x
Ganovskiy 2018	Journal	Medizin	Krankheitsbildes	Optimierung Targeting	Verkaufunterstützung	Verkaufunterstützung	Frühzeitige Erkennung von Asthma-Patienten mit Risiko einer deutlichen Verschlechterung des Krankheitsbildes			x

Quelle	Art	Brancho	Ursprung	Effekt	Cluster	Use Case	Nutzen	Methoden	Details	Praktisc
Jain 2016	Journal	Handel	Geringe Erfolgsquote von neu eingekauften Produkten trotz hohen Umsatzpotenzials	Verständnisgewinn	Wissen	Schnelles Erkennen der Ursache für geringe Nachfrage neuer Produkte	Gezielte und schnelle Reaktion möglich: Kommunikation	Social-Media-Analyse	Text-Analyse	x
Lisey 2017	Journal	Bibliothek	Lange Wartezeiten in Bibliotheken bei Ausleihe von Büchern und viele Interlibrary-Translationen	Beschleunigung	Prozessoptimierung	Automatische Erkennung zukünftiger Bedarfe	Bestand wird frühzeitig auf Bedürfnisse angepasst	Clustering	k-means	x
Munz 2018	Journal	Luftfahrt	Viele Stakeholder an Luftfahrt beteiligt; ungenutzte Akonizitätszellen	Ressourcennutzung	Prozessoptimierung	Live-System zur Schätzung von Kundenverhalten	Bessere Resourcenutzung	Regression	Neuronale Netze	x
Niklas et al. 2017	Journal	Handel	Hier Konkreterzdruck bei Online-Shops	Erhöhung Verkaufschance	Verkaufsunterstützung	Proaktive Ansprache von Online-Kunden	Verhinderung von Dropouts durch rechtzeitige und gezielte Wartung (führt zu Kostenreduktion)	Prognose		x
Phil et al. 2017	Journal	Industrie	Maschinenausfälle kostenintensiv und risikoreich	Ressourcennutzung	Prozessoptimierung	Erkennen von zu wartenden Komponenten vor Ausfall	Kostenreduktion	Klassifikation		x
Siegel 2016, S.298ff	Monographie	Öffentlicher Dienst	Polizei oft zu spät an Tatorten; Streifenwagen absichern	Sicherheitssteigerung	Sicherheit	Potenzial nächste Orte für Kriminaldelikte erkennen	Streifenwagen gezielt gesteuert			
Siegel 2016, S.298ff	Monographie	Versicherung	Nicht alle Transaktionen können durch Banken und Versicherungen gleichermaßen überprüft werden	Betrugsminderung	Prozessoptimierung	Betrugserkennung von Transaktionen / Anwendungen	Gezielte Überprüfung der Transaktionen mit höherer Betrugswahrscheinlichkeit			
Siegel 2016, S.298ff	Monographie	Online	Spam-E-Mails stellen Sicherheitsrisiko für den Anwender da	Sicherheitssteigerung	Sicherheit	Erkennung von Spam-E-Mails	Verschieben der potenziellen Spam-E-Mails in einen separaten Ordner			
Siegel 2016, S.298ff	Monographie	Human Resources	Firmen wollen (überraschende) Kündigungen von guten Mitarbeitern	Mitarbeiterbindung	Wissen	Erkennung der Mitarbeiter, die potenzielles Kundenpotenzial darstellen	Berücksichtigung der potenzielle Kundiger			
Siegel 2016, S.298ff	Monographie	Übergreifend / Vertrieb	Ungewünschte Kunden durch Kundenkartei	Identifikation von Trends	Verkaufsunterstützung	Identifizierung von Marketing-/Kunden, die langfristige abspinnen	Gezielte Auswertung von Marketing-/Kundenbindungsmaßnahmen für die gefährdeten Kunden			
Siegel 2016, S.298ff	Monographie	Online	Bestimmte Plattformen bieten Firmorschläge für Nutzer an	Kundenbindung	Verkaufsunterstützung	Vorschlag von Filmen auf Basis der angeschauten und bewerteten Filme	Höher Trefferquote in den Vorschlagswerten erhöht Kundenbindung			
Siegel 2016, S.298ff	Monographie	Übergreifend / Marketing	Ansprache vieler Marketingkontakte kosten- und ressourcenintensiv	Ressourcennutzung	Prozessoptimierung	Erkennen der Marketingkontakte mit der höchsten Response-Wahrscheinlichkeit	Das gezielte Kontaktieren der Kontakte mit der höchsten Wahrscheinlichkeit einer Reaktion spart Kosten und Ressourcen			
Siegel 2016, S.298ff	Monographie	Medien	Auspielung von Werbung wirkungslos, wenn keine Interaktion erfolgt	Optimierung Targeting	Prozessoptimierung	Auspielung von Online-Werbung mit der höchsten Interaktionswahrscheinlichkeit	Hohe Interaktion mit Werbung verspricht Marketingerfolg			
Wedel und Kannan 2016	Journal	Online	Kunden haben individuelle Bedürfnisse, die über generische Angebote nicht befriedigt werden können	Personalisierung	Kundenerlebnis	Individuelle Angebote für Kunden erstellen	Kundenindividuelle Angebote versprechen höhere Abschlussraten			
Yaemin 2013	Journal	Öffentlicher Dienst	Universitäten wollen hohe Abbruchquote, insbesondere bei Fernstudengängen, vermeiden	Studentenbindung	Verkaufsunterstützung	Studienabbruch bedingen und darüber Vorwarnung von potenziellen Abbruchrisikofaktoren	Potenzielle Abbruchrisikofaktoren können frühzeitig erkannt werden und mit Maßnahmen unterstützt werden	Klassifikation		x

Comparing Brand Perception Through Exploratory Sentiment Analysis in Social Media

Mario Cichonczyk and Carsten Gips

Bielefeld University of Applied Sciences, Minden, Germany
firstname.lastname@fh-bielefeld.de

Abstract. The presented student project outlines a natural language processing pipeline for brand metric comparison in the Twitter ecosystem. Sentiment calculation for an unlabeled data set is demonstrated and calibrated using the statistical Central Limit Theorem as a guidance to anchor the sentiment indicator in a homogeneous market. The process is evaluated by comparing the sentimental market performance of three leading German logistics companies. A support for the value of sentiment analysis for automated customer feedback analysis in real-time is concluded.

1 Introduction

The brand philosophy behind a business is usually a driving principle of the entrepreneurial actions it follows. Ideally, these actions accumulate in a brand strategy and a finely tuned marketing mix to acquire market share and establish brand awareness and perception. The success of these marketing efforts can be measured by evaluating the time-delayed return on investment of associated profit margins. This approach has a deficit in explanatory power as it lacks a fine-grained insight into the complex effects of diversely faceted, multi-channel marketing and brand positioning methods. Therefore, marketing research relies on qualitative and quantitative analyses and surveying techniques for a more sophisticated evaluation of marketing investment impact. Targeted studies with resource expenditure are employed to answer specific questions of subjective brand perception. With technological development and progress, new approaches may be introduced to increase the efficiency of effect monitoring and thereby reducing inertia in strategic realignment according to market feedback. [18]

Since social media is getting more established in everyday life, intelligence can be gathered through a new and essentially cost-free feedback channel [12]. While social media marketing is concerned with the public relations effort in direction to the customer, the same platforms allow for an inversion of communication from consumer to business. The presented work explores how brand perception can be measured and compared by making use of natural language processing

Copyright © 2020 by the paper's authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

in the Twitter ecosystem. To achieve this objective, the hashtag space of leading German logistics companies was analyzed and collected as an exemplary use case. An analytics pipeline was constructed and applied to the dataset to better understand the customer base and approximate overall consumer sentiment towards the logistics brands, linking a scalar metric to consensus opinion as outlined in [3]. This approach can show the value of sentiment analysis in social media analysis for customer satisfaction research of single companies and it will also allow for contextualization in regard to competitors. As social media is essentially a means of open many-to-many communication, the same pipeline can be applied to feedback aimed towards other brands and will then allow a more empirical comparison.

2 Approach

The main contribution of this work aims to show how current ideas in natural language processing may be applied to take advantage of automated social media brand perception analysis. We would like to outline our course of action and the thought process used in this applied research project as it may inspire other research towards the stated problem case. For this purpose, two typical marketing query questions were chosen as an example to demonstrate the approach:

1. How do leading German consumer logistics brands rank in Twitter user approval?
2. How can Twitter meta information augment user approval analysis?

Related works predominantly make use of machine learning on labeled data with the primary goal of method testing and evaluation [28][15][26]. These approaches perform well regarding categorization in sentiment classes, but they cannot be directly transferred to the investigated problem without manual data annotation as a pre-step for model training. Real-world data is unlabeled and seed datasets do not yet exist for the highly specific domain of German logistics, hence seemingly producing an unsupervised clustering problem with feature engineering and selection in focus.

The specific characteristics of brand perception analysis allows for a third option. Munoz & Kumar [23] point out that perceptual metrics help to gauge the effectiveness of brand-building activities across points of customer interaction. Brand profiling can be achieved by fixing an indicator within a metric of a market and then comparing this indicator to a companies brand and its competitors [23]. Therefore, the aim of the presented approach is to acquire such a polarity indicator. Tools, models and methods from both supervised and unsupervised natural language classification and processing become available with this modified problem definition. The constructed data pipeline adheres to this premise and is presented in detail in the following paragraphs.

2.1 Collection and Transformation

Data acquisition was implemented using the Tweepy¹ library for the Python programming language to interact with the commercially available Twitter API² for developers. Tweepy was chosen for its native ability to handle rate limited free access tokens and dynamic adjustment of traffic bandwidth. To stay within these limits and to focus on *customer* opinion, only the top three logistics companies for the courier, express and parcel services in Germany where selected, which as of 2018 are DHL, Hermes and DPD [13]. Therefore, the Tweepy filter

```
"#dhl OR #hermes OR #dpd OR dhl OR hermes OR dpd"
```

was constructed and 10594 tweets were collected over 3 weeks in the winter of 2018, limited to tweets with a set "German" language flag. The Twitter API returns tweets in JSON format, containing a large number of partly redundant data fields. All JSON data was parsed and attributes of interest to the research question were selected and named accordingly. The selected and renamed fields were "usr_id", "tweet_id", "usr_followers", "timestamp", "favorites", "retweets", "client", "hashtags" and the "text" itself. After transformation, the tweets were stacked as rows to form a 10594x9 matrix **M**.

2.2 Data Exploration, Filtering and Feature Construction

First exploration and inspection of the dataset had shown that the Tweepy filter was not sufficient to separate the communications concerning the three selected logistics brands in a satisfactory manner. The data contained tweets which did not directly relate to the domain of interest. It became clear that a more in-depth analysis of the communication topics was necessary to further sanitize the results. Tweet topic modeling through utilization of keyword annotations - better known as "hashtags" - was identified to hold valuable advantages for the solution of this problem [31][29]. Based on the work presented by Wang, Wei & Zhang [30], a graph model was defined by the set of all hashtags $\mathcal{H} = \{h_0, h_1, \dots, h_m\}$ contained within the dataset, where each hashtag h_i represents a node weighted by its global occurrence count and is associated with a set of tweets $\mathcal{T}_k = \{\tau_0, \tau_1, \dots, \tau_n\}$ in which it occurs. The set of edges \mathcal{E} consists of a link between two hashtags if they co-occur in the same tweet. The weight of an edge e_{ij} between h_i and h_j is incremented for each co-occurrence of h_i and h_j . The graph model $\mathbf{HG} = \{\mathcal{H}, \mathcal{E}\}$ was used to isolate the logistics subgraph of interest which reduced \mathcal{T} , and thereby the rows of **M**, to only hold tweets relevant to the research.

The "hashtags" column of **M** was transformed to **HG** and stored in GEXF-format³. This step made it possible to import the hashtag graph into existing graph analysis software⁴, providing access to pre-optimized methods. The Yifan

¹ <https://github.com/tweepy/tweepy>

² <https://developer.twitter.com/>

³ <https://networkx.github.io/documentation/stable/reference/readwrite/gexf.html>

⁴ <https://gephi.org/>

Hu multilevel layout algorithm was chosen for its speed and good quality on large graphs [11] to achieve topic modeling. With this technique, all node and edge weights are used as force simulations of attraction and repulsion, creating a dispersed planar graph projection. Clusters of hashtags in frequent co-occurrence form topic subgraphs while unrelated topics drift apart. After the embedding step, topics weakly connected to logistics can be visually identified.

The collective topics "dhl", "dpd" and "hermes" were found to form a distinct subgraph with minor outliers regarding "jobs" topics. Stronger interrelations to unrelated themes existed for the single topic "hermes". This effect can be attributed to the ambiguity of the term. Subgraphs concerning "fashion" and "export politics" were intertwined with tweets about the logistics company. The set of hashtags identifying those outlying tweets was added as a filter to remove non-logistics rows from \mathbf{M} , resulting in an on-topic dataset.

Tweets which did not resemble a consumer opinion were filtered out, e.g. advertisements and news were undesirable data as they would distort the sentiment analysis. Therefore, the column "client" was investigated. Under the assumption that consumer opinion is predominantly voiced through consumer client software, other agents can be ignored. All user agents of the "client" column were manually mapped to the categories "Android", "iOS", "Desktop", "Other" and "Professional". "Other" resembles multi-platform clients which cannot directly be associated with a single platform. The "Professional" label identifies all user agents known to be developed for commercial usage, e.g. tweet automation or social media management software. All "Professional" rows were removed. Commercial users who rely on consumer client software are exempt from this pruning step. These were further analyzed by the count of their followers. For all rows in \mathbf{M} , the unique "usr_id" fields were collected and then ranked by their "usr_followers" value. Since the Twitter follower count adheres to the power law [22], manually inspecting the top followed profiles was sufficient to mark opinion bearing commercial accounts for removal and neutralize their influence on global sentiment. For professional tweets which may have been remaining in the data after all filters, it was assumed that their opinion influx was significantly outweighed by the now superior consumer class tweets.

To finally conclude filtering and feature construction, \mathbf{M} was extended by the columns "is_dhl", "is_dpd", "is_hermes", "hour" and "weekday". With these extra columns, searching and querying the dataset for the following steps is faster and more convenient. Finding tweets associated with specific brands is done by simple boolean masking, resulting in desired subsets of \mathbf{M} without having to repeatedly parse and search tweet content for each query. Aggregation over time windows is sped up by pre-splitting the complex timestamp format of the Twitter API.

After the outlined data sanitation treatment, actual text analysis was possible.

2.3 Text Preprocessing

The Natural Language Tool Kit (NLTK) offers the basic functionalities necessary for natural language processing [2]. As such, it was used to pre-process the

tweet content information formulated by Twitter users. The "text" column was tokenized and all single tokens of interest were added as a new "tokens" column, containing a list of tokens for each tweet. All tokens were stripped of non-textual information, URLs removed, umlauts converted and otherwise undesirable information filtered out. The sanitized tokens represent all German words of a tweet, but not all words hold analytic information. NLTK provides a list of stop words for several languages, including German. Accordingly, all tokens were checked against the German NLTK stop word list and removed if they were considered a match. Afterwards, the term frequency of all tokens was calculated to identify other possible stop words. Several high ranking tokens were found to lack value for analysis:

```
"dhl, dpd, hermes, paket, mal, dass, schon, kommt, immer, seit,  
fuer"
```

These were added to the stop word list and also removed. Brand name tokens are redundant as their information is already stored in **M** (see Paragraph 2.2). With the token list constructed, semantic polarity estimation can be examined on a word level.

2.4 Token Polarity

Since algorithms do not by default have any understanding of the emotional impact of word semantics, sentiment analysis relies on human consensus opinion. Databases with annotated word polarities between $[-1, 1]$ for negative and positive sentiment respectively are used to look up a scalar value for a given token. For the German language, such a dictionary exists in the form of the SentiWS [27] project. As a first, naive approach, the pipeline's sentiment resolver tries to annotate the tokens of **M** directly through a query to SentiWS. If the word is present in the dictionary, no further search is required and its sentiment can be returned. SentiWS contains about 3500 basic forms and 34000 inflections. Despite this size, less than 15% of all tweet tokens were found in SentiWS. This result is not surprising given the combination of language syntax complexity, a raw tweet tokens count of more than 80000 and the effort involved in dictionary building. It was expected that only a low number of entries would directly match. Therefore, each token that could not be resolved in SentiWS is forwarded to the NLTK German stemmer. Stemming is the process of reducing a complex word, which might be an inflection, to its basic form [19]. This step can be understood as a simplification of the token, or in a more technical sense even a functional projection. The goal is to project the token space sample in such a way that its transformation aligns with the SentiWS target space. Thereby, a morphological alternative is potentially found and might allow for a sentiment look up. Increasing the number of token morphemes using this method also increased polarity coverage.

If the stemmer was not able to find a morpheme in SentiWS, syntactical alternative search is exhausted. Therefore, the actual *meaning* of the token can

be used to find synonyms whose sentiment is known. Liebeck [17] found the introduction of semantic equivalence to be of advantageous value for more thorough sentiment analysis and referred to synonym search in synset databases like GermaNet [10]. Mohtarami [21] et al. explored and alternatively suggested the use of vector-based approaches for the same purpose. They observed that WordNet [20] - and therefore GermaNet as a descendant - perform satisfactory when *semantic* synonyms are searched but lack in accuracy when *sentimental* equivalence is the primary metric. To accommodate this problem, they introduced emotional features of words to construct their vectors. The key insight for the presented work is the necessity of a more general human-like language intelligence to identify word alternatives beyond pure semantic synonyms. Webber [32] came to the same conclusion and presented a proof of concept disambiguation and language analysis system trained on half a million Wikipedia articles instead of a domain specific corpus. The results suggested superior context-based general language processing capabilities. Therefore, a similar approach was chosen. Instead of synonym search in statically assembled synsets, a Word2Vec model trained on Wikipedia articles was used to find alternatives for tokens which neither themselves nor their stemmed variants could be resolved through SentiWS. Word2Vec was chosen due to its proven performance [8] regarding this purpose and because a large (650 million words), pre-trained, general Wikipedia knowledge model already exists⁵ for the gensim⁶ Word2Vec implementation. Training a similarly large model would not have been possible for the purposes of this student project.

The lookup process was constructed as follows. Gensim is used to retrieve a vector space embedding for the token, e.g. the word "house". As stated above, this vector representation shares a topological vicinity with its contextual synonyms, e.g. "building" or "home". The aim is to find a spatially close synonym which can be associated with an entry in SentiWS. Starting from the "house" embedding, its neighbouring entries are probed in order of increasing distance. For example, "building" may highlight as the closest approximation to "house". The word "building" is therefore chosen as an alternative candidate. This candidate is then tested against SentiWS and if a polarity value could be retrieved, the candidate is selected. In this case, "building" may not have been a valid alternative, is rejected and the process continues with the next neighbour in increasing distance, which is "house". The new candidate is tested in the same manner and can be successfully matched with a sentiment, leading to its selection as a valid alternative to "house". The algorithm encapsulates the same process a human would employ in thinking about other phrasings of the same utterance. Figuratively speaking, the amount of imagination necessary to come up with another phrase is continually incremented until a suitable rephrasing is found. This can of course lead to misleading levels of synonymity if done ad infinitum. Therefore, it was decided to penalize the retrieved sentiment by the distance between the original token vector and its alternative, similarly to Kim & Shin [14].

⁵ <https://github.com/devmount/germanwordembeddings>

⁶ <https://radimrehurek.com/gensim/models/word2vec.html>

The complete algorithm to resolve the sentiment value for a token vector t can be defined as follows:

$$sentiment(t) = \max_{s \in \text{SentiWS} \cap \text{Word2Vec}} \frac{\text{SentiWS}(s)}{D(s, t)} \quad (1)$$

Note that this algorithm implicitly neutralizes alternatives if they are too broadly associated synonyms:

$$\lim_{D(s, t) \rightarrow \infty} sentiment(t) = 0 \quad (2)$$

The end behaviour of $sentiment(t)$ ensures the absence of polarity distortion in synonym search.

All tokens in the data were labeled using this process.

2.5 Sentiment Weighting and Analysis

After the tokens of \mathbf{M} were given an emotional weight as outlined in Section 2.4 on a word level, further analysis on a sentence level can proceed. Fang & Zhan [5] summarize that every word of a sentence has its syntactic role which defines how the word is used. These roles, also known as part of speech, have significant impact on the importance of their underlying sentiment for the polarity of the complete sentence. For example, words like pronouns usually do not contain any sentiment and are therefore neutral. In contrast, verbs or adjectives can hold different weights respectively [5]. Part of speech taggers are used to classify words according to their syntactic role. The NLTK tagger class has been extended for the German language and trained on the TIGER [4] corpus in a different project⁷, achieving an accuracy of 98% as stated by the authors. This effective tagger was chosen for the presented work due to its good performance, generalization capabilities and fast integration in NLTK. After POS processing, all tokens were labeled according to the STTS [33] tag system. Nichols & Song [25] have examined the relationship between scalar sentiment, part of speech and overall sentence polarity. They empirically compared the influence of POS strengths on classifier performance and approximated an optimal solution. Their exhaustive search for the set $POS = \{noun, verb, adjective, adverb\}$ and the strength weights $str(POS_i) \in \{1, 2, 3, 4, 5\}$ has shown that the best performance for purposes of sentiment analysis was achieved with the following scalar weight vector:

$$(str(noun) = 2, str(verb) = 3, str(adv) = 4, str(adj) = 5) \quad (3)$$

A mapping from the STTS tags to the categories utilized by Nichols & Song was introduced to ensure compatibility between the German tagger and their weighting approach. Afterwards, all tokens were accentuated according to their syntactical sentence function, resulting in increased sentiment variance and therefore more expressive overall tweet polarity.

As a last step before the culminating conflation of all individual token polarities per tweet, negations need to be handled as they significantly influence the

⁷ <https://github.com/ptnplanet/NLTK-Contributions/tree/master/ClassifierBasedGermanTagger>

calculated emotion by their valency scopes [6]. Two primary ways for negation handling were tested: syntax scope analysis and a heuristic approach. Carrillo [1] et al. proposed that superior performance is achieved if the negation scope is determined by examining the valence subtree of the negation token based on part of speech association. After successfully labeling each word with a POS tag, the grammatical syntax reveals the subtree which is supposed to be negated and therefore inversely influential on sentiment. While this approach would grant realistic language sentiment, it presupposes that the syntax tree is immaculate. Especially for Twitter, this is rarely the case. Gui [9] et al. found that the Twitter culture of mutual communication is inherently comprised of non-standard orthography and reconstructing an approximately valid syntax tree requires substantial effort. Their findings were confirmable and therefore opposed the syntactical negation handling as proposed by Carrillo [1] et al. for practical appliance in the presented project. For this reason, the more widely [6] used heuristics solution was employed. The German tagger was able to reliably identify the negation token itself (e.g. "nicht") and labeled it accordingly with the fitting STTS tag. This label gave an anchor to which a rule-based negation heuristic could be expediently attached. Inspired by the syntactical solution, the heuristic successively searches for the next token with a sentiment that has been weighted by its tag (see the beginning of this section). The rationale behind the algorithm is that the sentiment bearing successor feature is assumed to be the most likely target for negation. Samples suggested that this heuristic rule performs sufficiently in relation to the goals of analysis.

After all negations were handled, it was finally possible to propose an estimated polarity per tweet. Similarly to the work of Kumar & Sebastian [16], sentiment was calculated by summing the weighted and - if necessary - negated token polarity scalars. The resulting values were added as new column to **M**.

2.6 Scale Calibration

Having calculated a value which one might consider "sentiment" is not enough for actual market analyses due to two reasons:

1. The scale - while argumentative coherent and grounded in the outlined rationale - can be understood as a valid indicator, it is still arbitrarily defined. Its definition is sound, but given that the scale is supposed to measure levels of human emotion, it needs validation. Such a test would require human evaluation, altering the problem to a supervised interpretation.
2. As Section 2 stated, Munoz & Kumar [23] emphasize indicator fixation and anchoring within the metric of a market to achieve brand profiling. Only then is empirical comparison to the calibrated indicator, and thereby competitive brands, feasible.

These two problems seemingly demand further research and evaluation. Contradistinctively, it is argued that the *combination of both* allows for a use-case specific solution if the fundamental nature of the underlying data is exploited by

utilizing the broad scope of opinions being uttered on Twitter. This characteristic permits the introduction of the established statistical Central Limit Theorem (CLT). The CLT is the observation of the convergence behaviour of probability distributions of an increasing number of one- or multi-dimensional random variables to a normal distribution [7]. For a public opinion surveying purpose as presented, the CLT leads to a beneficial conclusion: if a sufficiently large number of unrelated, random sample opinions are gathered from a sample population, the overall sample mean will be normally distributed around the population mean. Furthermore, if the opinion distribution is limited to the interval $[-1, 1]$ by the pre-conceived sentiment constraint and additionally, baseline polarity is assumed to be neutral, all essential properties of the expected opinion distribution are therefore known in advance without human intervention for validation. To exploit this reasoning for the calibration of the proposed polarity estimation process, a second Twitter dataset **GT** (for **G**round **T**ruth) was collected using the Tweepy filter

"#2018 OR #2019 OR #december OR #january"

and is processed through the same pipeline as **M**, leading to a broad, dispersed set of tweets unrelated to any specific topic. As these tweets cover a wide range of *independent* themes and conversational domains, it can be reasoned that the global population sentiment characteristics behave according to the CLT. This theory resembles the missing link between the "arbitrarily" constructed polarity estimation pipeline and actual market sentiment, resulting in the desired indicator described by Munoz & Kumar. Establishing the connection mathematically is possible in a multitude of ways, as long as the link adheres to the following formalism. Since the global sentiment distribution of the general dataset **GT** should at best follow the listed constraints, its histogram $\Phi(GT_{polarity})$ should resemble the normal distribution as close as possible. As such, the aim is to find a projection of $GT_{polarity}$ which minimizes the error between the histogram and the normal distribution. If such a projection is found, it acts as the calibration metric for the analytics pipeline. Thereupon, the calibrated projection can be used on the actual logistics dataset to infer class labels in relation to the opinion of the general population. If the distribution is discretized at the interquartile ranges (IQR), half of all opinions fall in the central area. These will be considered "neutral" and make up the overall majority. A quarter of all opinions fall left of the first IQR marker and will be considered negatively extreme. Their class label is "negative". And lastly, the remaining datapoints fall beyond the third IQR marker and are hence labeled "positive" as they express positively extreme sentiment. This baseline polarity will be the ground truth reference and **M** can be labeled in the same way, using the absolute sentiment boundaries dictated by **GT**. Subsequently, sentiment analysis and classification are concluded and evaluation of logistics opinions is finally possible.

3 Analysis

For evaluation, the questions put forward in Section 2 were answered using the constructed pipeline.

After discretization, the Twitter class label distribution is normally distributed and zero centered. The IQR markers force the calibration into the CLT assumption. Therefore, specialized tweet topics can be compared by calculating the relative distance between the class label tendencies.

3.1 "How do leading German consumer logistics brands rank in Twitter user approval?"

For the complete logistics dataset **M**, the class labels deviate from the Twitter baseline **GT**. Neutral sentiment is 11.92% less present in tweets relating to logistics while positive and negative sentiment are 4.58% and 7.34% above baseline respectively. This observation of increased variance shows that users communicate with higher emotional tendencies towards the topic. It can be concluded that opinions regarding the logistics domain are mostly stated more vigorously. To reduce selection bias, the logistics brands must therefore be compared exclusively within their domain. Otherwise, their relative ranking would be distorted by the overall preconceived notions of opinion. Appendix Figure 1 (top) visualizes this distortion. On first glance, all three brands perform with high emotional response, skewed towards negative feedback. This issue is the result of the distributional relationship to $\Phi(GT_{polarity})$. Drawing the conclusion that the three specific brands perform bad on Twitter is not precise as the entire domain *generally* provokes the shown response. Due to this implication, a more accurate baseline indicator for performance ranking is the sentiment histogram of the logistics domain. All brands react differently and more truthfully to this metric and better conclusions can be drawn, as presented in Appendix Figure 1 (middle). Therein, it can be seen that the relative ranking is now increasingly expressive. DHL performs better than its competitors within the domain, having less negative class labels and more positive class labels than average. DPD and HERMES are negatively skewed beyond average expectation, performing worse than DHL. HERMES exclusively falls behind in both negative (more than average) and positive (less than average) opinions.

3.2 "How can Twitter meta information augment user approval analysis?"

The presented comparison solely relies on single tweet content and thus individual sentiment. The Twitter API grants access to information beyond pure textual data. Correlating the meta data to polarity can yield clarified insight. For example, one of the defining functionalities of Twitter is the ability to like and/or share ("retweet") opinions of other users. The implications for sentiment statistics are pivotal. Nagarajan, Purohit & Sheth [24] observed different levels of endorsement by peer users depending on tweet positivity. Hence, weighting tweet sentiment by

the amount of shared and approved peer affirmation links argumentative popularity to brand performance indication. Furthermore, if a large enough dataset is gathered, sentiment can be followed along the chain of retweets, forming an interesting graph traversal problem. It could be mapped out how positive and negative sentiment propagate through the Twitter ecosystem and how these multiplicative patterns differ for brands. For the presented project, the dataset is not large enough to reconstruct such patterns. Nonetheless, each entry of \mathbf{M} does contain the integer counts of likes and retweets and these values hold analytic value. Reasoned by the arguments above, the two integers were summed per row and added as a new column labeled "Propagation". The new value resembles the amount of persons sharing the view being expressed in the tweet and can be used as a weight vector for sentiment aggregation. The results of Nagarajan, Purohit & Sheth were observable afterwards aswell. Our observations indicate that tweets in the logistics domain are generally shared less often than baseline, but *if* they are shared, they are more likely negative in contrast to **GT**. Due to this finding, the class distributions change significantly if polarity propagation is factored into relational brand metrics as shown in Appendix Figure 1 (bottom). User approval leads to considerable amplification of disparity. DPD and HERMES shift their distributions to pronounced neutrality. They both reduce negative sentiment slightly but also exceedingly decrease in positive sentiment by a large factor. In contrast, DHL overwhelmingly profits from the shared user opinions. Negative and neutral sentiment labels are shifted to an 11.04% increase in positive sentiment.

In addition to likes and retweets, the data also contains the timestamp at which a tweet was written. Especially in the logistics business, time plays an important role and should therefore be targeted analytically. In Appendix Figure 2, a heatmap of aggregated class labels per day and hour, averaged in mean rows and columns, is shown. For aggregation, the labels were interpreted as $\{-1, 0, 1\}$ for $\{negative, neutral, positive\}$ respectively. In **GT**, it can be observed that negative sentiment correlates with business hours. Furthermore, negativity peaks towards the end of the business week. Intuitively, non-delivery at the beginning of the weekend may induce disappointment as the customer would have to wait past the work-free days until the next business day. Such a hypothesis could be further evaluated if domain knowledge is introduced.

For the individual brands, different observations stand out:

1. DHL: Generally, DHL performs above average on Mondays. Sentiment then declines with progression of the week. Negativity peaks at the weekend.
2. HERMES: Best performance is expected on Tuesdays. Daily peak negativity shifts with weekly progression. Customers tend to express negative feedback incrementally in later hours, peaking at 20:00 Hours. The relation may be connected to the longer business hours employed by HERMES.
3. DPD: No obvious pattern is present except a sentiment low on Mondays 14:00 Hours. Otherwise, the data correlates to the baseline.

The underlying tweet texts and their themes were investigated at the emphasized days and hours but did not reveal any apparent common denominator ex-

plaining their occurrence in addition to their mere temporal correlation. These time distributions can serve more appropriate analyses for research with added knowledge of the internal structures of the different companies. Then, more precise assumptions can be made about the cause of the observed patterns. The different client agents were also investigated in relation to sentiment but did not show any discernible correlation.

4 Conclusion

The outlined process has shown how powerful Twitter can be for sentiment analysis in brand perception polling. Especially when meta data is introduced, insights far beyond classical polling techniques become apparent. The different logistics brands have shown distinguishable approval performance and the inclusion of Twitter meta data was beneficial to inquire into these variations. Acquiring a professional API access may be costly at first, but the increase in data volume and quality can add more capabilities, such as instant metrics or even precise geolocation, a variable directly linkable to geographical key performance indicators in logistics. As such, the real-time data pipeline could be integrated into business intelligence and monitoring systems for anomaly detection and cause-effect analysis.

Furthermore, natural language processing methods were employed to demonstrate their value for modern-day feedback evaluation. In-depth studies into the many different ways to approach language processing problems may highlight even more fruitful pipeline steps. Building a domain specific language corpus should be the first obstacle to take in that direction. The current lack thereof discouraged the usage of many techniques like supervised machine learning. The same holds for the limited size of the tested data set. If more comprehensive collection was possible, access to other ways of statistical description and model building opens up. Especially in regard to finer recognition of irony and sarcasm, further improvement of the pipeline is required. In its current form, sarcasm was not adequately recognized. Relative brand metric comparison is still valid, as all brands suffered from this deficit in the same way. Sarcasm seems to be inherently linked to strongly opinionated utterances, which is a reason why the proposed workflow did not rely on emoticon recognition as other works suggest for label inference. Data exploration has shown that emoticons played a predominant role in sarcasm emphasis. This observation may hold value for further research but discouraged their importance for the current student project.

Summarizing, the work supports the assumption that modern social media can have a vital contribution to fast refinement of a brand's marketing mix for strategic realignment in real-time. This is a benefit to ensure positive reception of corporate philosophies directly as a reaction to automated analyses of innovative, digital consumer feedback channels, using contemporary research in natural language processing.

Appendix



Fig. 1. Twitter sentiment compared to logistics brands.

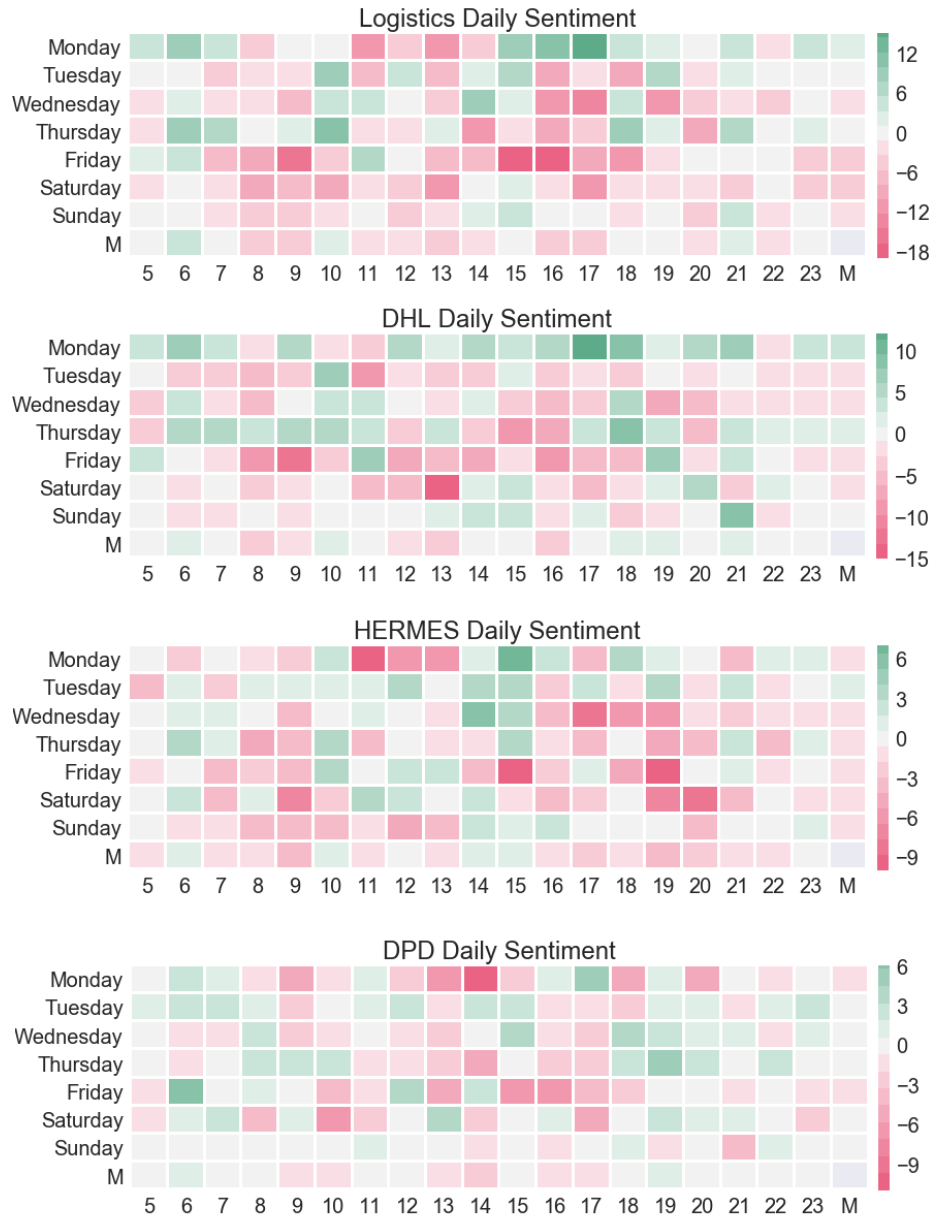


Fig. 2. Daily logistics sentiment.

References

1. Carrillo de Albornoz, J., Plaza, L., Diaz, A., Ballesteros, M.: Ucm-i: A rule-based syntactic approach for resolving the scope of negation. In: *SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012). pp. 282–287. Association for Computational Linguistics (2012), <http://aclweb.org/anthology/S12-1037>
2. Bird, S., Loper, E.: Nltk: The natural language toolkit. In: Proceedings of the ACL 2004 on Interactive Poster and Demonstration Sessions. ACLdemo '04, Association for Computational Linguistics, Stroudsburg, PA, USA (2004). <https://doi.org/10.3115/1219044.1219075>
3. Culotta, A., Cutler, J.: Mining brand perceptions from twitter social networks. *Marketing Science* **35**(3), 343–362 (2016). <https://doi.org/10.1287/mksc.2015.0968>
4. Dipper, S., Kübler, S.: German Treebanks: TIGER and TüBa-D/Z, pp. 595–639. Springer Netherlands, Dordrecht (2017)
5. Fang, X., Zhan, J.: Sentiment analysis using product review data. *Journal of Big Data* **2**(1), 5 (Jun 2015). <https://doi.org/10.1186/s40537-015-0015-2>
6. Farooq, U., Mansoor, H., Nongailard, A., Ouzrout, Y., Qadir, M.A.: Negation handling in sentiment analysis at sentence level. *Journal of Computers* **12**, 470–478 (01 2016)
7. Fischer, H.: A history of the central limit theorem: From classical to modern probability theory. Springer Science & Business Media (2010)
8. Goldberg, Y., Levy, O.: word2vec explained: deriving mikolov et al.’s negative-sampling word-embedding method. *CoRR* **abs/1402.3722** (2014), <http://arxiv.org/abs/1402.3722>
9. Gui, T., Zhang, Q., Huang, H., Peng, M., Huang, X.: Part-of-speech tagging for twitter with adversarial neural networks. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. pp. 2411–2420 (2017)
10. Hamp, B., Feldweg, H.: Germanet-a lexical-semantic net for german. *Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications* (1997)
11. Hu, Y.: Efficient and high quality force-directed graph drawing. *Mathematica Journal* **10**, 37–71 (01 2006)
12. Jansen, B.J., Zhang, M., Sobel, K., Chowdhury, A.: Twitter power: Tweets as electronic word of mouth. *JASIST* **60**, 2169–2188 (2009)
13. Jansen, B.J., Zhang, M., Sobel, K., Chowdhury, A.: Umsatzverteilung im kependkundenmarkt in deutschland nach anbietern im geschäftsjahr 2017/18. *Handelsblatt* **134**, 16 (2018)
14. Kim, Y., Shin, H.: A new approach for measuring sentiment orientation based on multi-dimensional vector space. *CoRR* **abs/1801.00254** (2018), <http://arxiv.org/abs/1801.00254>
15. Kouloumpis, E., Wilson, T., Moore, J.D.: Twitter sentiment analysis: The good the bad and the omg! *Icwsn* **11**(538-541), 164 (2011)
16. Kumar, A., Sebastian, T.: Sentiment analysis on twitter. *International Journal of Computer Science Issues* **9**, 372–378 (07 2012)
17. Liebeck, M.: Aspekte einer automatischen meinungsbildungsanalyse von online-diskussionen. In: Ritter, N., Henrich, A., Lehner, W., Thor, A., Friedrich, S., Wingerath, W. (eds.) *Datenbanksysteme für Business, Technologie und Web (BTW 2015) - Workshopband*. pp. 203–212. Gesellschaft für Informatik e.V., Bonn (2015)

18. Löffler, R., Wittern, H.: Markenwahrnehmung und marken-differenzierung im zeitalter des web 2.0. In: *Markendifferenzierung*, pp. 359–375. Springer (2011)
19. Manning, C.D., Raghavan, P., Schütze, H.: *Introduction to Information Retrieval*, p. 32. Cambridge University Press, New York, NY, USA (2008)
20. Miller, G.A.: Wordnet: a lexical database for english. *Communications of the ACM* **38**(11), 39–41 (1995)
21. Mohtarami, M., Amiri, H., Lan, M., Tran, T.P., Tan, C.L.: Sense sentiment similarity: An analysis. In: *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*. pp. 1706–1712. AAAI'12, AAAI Press (2012), <http://dl.acm.org/citation.cfm?id=2900929.2900970>
22. Mueller, J., Stumme, G.: Predicting rising follower counts on twitter using profile information. *CoRR* **abs/1705.03214** (2017), <http://arxiv.org/abs/1705.03214>
23. Munoz, T., Kumar, S.: Brand metrics: Gauging and linking brands with business performance. *Journal of Brand Management* **11**(5), 381–387 (2004)
24. Nagarajan, M., Purohit, H., Sheth, A.P.: A qualitative examination of topical tweet and retweet practices. In: *ICWSM* (2010)
25. Nicholls, C., Song, F.: Improving sentiment analysis with part-of-speech weighting. vol. 3, pp. 1592 – 1597 (08 2009)
26. Pak, A., Paroubek, P.: Twitter as a corpus for sentiment analysis and opinion mining. In: *LREC* (2010)
27. Remus, R., Quasthoff, U., Heyer, G.: Sentiws - a publicly available german-language resource for sentiment analysis. In: *LREC* (2010)
28. Schweidel, D.A., Moe, W.W., Boudreaux, C.: Social media intelligence: Measuring brand sentiment from online conversations (2011)
29. Steinskog, A., Therkelsen, J., Gambäck, B.: Twitter topic modeling by tweet aggregation. In: *Proceedings of the 21st Nordic Conference on Computational Linguistics*. pp. 77–86. Association for Computational Linguistics (2017), <http://aclweb.org/anthology/W17-0210>
30. Wang, X., Wei, F., Liu, X., Zhou, M., Zhang, M.: Topic sentiment analysis in twitter: a graph-based hashtag sentiment classification approach. In: *CIKM* (2011)
31. Wang, Y., Liu, J., Qu, J., Huang, Y., Chen, J., Feng, X.: Hashtag graph based topic model for tweet mining. In: *2014 IEEE International Conference on Data Mining*. pp. 1025–1030 (Dec 2014). <https://doi.org/10.1109/ICDM.2014.60>
32. Webber, F.D.S.: Semantic folding theory and its application in semantic fingerprinting. *CoRR* **abs/1511.08855** (2015), <http://arxiv.org/abs/1511.08855>
33. Westpfahl, S., Schmidt, T., Jonietz, J., Borlinghaus, A.: *Stts 2.0. guidelines fuer die annotation von pos-tags fuer transkripte gesprochener sprache in anlehnung an das stuttgart tuebingen tagset (stts)* (2017)

Substitution der Akteur-Beteiligung durch KI und BI am Beispiel eines Logistik-Projekts in den Neuss- Düsseldorfer Häfen

Claus Brell ^[0000-0001-8436-1994], Ralf Kuron and Wilhelm Mülder

Niederrhein University of Applied Sciences, Institut GEMIT,
Richard-Wagner-Straße 140, 41065 Mönchengladbach, Germany

claus.brell@hs-niederrhein.de
ralf.kuron@hs-niederrhein.de
wilhelm.muelder@hs-niederrhein.de

Abstract. Die Verkehrssituation im Gebiet der Neuss-Düsseldorfer Häfen soll im Rahmen des Forschungsprojekts logistiCS durch das Gladbacher Crowd Solving Konzept verbessert werden. Ursprüngliche Idee war, dass alle Akteure (Hafenanrainer, Schiffsverkehr, Logistik-Unternehmen/LKW, Hafen-Eisenbahn) Daten über aktuelle und geplante Verkehrsbewegungen für eine BI-Lösung liefern. Zusätzlich relevante, allgemein verfügbare, aber nicht an einer Stelle gebündelte Daten dienen zur Anreicherung einer Informationsdrehscheibe. Jeder Akteur kann dann entscheiden, ob er sein Verkehrsverhalten aufgrund der erhaltenen Informationen modifiziert. Letztlich wird hierdurch eine deutliche Reduktion von Verkehrsspitzen erwartet. Im Projektverlauf waren nur wenige Akteure bereit, ihre Daten über Schnittstellen bereitzustellen. Es mussten daher andere relevante Datenquellen erschlossen werden. Sensoren wurden zur aktuellen Erfassung der Parksituation installiert. Aktuelle Verkehrsdaten werden über eine neuentwickelte KI-basierte Videobeobachtung erfasst. Um Datenschutz-Probleme zu vermeiden, werden die Videodaten weder übertragen noch gespeichert, sondern unmittelbar von einem in der Kamera implementierten Neuronalen Netz verarbeitet.

Keywords: Verkehrsoptimierung, Crowd Solving, KI-Kamera, Edge-Computing, Informationsdrehscheibe

1 Projektbeschreibung

1.1 Ausgangssituation

Die Neuss-Düsseldorfer Häfen sind wichtige Verkehrsknotenpunkte am Niederrhein. Zentral gelegen, können sie in der Fläche nicht wachsen. Daher werden intelligente Konzepte benötigt, um die steigende Logistikintensität in Zukunft bewältigen zu

Copyright © 2020 by the paper's authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

können. Eine Vorstudie ergab: In Spitzenzeiten führen LKW-Staus zu hohen Zeitverlusten, Zeitfenster zur Belieferung oder Abholung sind schwer kalkulierbar. Es gibt lediglich begrenzte Parkflächen für LKW und PKW.

1.2 Ziele

Ziel des Projektes ist, die Verkehrssituation in den Neuss-Düsseldorfer Häfen zu verbessern und so eine Entspannung des Gesamtverkehrs im Umfeld zu erreichen. Das Vorhaben adressiert neben Innovationen für die Logistik auch soziale und ökologische Nachhaltigkeitsaspekte. Das Projekt wird im Zeitraum 2017-2020 aus dem Europäischen Fond für regionale Entwicklung (EFRE) im NRW-Leitmarkt Logistik gefördert. Primärer Hintergrund des Projektes ist Wirtschaftsförderung.

1.3 Methodisches Vorgehen

In Phase 1 (Fig. 1) wurden die spezifischen Anforderungen der Akteure – Unternehmen im Hafen und LKW-Fahrer - ermittelt. Methodisch wurden zunächst Problembereiche in Fokusgruppen identifiziert und folgend in Befragungen der Akteure konkretisiert. In einer frühen Projektphase wurde den Akteuren ein Prototyp der „Informationsdrehseibe“ an die Hand gegeben, welcher auf Basis des Feedbacks kontinuierlich iterativ optimiert und erweitert wurde.

Aufgrund der Erkenntnis, dass die aktive Beteiligung der Projektpartner im Projektverlauf hinter den Erwartungen zurückblieb [Kuron&Brell 2018], musste das handlungsleitende Gladbacher Crowd Solving Konzept modifiziert werden (Aktivität „Konzeptanpassung“ in Fig. 1).

Die Bereitschaft zur Datenbereitstellung war aus Wettbewerbsgründen, der befürchteten Transparenz und aus technischen Gründen gering. Deswegen mussten andere Datenquellen gefunden werden. Anstatt der Logistik-Daten der Akteure werden nun in Phase 2 Abschätzungen aus einer KI-basierten Videobeobachtung der Verkehrssituation in den Neuss-Düsseldorfer Häfen verwendet [BMVI 2018, Kuron 2020].

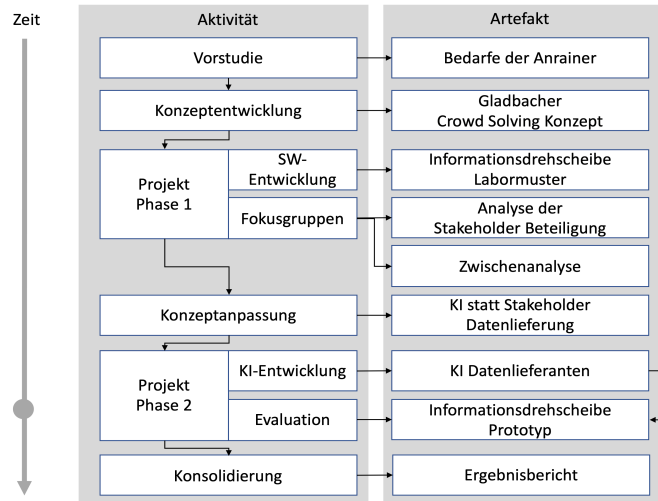


Fig. 1. Methodischer Gang des Projektes im Zeitablauf

1.4 Lösungsidee: Das Gladbacher Crowd Solving Konzept

Crowd Solving ist eine Untermenge von Crowdsourcing mit Fokus auf komplexen, gemeinschaftlichen Problemlösungen [Geiger&Fiehl 2011]. Für Fragestellungen in der Wirtschaftsinformatik wurde Crowd Solving zu einem generischen Ansatz zur Verkehrsoptimierung für logistikintensive Gebiete weiterentwickelt (Fig. 2). Das Gladbacher Crowd Solving Konzept beruht darauf, dass viele Akteure in einem logistikintensiven Gebiet ihre Logistik-Daten transparent machen und alle Akteure Zugang zu diesen Informationen erhalten. Weiterhin werden die Informationen mit weiteren relevanten Daten, die allgemein verfügbar aber nicht an einer Stelle gebündelt sind, angereichert. Damit lässt sich das Gladbacher Crowd Solving Konzept als Business-Intelligence-Ansatz verstehen [Müller&Lenz 2013]. Nun kann jeder Akteur entscheiden, ob er sein Verhalten aufgrund der erhaltenen Informationen modifiziert und diese Verhaltensänderung wiederum bekannt macht. Dadurch, dass alle Akteure nun ihr Verhalten an den verfügbaren Informationen ausrichten, wird eine deutliche Reduktion von Verkehrsspitzen erwartet. Als zentrale Stelle für den Abruf aller Informationen wurde eine Informationsdrehscheibe in Form eines Internet-Portals geplant.

Um die aktive Teilnahme und die Motivation [McClelland 1988] der Akteure am Gladbacher Crowd Solving Konzept zu erhöhen, werden Gamification-Ansätze [Herger 2014, Brell 2018] in das Konzept integriert. Die verwendeten Spiel-Design-Elemente [Deterding 2011] sind direktes Feedback, Schaffung von Gemeinschaften, mittelbare wirtschaftliche Vorteile [Schell 2016].

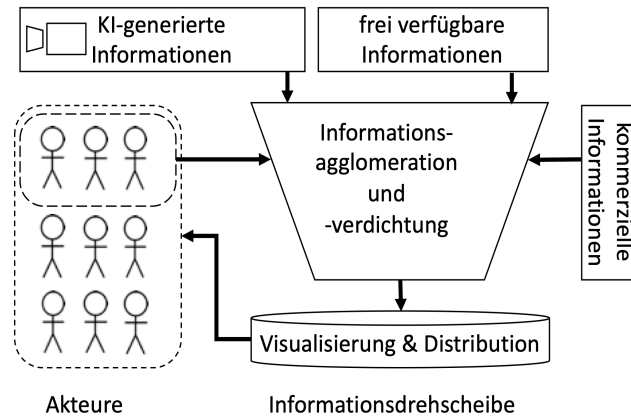


Fig. 2. Das Gladbacher Crowd Solving Konzept

2 Realisierung

2.1 Informationsdreh Scheibe

Eine Informationsdreh Scheibe für die Akteure im Hafen ist als Internetseite implementiert (<https://hafenneuss.de/>) und bündelt alle relevanten Informationen zum Hafengebiet und seiner Nutzung.

Die Informationen umfassen:

- Die aktuelle Verkehrssituation inklusive Prognosen aus INRIX-Verkehrsdaten.
- Bereitstellung von hafeneigenen Verkehrsdaten aus KI-Kameras.
- Bewegungen der Hafeneisenbahn und mögliche Beeinträchtigung des Straßenverkehrs. Statistische Auswertung der Bahnübergangs-Schließungszeiten werden ebenfalls durch Daten der KI-Kameras angereichert.
- Parkplatzsituation mittels Parksensoren.
- Luftqualität inklusive Prognosen durch Zugriff auf Modelldaten des Rheinischen Instituts für Umweltforschung an der Universität zu Köln (EURAD).
- Online-ÖPNV-Daten durch Zugriff auf die Daten der Deutschen Bahn.
- Sicherheitsmeldungen der zuständigen Hafenmeister.
- Alarmierungen durch Einbindung der Wetterwarnungen des Deutschen Wetterdienstes.

2.2 KI-Kameras

Bereits in den Befragungen und in der ersten Nutzung der Informationsdreh Scheibe zeigte sich, dass die Bereitschaft der Akteure, Logistikdaten bereitzustellen, geringer ist als zu Projektbeginn eingeschätzt [vgl. Nielsen 2006]. Daher wurde eine Lösung geschaffen basierend auf eigenentwickelten KI-Kameras, die mittels Videoaufnahmen

das Verkehrsgeschehen erkennen, analysieren und daraus Daten für die Informationsdrehzscheibe gewinnen.

Um datenschutzrechtliche Probleme zu vermeiden, werden die Videodaten weder übertragen noch gespeichert, sondern unmittelbar von einem in der Kamera implementierten Neuronalen Netz verarbeitet (Edge-Computing). Das Netz wurde auf die im Hafen relevanten Objekttypen wie PKW, Lkw und Züge trainiert. Die Objekte werden gezählt und ihre Positionen und Geschwindigkeiten bestimmt. Damit werden lediglich anonyme Daten zur Verkehrslage über das Internet übertragen.

Für die KI-Kameras wurden Nvidia-Komponenten (Jetson TX2/XAVIER) sowie spezielle Low-Light Kameras verwendet. Mit einer Bildrate von 30 fps sind Echtzeitanalysen möglich.

3 Projektstand und Ausblick

Das Projekt ist in der Phase Konsolidierung angelangt (Fig. 1). Die Informationsdrehzscheibe ist seit zwei Jahren online, die KI-Kameras liefern erste Verkehrsdaten. Für die Zukunft können in einem Gebiet mehrere KI-Kameras aufgestellt werden mit dem Ziel einer flächendeckenden Verkehrsanalyse- und Modellierung. Die Informationsdrehzscheibe wurde in diesem Forschungsprojekt zwar für die Neuss-Düsseldorfer Häfen realisiert, das Konzept ist aber auf jede Art von logistikintensiven Gebieten übertragbar. Inwieweit eine Reduktion des Verkehrsaufkommens in anderen Gebieten mit hoher Effektstärke möglich ist, soll in Folgeprojekten untersucht werden.

Die Informationsdrehzscheibe für die Neuss-Düsseldorfer Häfen wird nach Projektende durch den Wirtschaftspartner weiter betrieben. Schon während der Projektlaufzeit haben sich technische Möglichkeiten (kleinere, preiswertere und leistungsfähigere KI-Kameras, neue Framework-Versionen) verbessert, die Informationsdrehzscheibe wird in Zukunft kontinuierlich an die technische Entwicklung angepasst.

References

1. BMVI (Bundesministerium für Verkehr und digitale Infrastruktur): Digitalisierung und Künstliche Intelligenz in der Mobilität. Online-Ressource: https://www.bmvi.de/SharedDocs/DE/Anlage/DG/aktionsplan-ki.pdf?__blob=publicationFile (2018)
2. Brell, C. Wie Gamification den Methodenapparat der Wirtschaftsinformatik bereichert. In: Informatik Aktuell. Frechen. Online Ressource <https://www.informatik-aktuell.de/management-und-recht/projektmanagement/wie-gamification-den-methodenapparat-der-wirtschaftsinformatik-bereichert.html> (2018):
3. Deterding S, Dixon D, Khaled R, Nacke L.: From game design elements to gamefulness: defining gamification. In: Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments. S. 9-11. doi: 10.1145/2181037.2181040 (2011)
4. Geiger D, Rosemann M, Fiel E.: Crowdsourcing information systems: a systems theory perspective. In: Proceedings of the 22nd Australasian Conference on Information Systems (ACIS), online-Ressource <https://pdfs.semanticscholar.org/3e4c/19f558862a92c8e7485758b5809a0b8338db.pdf>. (2011)
5. Herger, M.: Enterprise Gamification – Engaging People by letting them have fun. Book 1 of 6. Leipzig. (2014)
6. Kuron, R.; Brell, C.: Informationen verfügbar machen. In: Hafenzeitung 02/2018. Düsseldorf. S. 3. online Ressource http://hafenzeitung.de/downloads/archiv/pdf/hafenzeitung_2018_02.pdf (2018)
7. Kuron, R.: Mit moderner IT gegen zunehmende Verkehrsprobleme. Städte- und Gemeinderat (05), S. 23-25 (2020).
8. McClelland, D.: Human Motivation. Cambridge: Cambridge University Press. doi:10.1017/CBO9781139878289 (1988)
9. Müller, R.M.; Lenz, H.-J.: Business Intelligence. Heidelberg. (2013)
10. Nielsen, J.: Participation Inequality: Encouraging More Users to Contribute. In: Nielsen Norman Group. Online-Ressource: <https://www.nngroup.com/articles/participation-inequality/> (2006)
11. Schell, J.: Die Kunst des Game-Designs – Bessere Games konzipieren und entwickeln. 2. Auflage. Frechen. (2016)

FGDB Workshop

Discovery of Ontologies from Implicit User Knowledge

David Haller¹[0000-0001-5287-7187] and Richard Lenz¹[0000-0003-1551-4824]

Chair of Computer Science 6 (Data Management)
University of Erlangen
david.haller@fau.de
richard.lenz@fau.de
<https://www.cs6.tf.fau.eu>

Abstract. The purpose of the Semantic Web is to enable worldwide access to humanity’s knowledge in a machine-processable way. A major obstacle to this has been that knowledge is often either represented in an incoherent way, or not externalized at all and only present in people’s minds. Populating a knowledge graph and manually building an ontology by a domain expert is tedious work, requiring great initial effort until the result can be used. As a consequence, knowledge will often never be made available to the Semantic Web. The aim of this project is to develop a new approach for building ontologies from implicit user knowledge that is already present, but hidden in various artifacts like SQL query logs or application usage patterns.

Keywords: Semantic Web · Knowledge Graph · Schema Inference · Query-Driven · Data Integration

1 Introduction

In the last decades, the World Wide Web indisputably changed human society and economy. Computers, paradoxically, although essentially operating the Web, cannot make use of it on their own. Knowledge on the Web is represented mostly in a way suitable for humans, as pages containing plain text and graphics. Although web pages can be structured hierarchically and linked with each other, their inherent semantics are only accessible by a human being perceiving the content. Querying the Web is usually restricted to simple keyword-based search engines or web services with proprietary APIs. Apart from these technical obstacles, the Web also does not define coherent sets of terms that shall be used to describe concepts and entities of a particular domain, leaving that tasks to human interpretation.

The Semantic Web [1] offers a framework for machines to make use of this knowledge. Instead of storing linked HTML documents, the Semantic Web links

Copyright © 2020 by the paper’s authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

facts with each other. This is done using the Resource Description Format (RDF), which operates on a graph-based data model. The graph can then be queried using SPARQL, the default RDF query language, which has a similar expressivity as SQL has for relational databases.

For being able to actually interpret RDF data, an ontology must be defined. This can be done either in RDF Schema or in the Web Ontology Language (OWL). In a nutshell, an ontology is a set of axioms which constrain what statements can or cannot be true and allows to deduce new statements from existing statements. Creating these ontologies manually is tedious work and therefore a blocker for Semantic Web adoption.

Knowledge already exists somewhere, either in people's minds or in various kinds of artifacts: semi-structured file formats like CSV or JSON, plain text in natural language, applications source code, log files, or SQL queries. Transferring all this knowledge by hand into a graph is time-consuming and expensive, wherefore this can be applied only for limited use cases. Developing an at least partly automated method to perform this task could drastically lower the costs for deploying Semantic Web techniques.

2 Past Research

This PhD research project will extend the scope of the previous master thesis project PHAROS, which results have been published in a followup research paper [6]. The focus of PHAROS was to improve the understanding of heterogeneous data sources within a data lake by analyzing SQL query logs accessing these sources and by extracting knowledge fragments from those queries in order to gain insights about the underlying schema. This may seem unintuitive at first glance, as SQL is usually associated with relational databases, where schemata are already known, but SQL has evolved into a general query language for heterogeneous data sources. When a data scientist encounters an unknown data source, he needs a great deal of cognitive effort to understand its semantics prior to writing the queries that use the data sources for analytics. Therefore, each query implicitly contains hidden knowledge and assumptions about the data and can be seen as a partial schema definition.

For example, joining two tables over an attribute indicates that the data analyst probably identified a foreign-key-relationship, otherwise he would not have made that join. Renaming columns with speaking names or explicit type casts give hints about their meaning.

```
select sum(p.salary), dep.id, dep.name
from person p join department dep
on p.dep_id = dep.id
where dep.location='DE' or dep.location = 'FR'
group by dep.id, dep.name
order by dep.name;
```

Listing 1.1. Example query with partial schema information

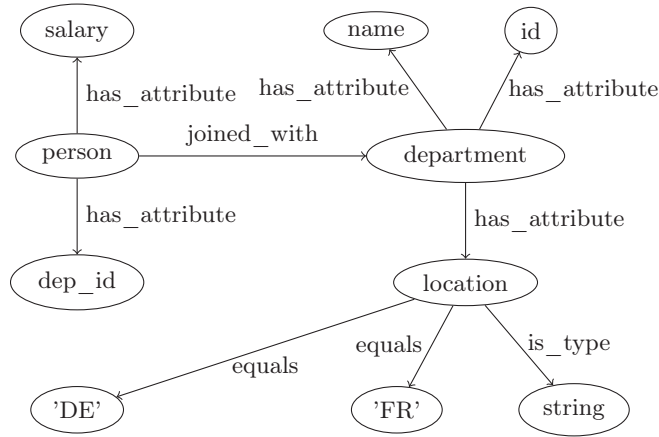


Fig. 1. Partial schema information stored in a knowledge graph

We had decided to build a knowledge graph from SQL query logs that could be used to help understanding the mental model behind data sources. A prototype was written that demonstrates the feasibility of the approach. It was implemented in form of a JDBC proxy driver that can capture and analyze all SQL queries a Java application sends to a SQL query engine like Apache Drill, allowing a minimal-invasive deployment of the prototype into existing workflows, as it is compatible with any software using JDBC drivers. The prototype was evaluated using a test database with a known schema and a set of test queries, based on exercises of our introductory database lecture.

3 Research Objectives

The resulting knowledge graph describes how data sources have been used by analysts, but does not describe the semantics of the data sources themselves, like value constraints or foreign key relationships. Human interpretation of the results is required to gain insights about the used data sources. Therefore, the next level will be to perform automatic reasoning on top of the knowledge graph. This requires to generate an (incomplete) ontology for describing the semantics of data sources. As knowledge derived from SQL queries may be contradictory - when the query log contains queries that are not conforming to the underlying schema - an approximate approach is needed to deal with this ambiguity.

Queries do not reflect the semantics of a data source, but the mental model a data scientist has made of it. Analyzing these mental models can already give valuable insights. For example, if someone uses a “grade” attribute and compares it with values that are not present in the dataset, the explanation could be that the user originated from a country with a different school grading system. There are multiple concepts out there about what a “grade” should be,

a query-driven approach could provide more transparency about these concepts as an intermediate step.

When analyzing SQL query logs, there will always be queries that are based on wrong assumptions about the schema, especially if the origin of the query log is from an interactive session where queries with undesired results may be rewritten. With multiple query logs from different sessions and users, finding the similarities in their behavior and their mental model could lead to the intended semantics of the used data sources.

A self-learning system shall be developed that makes suggestions to data scientists about suitable sources or queries they may find helpful for their task. Based on their given feedback and performed queries, the system shall incrementally approximate the true semantics behind the data sources.

Thus far, only SQL query logs were considered as a source for query-driven schema inference. But there are other types of queries to consider, like query strings from search engines, application usage patterns extracted from graphical analysis tools or even source code from programs accessing a data source. Other query languages like XQuery or languages from various NoSQL database systems could be included. The approach does not depend on a specific language.

4 Related Work

Many approaches for schema inference are *data-driven*, using data profiling methods to reconstruct the underlying schema of a given dataset. A significant example is the Metanome project [9], which provides an extensible framework offering various algorithms, for example to discover functional dependencies [7]. Datatype-based schema inference for JSON datasets is demonstrated in [2] and [3]. In [8] it is shown how to identify the domains the values of a column come from. The Datamaran project [5] aims to discover structure in text files like applications logs and transforming them into normalized relational tables. The ESKAPE platform [11] allows users to assign instances to semantic models [10]. A general overview of dataset search and integration techniques is given in [4].

5 Evaluation Approach

The existing prototype will be extended to use Semantic Web reasoning techniques to deduce the meaning of a data source by the knowledge extracted from query fragments. A framework of rules should be defined to achieve this, possibly with the Rule Interchange Format (RIF). This prototype shall then be tested on real world query logs, so the resulting knowledge graph can be compared with the actual schema the data sources are based on. A supplementary user study will show if the software is able to enhance the workflow of data scientists to understand heterogeneous data sources.

References

1. A Semantic Web Primer. Cooperative Information Systems, MIT Press, Cambridge, Mass, 3rd ed edn. (2012)
2. Baazizi, M.A., Ben Lahmar, H., Colazzo, D., Ghelli, G., Sartiani, C.: Schema inference for massive JSON datasets. In: Proceedings of the 20th International Conference on Extending Database Technology. pp. 222–233. Venice, Italy (2017)
3. Baazizi, M.A., Colazzo, D., Ghelli, G., Sartiani, C.: Parametric schema inference for massive JSON datasets. *The VLDB Journal* **28**(4), 497–521 (Aug 2019)
4. Chapman, A., Simperl, E., Koesten, L., Konstantinidis, G., Ibáñez, L.D., Kacprzak, E., Groth, P.: Dataset search: A survey. *The VLDB Journal* **29**(1), 251–272 (Jan 2020)
5. Gao, Y., Huang, S., Parameswaran, A.: Navigating the Data Lake with DATA-MARAN: Automatically Extracting Structure from Log Datasets. In: Proceedings of the 2018 International Conference on Management of Data - SIGMOD '18. pp. 943–958. ACM Press, Houston, TX, USA (2018)
6. Haller, D., Lenz, R.: Pharos: Query-Driven Schema Inference for the Semantic Web. In: Machine Learning and Knowledge Discovery in Databases, vol. 1168, pp. 112–124. Springer International Publishing, Cham (2020)
7. Jiang, L., Naumann, F.: Holistic primary key and foreign key detection. *J Intell Inf Syst* (Jun 2019)
8. Ota, M., Müller, H., Freire, J., Srivastava, D.: Data-driven domain discovery for structured datasets. *Proc. VLDB Endow.* **13**(7), 953–967 (Mar 2020)
9. Papenbrock, T., Bergmann, T., Finke, M., Zwiener, J., Naumann, F.: Data profiling with metanome. *Proc. VLDB Endow.* **8**(12), 1860–1863 (Aug 2015)
10. Pomp, A., Kraus, V., Poth, L., Meisen, T.: Semantic Concept Recommendation for Continuously Evolving Knowledge Graphs. In: Enterprise Information Systems, vol. 378, pp. 361–385. Springer International Publishing, Cham (2020)
11. Pomp, A., Paulus, A., Jeschke, S., Meisen, T.: ESKAPE: Information Platform for Enabling Semantic Data Processing. In: Proceedings of the 19th International Conference on Enterprise Information Systems. pp. 644–655. SCITEPRESS - Science and Technology Publications, Porto, Portugal (2017)

Sense Tree: Discovery of New Word Senses with Graph-based Scoring

Jan Ehmüller¹, Lasse Kohlmeyer¹, Holly McKee¹, Daniel Paeschke¹,
Tim Repke², Ralf Krestel², and Felix Naumann²

Hasso Plattner Institute, University of Potsdam, Germany

¹`first.last@student.hpi.uni-potsdam.de`, ²`first.last@hpi.uni-potsdam.de`

Abstract. Language is dynamic and constantly evolving: both the usage context and the meaning of words change over time. Identifying words that acquired new meanings and the point in time at which new *word senses* emerged is elementary for *word sense disambiguation* and *entity linking* in historical texts. For example, *cloud* once stood mostly for the weather phenomenon and only recently gained the new sense of *cloud computing*. We propose a clustering-based approach that computes *sense trees*, showing how meanings of words change over time. The produced results are easy to interpret and explain using a drill down mechanism. We evaluate our approach qualitatively on the Corpus of Historic American English (COHA), which spans two hundred years.

1 The Evolution of Language

As language evolves, words develop new senses. Detecting these new senses over time is useful for several uses, such as *word sense disambiguation* and *entity linking* in historic texts. Current machine learning models that represent and disambiguate word senses or link entities are trained on recent datasets. Therefore, these models are not aware of how language changed over the last decades or even centuries. When disambiguating words in historic texts, it is not possible for such a model to know which senses a word could have had at that time. For instance, the word *cloud* was once used mostly in the newspaper section of weather forecasts. Nowadays, it is increasingly used in the context of *cloud computing*. Hence, when disambiguating *cloud* in texts from the 19th century with a model trained on current data, the model would not be aware that the meaning of *cloud computing* did not yet exist at that point in time.

New word senses enter colloquial language from many aspects of human life, such as popular culture, new technologies, world events, and social movements. In 2019 alone, over 1,100 new words and meanings were added to the Merriam-Webster Dictionary. Because the usage of context words differs over time, we can discover word senses by taking advantage of this temporal aspect. Discovering new or different senses of a word is known as word sense induction (WSI).

Copyright © 2020 by the paper's authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

An example for WSI can be found in the context of *depression*. *Depression* is commonly known in the sense of a mental health disorder, but also in the sense of economic crisis in the term *great depression*. The aim of WSI is to find out which senses the word *depression* has. Word sense disambiguation (WSD), on the other hand, is known as the automated disambiguation of a word sense within a text [15]. While WSI aims to discover different senses for one word, WSD aims to decide which sense a word has in a specific context, such as in a sentence.

We propose a WSI method that detects new senses by creating multiple co-occurrence graphs over time, and extracts word senses based on so-called ego-networks. For a given word, it uses graph clustering to extract word senses from the word’s ego-network. These word senses are matched over time to create a forest of so-called “sense trees”. This forest can be explored to find out if and when a word gained new senses over time. With the help of linguists, we annotated a list of 112 words for evaluation¹ and tested our approach using the Corpus of Historical American English (COHA), with 400 million words the largest corpus of its type [4].

2 Related Work

Research on word sense induction and word sense disambiguation addresses how to improve information retrieval in search engines, information extraction for specific domains, entity linking over time, machine translation, and lexicography [15]. Our research focuses on WSI and aims to discover the emergence of new word meanings over time. Approaches can be divided roughly into vector space models that include word embeddings [5], graph clustering techniques [19], which include word clustering [5], and co-occurrence graphs [14].

Word embeddings are vector representations of words in a semantic space that allow for word meanings to be induced by context. Word embeddings are trained by a neural network that learns to predict a word based on its context words or vice versa. Hence, words with similar contexts have a similar vector and are thus in close proximity to each other. The initially proposed word2vec model [13] computes only one vector and thus, allows for only one meaning per word. Natural language is ambiguous and most words are polysemous — they have multiple senses. Sense embeddings, as proposed by Song et al. [17], allow for multiple senses by representing each sense instance as its own vector.

However, neither traditional word embeddings, like word2vec, nor sense embeddings consider the temporal aspect and assume that words are static across time. This creates a challenge in the face of dynamic and changing natural language [1]. Word embeddings can be used for temporal tasks if multiple embeddings are trained separately for separate time periods. As those embeddings are trained separately, they do not lie in the same semantic embedding space [10]. To ensure that they are in the same space and thus are comparable, they need to be aligned with each other [1]. To resolve such alignment issues, dynamic or

¹ <https://hpi.de/naumann/s/language-evolution>

diachronic word embeddings were introduced [10]. Kim et al. solve this by training their model on the earliest time periods first. Using the obtained weights as the initial state for the next training phase, they move through subsequent periods, allowing the embeddings to gain complexity and pick up new senses [9]. Yao et al. present another idea, called dynamic word embeddings, where alignment is enforced by simultaneously training the word embeddings for different time periods [21]. The disadvantage of their approach is that it addresses either only the temporal semantic shift or multi-sense aspect of words. In contrast, our approach takes both aspects into consideration.

Another method for extracting word senses is through the use of graph clustering or community detection. This method builds a graph based on co-occurrence features from a corpus. Such graphs represent the relation of words to each other, and allows for the extraction of sense clusters through graph clustering. Automatic word sense change detection based on curvature clustering can help understand the different senses in historic archives [18]. Their manual evaluation chose 23 terms with known sense changes (e.g., “gay”). Hope and Keller introduce the soft clustering algorithm MaxMax [7]. They identify word senses by transforming the co-occurrence graph around a given word into an unweighted, directed graph.

Mitra et al. present an approach that builds graphs based on distributional thesauri for separate time periods [14]. From those graphs they extract so-called “ego-networks”. With the randomized graph-clustering algorithm Chinese Whispers, sense clusters are induced for specific words from their ego-network [2]. A similar and more recent approach by Ustalov et al., performs the clustering step with the meta-algorithm WATSET. This algorithm uses hard clustering algorithms, such as Louvain or Chinese Whispers, to perform a soft clustering [19]. Hard clustering algorithms assign nodes to exactly one cluster, whereas soft clustering produces a probability distribution of cluster assignments.

Besides embeddings and graph models, topic modeling has been applied to WSI as well. Lau et al. model word senses as topics of a word by application of latent Dirichlet allocation (LDA) as well as non-parametric hierarchical Dirichlet processes (HDP). They also applied their approach on the field of novelty sense detection by comparing induced senses of words of a diachronic corpus containing two time periods for a self developed dataset of 10 words [11].

Jatowt and Duh provide a framework for discovering and visualizing semantic changes w.r.t. individual words, word pairs and word sentiment. They use n-gram frequencies, positional information and Latent Semantic Analyses to construct word vectors for each decade of Google Books and COHA. To derive changes of word senses, they calculate the cosine similarity of vector representations of the same word at different time points. The authors show results of a case study for semantic changes of single words, inter-decade word similarity and contrasting word pairs in 16 experiments with mostly different words [8]

Our approach is similar to that of Mitra et al. [14], but differs in that we build a simpler co-occurrence graph and compare three different clustering algorithms. Additionally, our approach interprets the results of comparing sense clusters

automatically to rank a word according to the likelihood of having gained a new sense over time. Furthermore, we use more fine-grained time slices, to more precisely narrow in on the point in time where a new sense emerges, whereas Mitra et al. merely oppose two time slices at once. This point can be visualized by our drill down, which shows the forest of sense trees built from ego-networks. This ability makes our approach valuable as an exploration tool in the field of historical and diachronic linguists, specifically for hypothesis testing.

3 A Forest of Word Senses

Our approach can be divided into the three parts visualized in Figure 1 using a fictitious example: (i) construction of a weighted co-occurrence graph, (ii) extraction of word senses from a word’s ego-network in the form of sense clusters, and (iii) matching them over time to create a forest of sense trees. We separate the corpus into n time slices C_t , which are handled as subcorpora. Similar to Mitra et al. [14], our analysis is limited to nouns. For each subcorpus, we construct a filtered co-occurrence graph and extract sense clusters for each word. Finally, these clusters are connected across time slices to form sense trees.

3.1 Building a Co-Occurrence Graph

To build a weighted co-occurrence graph for time slice C_t , we need to count how often words appear in the same context in C_t . Two nouns co-occur if they are part of the same sentence and the word distance between them is $\leq n_{window}$. Therefore, the parameter n_{window} controls the complexity of the resulting co-occurrence graph. A smaller window size results in fewer co-occurrence edges and hence in a sparser graph. Having computed the co-occurrences, we can create the graph $G_t = (V_t, E_t)$ for each time slice C_t . The sets of nodes and edges of G_t for time slice C_t are defined as

$$\begin{aligned} V_t &= \{u \in C_t \mid u \text{ is noun} \wedge tf(u) \geq \alpha_{tf}\} \\ E_t &= \{\{u, v\} \mid u, v \in V_t \wedge u \neq v \wedge cooc(u, v) \geq \alpha_{cooc}\} \end{aligned} \quad (1)$$

where $tf(u)$ is the term frequency of u in the given time slice and $cooc(u, v)$ is the number of times words u and v appear within the same window. We use the threshold parameters α_{tf} and α_{cooc} to exclude rarely occurring words, as, depending on the window size n_{window} , the number of edges in this graph would rapidly increase. Oftentimes, a corpus has an unbalanced distribution of data across all time slices. As a result, some have much higher raw co-occurrence values. This especially affects frequently occurring words with generic meanings, such as *man* and *time*. Our initial filtering does not account for that. We use the point-wise mutual information (PMI) [20] $pmi(u, v) = cooc(u, v) / (tf(u) * tf(v))$ to reduce the importance of frequent words. After filtering more edges and possible unconnected nodes, the final co-occurrence graph graph $G'_t = (V'_t, E'_t)$ for time slice C_t , as depicted as one column in Figure 1a, is defined by

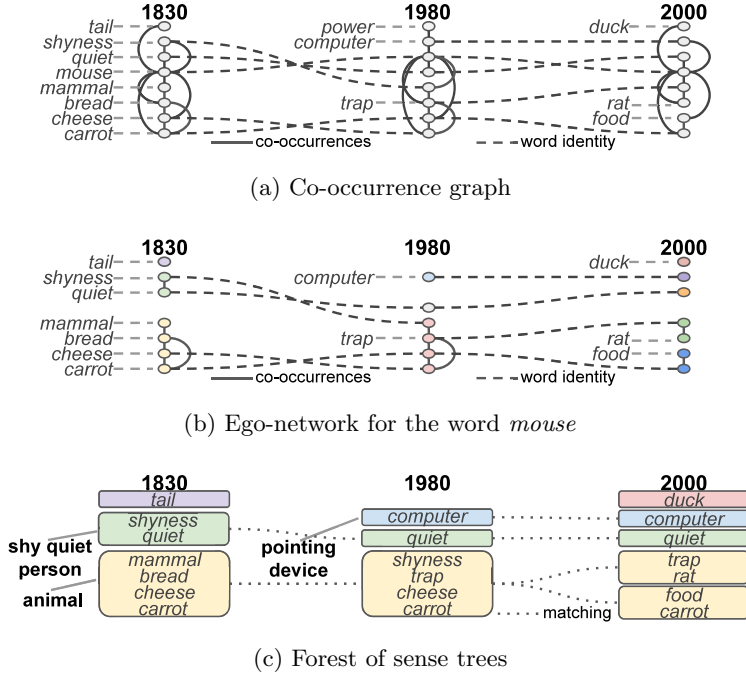


Fig. 1: Example for the construction of a simple sense forest of the word *mouse*

$$V'_t = \bigcup E'_t \quad E'_t = \{\{u, v\} \in E_t \mid pmi(u, v) \geq \alpha_{pmi_t}\} \quad (2)$$

where α_{pmi} is a threshold parameter to remove the most loosely associated words in the given time slice.

3.2 Word Sense Extraction

We use G'_t to extract information about word contexts in time slice C_t . We hypothesize that the context of a word is an indicator for its senses as suggested by Lindén and hence use clustering to extract those senses [12]. The context of word w can be extracted in the form of an ego-network, which contains all neighbors of w and edges among them, but not w itself. Figure 1b shows such a network for the word *mouse*. The different colors indicate the different clusters of a time slice that were produced by a graph-clustering algorithm. Formally, we define the ego-network $\hat{G}_w = (\hat{V}_w, \hat{E}_w)$ of word w in time slice C_t as

$$\hat{V}_w = \{u \in V'_t \mid \{u, w\} \in E'_t\} \quad \hat{E}_w = \{\{u, v\} \in E'_t \mid u \in \hat{V}_w \wedge v \in \hat{V}_w\} \quad (3)$$

To extract different senses of w , we cluster the nodes of its ego-network \hat{G}_w . In Section 4 we compare different clustering strategies. Each of these clustering

algorithms produces a set of p disjoint clusters $S_{w_t} = \{c_1, \dots, c_p\}$ from the ego-network of word w in time slice C_t . Following Mitra et al. [14], we assume that each of the resulting clusters represents a “sense cluster”. Ideally, each meaning of w in C_t is represented by exactly one sense cluster. By relaxing this condition and allowing more than one sense cluster for each sense, we are able to get better and more fine-grained results by the clustering algorithms.

3.3 Matching Word Senses Over Time

In the last step, sense clusters are matched across time slices. We use the Jaccard similarity to compare sets of words, which are given by the clusters of word w from two neighboring time slices C_t and C_{t-1} . Let $c_i \in S_{w_t}$ be a sense cluster for word w in time slice C_t . After a pairwise comparison between all sense clusters across two time slices, we use a greedy approach to iteratively match those with the highest score. However, if there is no cluster $c'_j \in S_{w_{t-1}}$ that shares any words with $c_i \in S_{w_t}$, it remains unmatched. In this case, it would become the root cluster of a new sense tree.

A disadvantage of this matching strategy is that a word sense might simply not occur in some time slices and thus interrupt the lineage of that word sense. This may happen with sense clusters whose words have low frequencies, such as words that appear in specific scientific literature. Because they cannot always be matched between neighboring time slices, we match them across a longer time span. Sense clusters in S_{w_t} are matched not only to the sense clusters in $S_{w_{t-1}}$, but also to any other previous sense cluster in $S_{w_1}, \dots, S_{w_{t-2}}$ that remained unmatched. We call this matching strategy leaf-matching.

$$S'_{w_{t-1}} = S_{w_{t-1}} \cup \bigcup_{i=1}^{t-2} \{c' \in S_{w_i} \mid c' \text{ not matched to any } c'' \in \bigcup_{j=i+1}^{t-1} S'_{w_j}\} \quad (4)$$

We compare for each time slice and produce a forest of sense trees $F_w = (V_w, E_w)$:

$$V_w = \bigcup_{i=1}^n S_{w_i}; \quad E_w = \{(c, c') \mid c \in S_{w_i} \wedge c' \in S_{w_j} \wedge i < j \wedge c \text{ matched } c'\} \quad (5)$$

F_w contains sense clusters without incoming edges. These clusters are root clusters and represent the beginning of a sense tree. Given a root cluster r , the respective sense tree $F_{w_r} = (V_{w_r}, E_{w_r})$ is defined as follows:

$$\begin{aligned} V_{w_r} &= \{c \in V_w \mid \exists p = ((r, c'_1), \dots, (c'_k, c))\} \\ E_{w_r} &= \{(c, c') \in E_w \mid c, c' \in V_{w_r}\} \end{aligned} \quad (6)$$

Such a sense tree represents a distinct sense of the word w . For example, in Figure 1c, the matched clusters make up three sense trees, referring to three different meanings of the word *mouse*.

4 Comparison of Algorithms for Sense Clustering

In this section we introduce the following algorithms for the clustering step in Section 3.2: Chinese Whispers [2], Girvan-Newman [6], and Louvain [3]. We also describe the insights gained by setting up experiments using our drill down to inspect the resulting clusters of these algorithms.

Chinese Whispers is an agglomerative clustering algorithm introduced by Biemann [2]. It does not have a fixed number of clusters, which is a property that suits word senses whose number is not known apriori. Its only parameter is the number of iterations. Depending on the size of the graph, a small number of iterations might never produce larger sense clusters. We chose 1 000 iterations to ensure that the algorithm converges as suggested by Biemann [2].

Inspecting the computed clusters, we discovered that Chinese Whispers produces one very large sense cluster that contains nearly every word in the ego-network, along with very few other small sense clusters. Since the large sense cluster contains more than one meaning, the results are not fitting our use case.

Girvan-Newman is a community detection algorithm named after its authors [6]. The algorithm is a hierarchical method that iteratively removes edges from the graph to find communities. It always removes the edge with the highest betweenness centrality or a custom metric, such as the co-occurrence frequency. Hence, the initially computed sense cluster contains all nodes of the ego-network. With each iteration, it produces more detailed sense clusters. This algorithm produces also a variable number of clusters and hence fits well for the task of identifying an unknown number of word senses. However, in our configuration, the computational costs of Girvan-Newman are around 25-30 times higher than those of Louvain and Chinese Whispers. Due to this high computational cost, we choose only three iterations. Because the fine-granularity of the hierarchy depends on the number of iterations, Girvan-Newman effectively produces one large cluster that contains most words of an ego-network in addition to many one-node sense clusters. As with Chinese Whispers, these results do not fit the use case of identifying multiple meanings of a word.

Louvain is a hierarchical community detection algorithm proposed by Blondel et al. [3]. It uses the Louvain modularity, which measures the difference of edge density inside communities to the edge density outside communities, to identify communities in a graph. Similarly to the previous algorithms, its number of detected clusters is not fixed. From the computed hierarchy it greedily chooses the graph partition that optimizes the algorithm's modularity measure. While investigating the computed sense clusters, we found that this partitioning produced a fairly balanced amount of sense clusters in terms of the cluster size. The results are applicable for our use case, since in most cases, sense clusters can be assigned to exactly one meaning of a word. However, in the later time slices with an increasing number of documents and thus co-occurrences, the computed sense clusters are not partitioned as well. In some cases it produces quite large sense clusters that contain multiple meanings of a word.

5 Evaluation

The evaluation of WSI approaches is an open challenge, as there are no standardized experimental settings or gold standards. Kutozov et al. address the need for standardized, robust test sets of semantic shifts in their 2018 survey [10]. Other researchers created manually selected word lists, which can vary widely and make it difficult to compare the accuracy of approaches [14,21]. In this section, we introduce our evaluation data, which we compiled by aggregating the different approaches used in related work and annotated with the help of expert linguists. We also discuss the impact of the hyperparameters of our approach and demonstrate the effectiveness qualitatively.

5.1 Compiled Word List for Evaluation

The word list compiled by Yarowsky [22] is used in several publications on word sense disambiguation [16]. Yarowsky developed an algorithm that disambiguates the following twelve words with two distinct meanings: *axes*, *bass*, *crane*, *drug*, *duty*, *motion*, *palm*, *plant*, *poach*, *sake*, *space*, and *tank*. We extended this list by adding new dictionary entries, novel technical terms and words based on related work [15]. It also contains words that did not gain new meanings in the last 200 years. The words were annotated with respect to new sense gain using word entries from the Oxford English and Merriam-Webster Dictionaries, which include references to the first known use of words with a particular meaning. WordNet², a commonly used lexical resource for computational linguistics, groups words with regard to word form as well as meaning. To merge the result obtained from the two different approaches we use a logical OR-operation. Thus, a word is considered to have gained a new sense if it is labeled positively from either of our two approaches.

In total we compiled a set of 112 words with either a single sense, multiple, but stable senses, or words that gained at least one new sense. Each candidate word was annotated by 15 linguists (C1 and C2 level) as “*Gained an additional sense since 1800*” or “*No additional sense since 1800*”. We measure an overall agreement of 61% and a free-marginal Fleiss kappa of 0.42 (fixed-marginal: 0.22). Based on a simple majority vote, 42 words gained an additional sense, whereas 70 did not. For some words, such as *android*, *beef*, or *pot*, we saw a high annotator agreement over 75%. When using this as a threshold, 14 words gained an additional sense, whereas 31 did not. The linguists were particularly undecided on the words *cat*, *honey*, *power*, and *state*. The annotations are on our website.[1]

5.2 Corpus of Historical American English (COHA)

For demonstrating our proposed approach, we use the Corpus of Historical American English (COHA)³. It is one of the largest temporal corpora over one of the

² <https://wordnet.princeton.edu/>

³ <https://www.english-corpora.org/coha/>

Table 1: Comparison of graph clustering algorithms

Algorithm	Precision@5	Precision@10	Precision@20
Chinese Whispers	0.4	0.4	0.35
Girvan-Newman	0.2	0.5	0.55
Louvain	0.6	0.6	0.4

longest time ranges [4]. COHA spans texts from the years 1810 to 2009 and contains more than 100,000 single texts in fiction, popular magazines, newspapers and non-fiction books with a total of 400 million words.

We split the corpus by decade to generate the time slices, since vastly different vocabularies would result in different word contexts. Ideally, the vocabulary and word distribution is relatively stable across all time slices. Since the number of tokens increases with each decade, we measure the vocabulary overlap. In our measurements (not shown due space constraints), neighboring decades share 20-30% of their vocabulary, while the decades that are further away from each other share only 5-15% of their vocabulary for both measures. Very high values produced by the cosine similarity highlight that frequently occurring words appear in most decades. We can conclude that using a frequency- and co-occurrence-based approach to extract information about changes of word senses over time is feasible. For a deeper statistical analysis of the corpus, we refer readers to the work by Jatowt et al. [8]

5.3 Hyperparameter Evaluation

For evaluation, we derive a score of how likely it is, that a word gained a new sense over time. Therefore we count the number of sense trees for a word and their distribution over time. Sense trees with a length of 1 are ignored. This count is used to rank words, such that word that gained a new sense are at the top. Using our annotated data we calculate the precision@ k with $k \in \{5, 10, 20\}$.

The following parameter settings were found by optimizing our approach on the annotated word list. We set the pointwise mutual information (PMI) threshold parameter $\alpha_{pmi} = 0.01$, we used the Jaccard similarity as similarity measure, and we used leaf-matching to match sense clusters across time slices.

We compare the three graph clustering algorithms introduced in Section 3: Chinese Whispers, Girvan-Newman, and Louvain. Table 1 presents the results with these algorithms. The highest precision values are highlighted bold.

Louvain outperforms the other two algorithms at $k = 5$ and $k = 10$. Girvan-Newman has a much lower precision at $k = 5$, but performs much better at the values 10 and 20 for k . Chinese Whispers does not perform as well. We suggest Louvain as clustering algorithm, because it fits best for our use case that a sense cluster should only represent a single meaning.

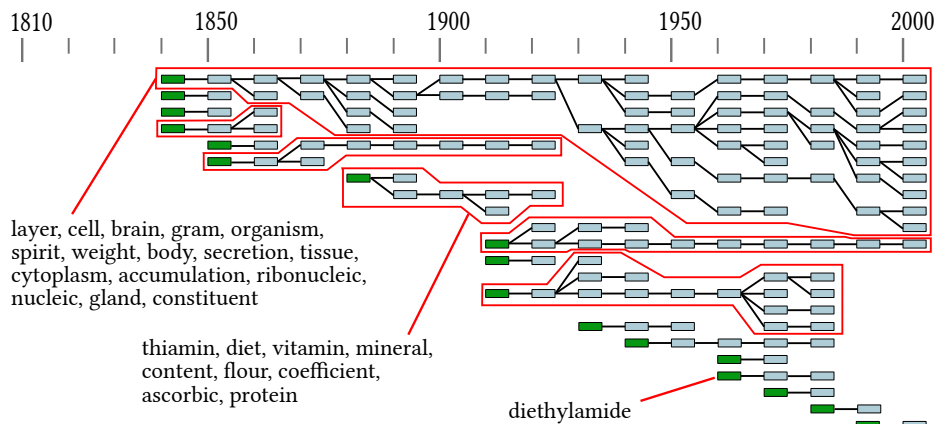


Fig. 2: Sense tree for the word *acid*. Green boxes represent the initial cluster for a sense, matched clusters are connected by straight lines.

5.4 Qualitative Evaluation

To qualitatively evaluate our approach, we take a closer look at the drill down example for the word *monitor*. Our approach produced five sense trees, each of which can be matched to a single meaning of the word: *hall monitor* starting in the 1830s, the warship of that name starting in the 1860s (two sensetrees), *monitor* in the technological sense of a screen starting in the 1920s, and the surveillance sense starting in the 1960s. The time periods at which new senses emerged are accurate. For example, the warship was created in the 1860s and that is also the time slice in which we detect that sense. One weakness of the approach is that although the earliest sense tree can be interpreted easily as sense of *hall monitor*, a closer look reveals that the clusters are only matched by the word *master*. This shows that the matching can be influenced easily by just a few words. Another weakness is that in fact two sense trees represent the meaning of warship. The later sense trees represent distinct senses of *monitor*: the technological sense and the surveillance sense. However, the branches of these sense trees are not separated sufficiently.

Our graph representation of a text corpus can be used to visually explore linguistic features. Figure 2 shows the forest of sense trees for the word *acid*. Each of the boxes represents a set of words $c_i \in S_{w_t}$ as defined before. Each sense tree begins with a green box, the following clusters that were matched across time slices are connected by straight lines. Interestingly, we can see the discovery of DNA in the late 19th century and LSD in the 1960s.

5.5 Limitations

Although our approach is able to identify different senses of some words, there is still room for improvement: graph clustering algorithms and matching strategies,

the evaluation of different window sizes used during the co-occurrence graph creation, and the conduction of a survey to obtain a word dataset that can be used as gold standard for evaluating temporal WSI approaches. Useful features make our approach available as an explorative tool, adjustments in the used language models to make our approach applicable to historic language, a quantitative comparison to word embeddings, verification of the stability of our approach, a custom slicing of time periods, sensitivity for differently spelled variants of the same word, and detecting not only the birth of new word senses but also the death of word senses. However, our approach struggles to create well-partitioned sense clusters for different senses, which also affects the matching strategies. Additionally, these strategies do not prevent sense drifting and sense trees may change their meaning significantly over multiple time slices.

6 Conclusion

We proposed an approach to identify words that gained new meanings over time. Additionally, our approach is interpretable and produces intermediate results that can be used to investigate and understand how the sense gain score for a specific word was constructed. We presented a drill down into specific words with two different visualizations that allow key components of our approach to be easily understood. It enables seeing both the created sense clusters and sense trees, and thus allows one to find the point in time at which new senses of a word emerged.

We applied our approach to COHA, which spans 200 years and is the largest corpus of its kind. We showed anecdotal evidence for the functioning of our approach by manually annotating and evaluating 109 words. We also evaluated our approach qualitatively by using our drill down to inspect the intermediate results of the word *monitor*. We found that our approach was able to successfully identify word senses in sense trees.

References

1. Bamler, R., Mandt, S.: Dynamic word embeddings. In: Proceedings of the International Conference on Machine Learning (ICML). pp. 380–389. JMLR Inc. and Microtome Publishing (2017)
2. Biemann, C.: Chinese whispers: an efficient graph clustering algorithm and its application to natural language processing problems. In: Proceedings of the Workshop on Graph-based Methods for NLP. pp. 73–80. ACL (2006)
3. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* **2008**(10) (2008)
4. Davies, M.: Expanding horizons in historical linguistics with the 400-million word corpus of historical american english. *Corpora* **7**(2), 121–157 (2012)
5. Feuerbach, T., Riedl, M., Biemann, C.: Distributional semantics for resolving bridging mentions. In: Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP). pp. 192–199. ACL (2015)

6. Girvan, M., Newman, M.E.: Community structure in social and biological networks. *Proceedings of the National Academy of Sciences* **99**(12), 7821–7826 (2002)
7. Hope, D., Keller, B.: MaxMax: A graph-based soft clustering algorithm applied to word sense induction. In: *Proceedings of the International Conference on Computational Linguistics and Intelligent Text Processing (CICLing)*. pp. 368–381. Springer-Verlag (2013)
8. Jatowt, A., Duh, K.: A framework for analyzing semantic change of words across time. In: *Proceedings of the IEEE/ACM Joint Conference on Digital Libraries (JCDL)*. pp. 229–238 (2014)
9. Kim, Y., Chiu, Y.I., Hanaki, K., Hegde, D., Petrov, S.: Temporal analysis of language through neural language models. In: *Proceedings of the Workshop on Language Technologies and Computational Social Science*. pp. 61–65. ACL (2014)
10. Kutuzov, A., Øvrelid, L., Szymanski, T., Velldal, E.: Diachronic word embeddings and semantic shifts: a survey. In: *Proceedings of the International Conference on Computational Linguistics (COLING)*. pp. 1384–1397. ACL (2018)
11. Lau, J.H., Cook, P., McCarthy, D., Newman, D., Baldwin, T.: Word sense induction for novel sense detection. In: *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*. p. 591–601. ACL (2012)
12. Lindén, K.: Evaluation of linguistic features for word sense disambiguation with self-organized document maps. *Computers and the Humanities* **38**, 417–435 (2004)
13. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: *Proceedings of the International Conference on Learning Representations (ICLR)*. pp. 1–12 (2013)
14. Mitra, S., Mitra, R., Maity, S.K., Riedl, M., Biemann, C., Pawan, G., Mukherjee, A.: An automatic approach to identify word sense changes in text media across timescales. *Natural Language Engineering* **21**(5), 773–798 (2015)
15. Navigli, R.: Word sense disambiguation: A survey. *ACM Computing Surveys* **41**(2), 1–69 (2009)
16. Rapp, R.: Word sense discovery based on sense descriptor dissimilarity. In: *Proceedings of Machine Translation Summit (MTSummit)*. pp. 315–322. European Association for Machine Translation (2003)
17. Song, L., Wang, Z., Mi, H., Gildea, D.: Sense embedding learning for word sense induction. In: *Proceedings of the Joint Conference on Lexical and Computational Semantics (*SEM)*. The *SEM Organizing Committee (2016)
18. Tahmasebi, N., Risse, T.: On the uses of word sense change for research in the digital humanities. In: *Proceedings of the International Conference on Theory and Practice of Digital Libraries (TPDL)*. pp. 246–257. Springer-Verlag (2017)
19. Ustalov, D., Panchenko, A., Biemann, C., Ponzetto, S.P.: Watset: Local-global graph clustering with applications in sense and frame induction. *Computational Linguistics* **45**(3), 423–479 (2019)
20. Yang, H., Callan, J.: A metric-based framework for automatic taxonomy induction. In: *Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP)*. pp. 271–279. ACL (2009)
21. Yao, Z., Sun, Y., Ding, W., Rao, N., Xiong, H.: Dynamic word embeddings for evolving semantic discovery. In: *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*. pp. 673–681. ACM (2018)
22. Yarowsky, D.: Unsupervised word sense disambiguation rivaling supervised methods. In: *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*. pp. 189–196. ACL (1995)

Schema Evolution and Reproducibility of Long-term Hydrographic Data Sets at the IOW

Tanja Auge¹, Erik Manthey¹, Susanne Jürgensmann²,
Susanne Feistel², and Andreas Heuer¹

¹ University of Rostock, Germany
{firstname.lastname}@uni-rostock.de
<http://dbis.informatik.uni-rostock.de>

² Leibniz Institute for Baltic Sea Research Warnemünde, Germany
{firstname.lastname}@io-warnemuende.de
<https://www.io-warnemuende.de>

Abstract. National and international exploration of the Baltic Sea ecosystem can be traced back to the 19th century. In its quite long history, the *Leibniz Institute for Baltic Sea Research Warnemünde* (IOW) is the only research institution in Germany that has made interdisciplinary research of the Baltic Sea its central mission. The IOW hosts data from more than 130 years of research work.

Using the example of hydrographic datasets that have been created over a period of about 50 years, this paper examines changes in the data and the associated schemes that have resulted from the continuous development and refinement of measurement methods over time. The paper focuses on the schema development operators: What kind of schema development has taken place over the years, and what are the important basic schema development operators that can be identified? It classifies well-known schema evolution operators which can be expressed as schema mappings, and defines two new operators for merging and splitting attributes, up to now not considered in other research works. These operators have proven to be essential for development of a new universal schema for the central oceanographic database on the IOW – the *IOWDB*.

Keywords: Schema Evolution · Baltic Sea · Long-term Data · Research Data Management.

1 Introduction

National and international exploration of the Baltic Sea ecosystem can be traced back to the 19th century. In its quite long history, the Leibniz Institute for Baltic Sea Research Warnemünde (IOW) is the only research institution in Germany that has made interdisciplinary research of the Baltic Sea its central mission. The IOW hosts data from more than 130 years of research work. Due to their origin they were stored in various formats and on diverse storage media.

Copyright © 2020 by the paper's authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

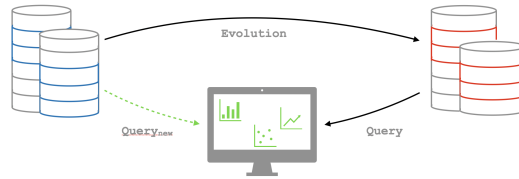


Fig. 1: Unification of Query Evaluation, Provenance and Evolution [1]

The optimal preparation of such research data is part of the so-called *FAIR principles* (**F**indable, **A**ccessible, **I**nteroperable, **R**eusable). These principles define "characteristics that contemporary data resources, tools, vocabularies and infrastructures should exhibit to assist discovery and reuse by third-parties" [12]. They should ensure sustainable research data management by processing the data and their metadata.

The IOW publishes many of its newer data online on IOWMeta³, realizing the F and A of FAIR. We are therefore concentrating on interoperability (I) and reusability (R) to ensure the reproducibility of the data evaluations. To be able to determine exactly those research data that have an influence on the evaluation result, we have to automatically compute the evaluation queries, the scheme and data evolution stages as well as the provenance queries in a homogeneous framework (see Fig. 1). This is important not only for the reproducibility of evaluation results, but also for the updating of evaluations over time.

In this paper, we focus on the process of data collection with a so-called *CTD probe* (see Section 3) in the period from the 1970s to the present. Within this time, the scientific requirements changed dramatically and were accompanied by significant improvements in instrumentation, data acquisition and processing on board of the research vessels and, last but not least, data storage on land.

Within the scope of hydrographic data collection with a CTD probe, new sensors were developed or existing ones improved. Correspondingly new measurement parameters were defined or replaced. The measurements could then be carried out faster and more accurately. Therefore, not only the physical storage media used at the IOW developed enormously during this time, from simple paper records, punch cards and magnetic tapes to modern databases (see Fig. 2), but also the underlying concepts for data processing and structuring had to be reconsidered continuously.

However, this change in the data formats and structures leads to problems nowadays when newer evaluations of the data have to be performed on old data sets or long-term data ranging over some decades [4]. Thus, one goal is the reproducibility of old results with the help of a new evaluation technique.

An evaluation of the data over the entire period of time and the tracing of individual tuples is only possible with considerable effort. In database terms the development of the new structures can result in the creation of new attributes or tables as well as a reassignment of the affected tuples. It is possible that

³ <https://iowmeta.io-warnemuende.de>

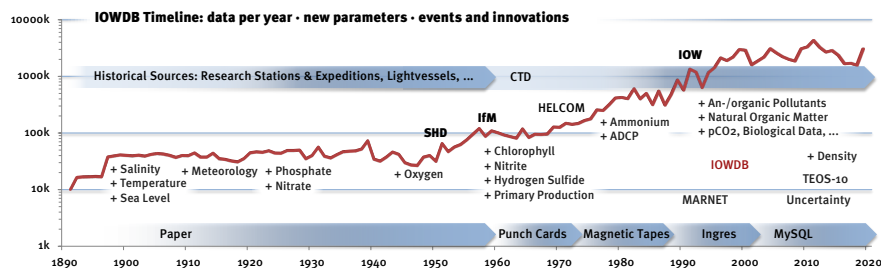


Fig. 2: (Logarithmic) Data increase at IOW [3], showing only the new data per year

the variety of measured variables changes over time. Thus, past and current schemas differ significantly in this respect as well. Columns are renamed, created or deleted. Merging or splitting of attributes can be observed.

To be able to support reproducibility over time, we have to combine (1) the data evaluations (represented as database queries including some linear algebra functions), (2) the schema evolution steps, and (3) the process of inverting these steps by means of data provenance techniques (*why-* and *how-*provenance). Since we already developed formal techniques based on schema mappings (such as s-t tgds, i.e. source-to-target tuple-generating dependencies) [8] for steps (1) and (3), we have to integrate schema evolution steps by the use of schema mappings, as well [2]. This can be done by using formally defined schema modification operators such as the ones introduced by the PRISM++ project [5].

The combination of data provenance with schema and data evolution enables us to evaluate provenance queries on evolving data and schemas. Through inverse evolution steps, the new database can be transferred to the old schema. Formally, we use the CHASE algorithm, a universal tool of database theory. Our application of the CHASE for schema (and corresponding instance) mappings is based on the s-t tgds mentioned above. It is shown in [5], that the *Schema Modification Operators* (SMOs) and the *Integrity Constraint Modification Operators* (ICMOs) can be transformed to such kind of schema mappings.

In Section 2 we first introduce the original SMOs and ICMOs of PRISM++. In this paper, we focus on the analysis of the schema evolution process at the IOW, that is briefly presented in Section 3. We identify the most important SMOs and ICMOs which are necessary to describe the IOW schema evolution, and define two new operators `MERGE Column` and `SPLIT Column` (see Section 4).

2 State of the Art

Our work is based on three different schema definitions: Schema Mapping, Schema Evolution and Schema Operator. We understand *Schema Mapping* as a concrete evaluation query, *Schema Evolution* as the overall process of evolution and *Schema Modification* as the individual step itself.

Table 1: Schema Modification Operators, based on [6, 10]

SMOs	SMOs
COPY Table R INTO S	ADD Column d [AS const func(a,b,c)] INTO R
CREATE Table R(a,b,c)	COPY Column c FROM R INTO S WHERE cond
DECOMPOSE Table R(a,b,c) INTO S(a,b), T(b,c)	DROP Column c FROM R
DISTRIBUTE Table / PARTITION Table R INTO S WITH cond, T	MOVE Column c FROM R INTO S WHERE cond
DROP Table R	RENAME Column b IN R TO d
JOIN Table R, S INTO T WHERE cond	NOP
MERGE Table R, S INTO T	
RENAME Table R INTO S	

Schema Mapping: *Schemas* and *Schema Mappings* are two fundamental components of heterogeneous data management. While schemas specify the structure of the various databases, schema mappings describe the relationships between them. Schema mappings can be used in particular to transform data between two different schemas.

Schema Evolution: *Schema evolution* describes the schema changes of a database over time. It therefore give a description to the overall process. This includes changes in the relational schemas themselves as well as changes in the key and integrity conditions. In order to enable earlier states to be analyzed retrospectively and queries on other schema versions to be made, starting from the current database state, PRISM++ [5, 6, 10] introduced special operators which describe the changes in schemas and integrity conditions in a compact way. For defining these operators, the authors examined several schemas from practice concerning their modifications [6]. The most frequently occurring operators formed the core of these *Modification Operators* which are divided into *Schema Modification Operators* (SMOs) and *Integrity Constraint Modification Operators* (ICMOs). While SMOs can describe the direct changes of relations, such as the creation or deletion of individual attributes or entire relations, ICMOs are used whenever key or integrity conditions in relations change. Thus, these conditions can be generally tightened or weakened by adding or discarding restrictions.

Schema Modification Operators (SMOs): The so called *Schema Modification Operators* are basic operators used to describe changing database schemas. Thirteen operators are defined in [6, 10], summarized in Table 1. Each operator captures an atomic change and by combining them, it is possible to express complex evolutions. The most common ones are CREATE Table, DROP Table, ADD Column, DROP Column and RENAME Column [11, 13]. These also correspond to the operators relevant to the IOW, gray highlighted.

Integrity Constraints Modification Operators (ICMOs): In addition to the SMOs, there is another class of Schema Evolution Operators. These *Integrity Constraints Modification Operators* (ICMOs), first introduced in [5], describe changes in integrity conditions. More precisely, the six ICMOs handle the

Table 2: Integrity Constraints Modification Operators, based on [5]

ICMOs	
ALTER Table R	ADD PRIMARY KEY pk1(a,b) <policy>
ALTER Table R	ADD FOREIGN KEY fk1(c,d) REFERENCES T(a,b) <policy>
ALTER Table R	ADD VALUE CONSTRAINT vc1 AS R.e = "0" <policy>
ALTER Table R	DROP PRIMARY KEY pk1
ALTER Table R	DROP FOREIGN KEY fk1
ALTER Table R	DROP VALUE CONSTRAINT vc1

three restrictions PRIMARY KEY, FOREIGN KEY, and VALUE CONSTRAINT. Attention must be paid to the Enforcement Policy <policy>, which is necessary for the first three ICMOs. Here we can choose between

- (1) **CHECK**: The system checks whether the current table satisfies the new primary key. If not, no primary key is added.
- (2) **ENFORCE**: The primary key is added in any case. Tuples that violate the new key are deleted.

However, before we begin to investigate the schema evolution, we will briefly introduce the IOW in general as well as the analyzed data sets of the IOW.

3 Research at the IOW

National and international exploration of the Baltic Sea ecosystem can be traced back to the 19th century. In its quite long history, the Leibniz Institute for Baltic Sea Research Warnemünde is the only research institution in Germany that has made interdisciplinary research of the Baltic Sea its central mission. The IOW hosts data from more than 130 years of research work. Due to their origin they were stored in various formats and on diverse storage media.

Leibniz Institute for Baltic Sea Research Warnemünde (IOW): The story of the IOW goes back to the German Democratic Republic (GDR) of the 1950s. At that time, the Seehydrographic Service of the GDR was founded in order to be able to operate independently of the west-German equivalent. It included, among other things, a separate department for oceanography, which was known as the *Institute of Oceanography Warnemünde* (IfM-W) in 1960. The institute achieved international recognition through independent research cruises and was able to establish itself as a permanent fixture in worldwide marine research.

The systematic exploration of the Baltic Sea finally began in 1964 with the *International Synoptic Recording of the Baltic Sea*. In the following decades, these regular scheduled cruises, i.e. cruises at fixed times and places, became an important part of Baltic Sea research within the IfM-W.

After the German reunification in 1990, research and science in the whole of Germany had to be newly regulated and ordered, which led to the foundation of the *Institute for Baltic Sea Research Warnemünde*⁴ in 1992.

⁴ <https://www.io-warnemuende.de>

IOWDB: The *Oceanographic Database of IOW* (IOWDB) had originally been designed for particular internal requirements of the IOW. The IOWDB has always been aimed at the management of historical and recent oceanographic measurements.

Research cruises have been conducted since 1949, and their data systematically collected. The post-processing and analysis of the resulting observational data was carried out by varying methods and with changing quality. A substantial fraction of legacy data was successively transferred to modern storage media, their quality was controlled and, if possible and necessary, improved by detailed individual scientific review.

The content of the database includes oceanographic readings and metadata (mainly Baltic Sea) from 1877 to 2020 obtained during 951 research campaigns of the IOW and cooperating institutions. As of June 2020, the IOWDB contains more than 78 million measured samples representing georeferenced point data from the water column, primarily from CTD profiles, hydrochemical and biological sampling, current-meter time series, trace metal sampling and long-term monitoring. Phyto- and zooplankton data are available for 1988 to 2018.

CTD measurement: One third of the stored data in the IOWDB are obtained with a so-called *CTD probe*. Primary parameters of this instrument are **C**onductivity (electrical conductivity, which is used to determine salinity), **T**emperature and **D**epth, which is determined by the prevailing pressure.

Optional sensors, e.g. for oxygen, fluorescence and other parameters may be added. The CTD probe is surrounded by several water samplers, allowing additional water samples to be taken. These are then used for further analyses.

The examined CTD data originates from regularly conducted monitoring cruises. On these cruises fixed locations at the Baltic Sea are travelled, so the changes of CTD values can be recorded. After first recording the primary data from the cruise, the resulting records are validated. In this case validation means that the values from missing depths are interpolated to get an approximated value. Within this paper's research only these validated data was examined.

Next we will take a closer look at the schema evolution at the IOW. We focus here on the evolution of the CTD data.

4 Schema Evolution at IOW

We describe the schema and the corresponding changes over a period from 1977 to the present day solely for the data resulting from CTD measurements. It should be noted that adding measurement data of a new type would of course result in a new schema.

4.1 Schema Changes over the Years

Although data from CTD measurements exist from the early 1950s on, the data files from the years before 1977 are very difficult to interpret. Essentially, the

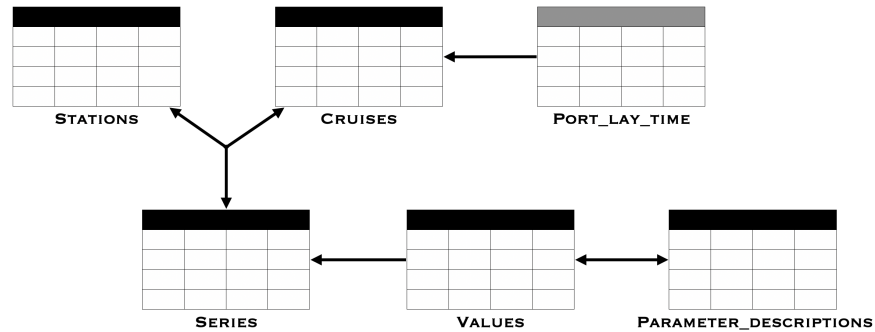


Fig. 3: Universal schema from 2017

files contain a large amount of measured values, but there are hardly any descriptions or parameter names and thus insufficient schematic information. Without historical background knowledge of the underlying process of data collection, only vague assumptions can be made to identify the underlying scheme. For this reason, we start with a more detailed investigation in 1977.

Because no structural changes occurred during each period, the schema changes over the years can be divided into three blocks: one period between 1977 and 1996, another between 1997 and 2016 and the years after 2017 [9].

1977-1996: The schema is clearly defined and divided into five relations describing the metadata of the CRUISES, of the measurement STATIONS, the SERIES, as well as the actual recorded CTD VALUES and the corresponding PARAMETER_DESCRIPTIONS. The five relations are connected by three relationships (see Fig. 3): In case of CTD measurements there is a trivalent relationship ($\leftarrow_{\mathbb{X}}$) between the cruise, station and series relations. The relation with the CTD values is dependent (\leftarrow) on the relation containing information about the measurement series. And the relation with the parameter descriptions is connected to the relation with the CTD values by a "belongs to" relationship (\leftrightarrow). The set of attributes corresponding to the relations and relationships is quite limited compared to the schemas of the following years. For example, in 1977 the schema contained only 34 attributes, whereas in 1997 it already contained 61 attributes, and the trend is rising.

1997-2016: However, the 1997 scheme brings some major changes. The previous entities remain unchanged, but we can record a lot of new attributes, a new relation (highlighted in Fig. 3) and associated changes in integrity conditions. Thus, in the SERIES and CRUISES relations, the attributes to be stored increase from 15 to 26 and from 5 to 17 attributes, respectively. The new relation PORT_LAY_TIME records the laytime of a research vessel in a port and is therefore dependent on the CRUISE relation. STATIONS, VALUES and PARAMETERS_DESCRIPTION remain unchanged. Furthermore, some attributes are split

while others are merged. All in all these are some minor and major changes. We will discuss this in more detail later on.

After 2017: The previous structure of the schemas from 1997 to 2016 remains unchanged, only seven attributes are added to the relations. Overall, there are no major changes and all schemas described previously can be mapped to it. The 2017 schema is therefore used as the universal schema, shown in Fig. 3.

4.2 Specification of Schema and Data Changes

Operations that are repeatedly performed when converting schemas before 2017 include, in particular, the addition, merging and splitting of attributes (see Table 3 (a)). Since the schemas gain more information over the years, this is not surprising. This continuous process of expansion is only supplemented a few times by renaming individual attributes, creating the new table `PORT_LAY_TIME` and adapting related integrity constraints in 1997. However, we will investigate this further.

Most evolution steps can be described easily using the SMOs and ICMOs defined for PRISM++ [9]. Only merging and splitting attributes requires a new operator can be understood as a sequence of the basic operators `ADD Column` and `DROP Column`. They are defined in Section 4.3. Because of their special importance at the IOW they shall be defined here as separate operators. But first of all let's have a look on the obvious schematic changes.

Adding and deleting attributes: These types of schema changes can be identified by the SMOs `ADD Column` or `DROP Column`. Adding attributes is one of the easiest described schema changes and occurs comparatively often but least frequently. There are almost fifty new attributes in total. This observation is also consistent with the results of other studies such as [7]. Here 38.7% of all schema changes were due to the addition and 26.4% to the deleting of attributes.

Looking at the relation `SERIES`, we observe the almost complete substitution of all associated attributes. However, contrary to our initial assumption, these attributes are not deleted irreversibly. They are rather merged into new tuples or split into new attributes. This results in the fact that we have to develop the two new operators `MERGE Column` and `SPLIT Column`.

Table 3: Detected schema changes at IOW

	# changes	percentage share		# changes	percentage share
CREATE Table	1	1.6%	CREATE Table	1	2.3%
ADD Column	45	72.6%	ADD Column	35	79.5%
DROP Column	15	24.2%	DROP Column	0	0%
RENAME Column	1	1.6%	RENAME Column	1	2.3%
			MERGE Column	4	9.2%
			SPLIT Column	3	6.8%

(a) with basic SMOs

(b) extended with `MERGE Column` and `SPLIT Column`

Example 1. The SMO `ADD COLUMN` extends `CRUISES` by the possibility of a comment, formally :

```
ADD COLUMN Comment INTO Series.
```

Creating relations: This form of schema change occurs only once in the data sets examined. Since 1997, the relation `PORT_LAY_TIME` is part of the schema. It should be noted that in addition to the SMO `CREATE Table` itself, additional integrity conditions must be specified. On the one hand, it is necessary to create a primary key. On the other hand, due to the dependence on `PORT_LAY_TIME` and `CRUISES`, a suitable foreign key must be specified.

Adding and deleting primary keys: Creating relation `PORT_LAY_TIME` implies the requirement of a specialized attribute, the so-called primary key, which clearly identifies the tuples of the new relation `RESIDENCE`. Furthermore, when mapping the schema from 1977 to the universal schema, the primary key of `CRUISES` is changed. This is equivalent to deleting the old primary key and adding a new one.

The primary key can be modified using the ICMOs `ADD PRIMARY KEY` or `DROP PRIMARY KEY`. When creating a primary key, the enforcement policy must be observed as well. This checks whether individual tuples violate the new integrity condition or not. If so, the ICMO will not be applied (`CHECK` policy), if not, all tuples violating the newly introduced constraint are removed (`ENFORCE` policy). Accordingly, the primary key should always be created directly after or during the creation of the new table.

Adding foreign keys: Since the new relation `PORT_LAY_TIME` is dependent on `CRUISES`, a foreign key must be specified additionally to the primary key. This is derived as usual from the primary key of the higher-level relation.

Example 2. Let us take a closer look on the relation `PORT_LAY_TIME`. Introduced in 1997, it is dependent on `CRUISES`. Besides the creation of the relation itself using the SMO

```
CREATE Table Port_lay_time
    (ArchiveNo, Reason_for_stay, Date, Harbour, Duration),
```

a primary key consisting of the attributes *ArchiveNo*, *Date*, *Harbour* and *Duration* must be defined and the foreign key condition of *ArchiveNo* be specified. Adding key and integrity constraint can be realized using the ICMOs

```
ALTER Table Port_lay_time
ADD PRIMARY KEY pkLZ
    (ArchiveNo, Date, Harbour, Duration)
CHECK
```

and

```
ALTER Table Port_lay_time
ADD FOREIGN KEY fk (ArchiveNo) REFERENCES Stations(ArchiveNo)
CHECK.
```

Table 4: SMOs for merge and split

MERGE Column a,c AS func(a,c) IN R TO d	SPLIT Column a IN R TO d USING func ₁ , e USING func ₂
ADD Column d AS func(a,c) INTO R; DROP Column a FROM R; DROP Column c FROM R;	ADD Column d AS func ₁ (a) INTO R; ADD Column e AS func ₂ (a) INTO R; DROP Column a FROM R;

Renaming an attribute: Also the SMO `RENAME Column` exists only a few times. Even if this looks rather harmless on first sight, it must be ensured that the old evaluations can no longer be easily reproduced on the new schema. It occurs for example in the evolution of the relation `SERIES`.

Example 3. When mapping the schema from 1997 to the universal schema, the Attribute `ws` is renamed to `ws-ID`. This can be described by:

```
RENAME COLUMN ws IN Series TO ws-ID.
```

Merging and splitting attributes: In the course of the years and decades, attributes are repeatedly summarized at the IOW. According to PRISM++ there is no independent Schema Modification Operator for these changes. So let us next define these two operators `MERGE Column` and `SPLIT Column`.

Example 4. The aim of the two new operators is a representation of the form

```
MERGE Column BeginDate AS func(StartYear,StartMonth,StartDay)
      INTO Series.
```

and

```
SPLIT Column Date IN Series TO
      StartDate USING func1(Date), EndDate USING func2(Date).
```

4.3 MERGE Column and SPLIT Column

Besides the schema changes, which can already be described by the operators defined in [5] and [6], there are at least two more, which can be expressed by composites of these and adding one or more functions. These additional changes consist in the merging and splitting of columns as seen in Table 4. At the IOW they occur, among other things, when time and date columns are changed.

MERGE Column: As seen in Table 4, merging two columns into a new column is realized by executing `ADD Column` and `DROP Column` one after the other. For creating the new attribute values a function `func` is used, which combines the old attributes. Accordingly, merging consists of creating a new column and then deleting the old ones.

In the case of relation `SERIES` the three columns `StartYear`, `StartMonth` and `StartDay` are merged into the new column `BeginDate`. The SMO `MERGE Column` can be interpreted as

```

ADD Column BeginDate AS func(StartYear,StartMonth,StartDay)
    INTO Series
DROP Column StartYear FROM Series;
DROP Column StartMonth FROM Series;
DROP Column StartDay FROM Series.

```

Assuming that the date type of the attribute *BeginDate* is a string of the format DD.MM.YYYY, we can choose the function `func` as concatenation of the form

```
func := CONCAT(StartDay, '.', StartMonth, '.', StartYear).
```

However, this choice is in no way binding.

SPLIT Column: Splitting a column has a similar structure. Here too, new columns are created and old ones deleted. One major difference, however, is that each new column requires its own function, implemented in SQL, for example.

So, in the case of *CRUISES*, the column *Date* is split into *StartDate* and *EndDate*. The `SMO SPLIT Column` can be interpreted as

```

ADD Column StartDate AS func1(Date) INTO Cruises;
ADD Column EndDate AS func2(Date) INTO Cruises;
DROP Column Date FROM Cruises.

```

Assuming the format DD.MM.YYYY we can divide the string into two substrings of length 10 using the functions

```

func1 := SUBSTRING(Date,1,10),
func2 := SUBSTRING(Date,12,10).

```

This static approach is only a simplified example because of the high error rate. Date values may have been saved incorrectly (e.g. typing errors) or in a different format (e.g. American date format). Functions that automatically split the source string are of course better, but would go beyond the scope of this example.

In total, we could identify seven merge and split operations. These contain 10 times `ADD Column` and 15 times `DELETE Column`. Compared with the previous studies we get the new distribution shown in Table 3 (b). The most common operator remains `ADD Column`, but we no longer need the `DROP Column` operator, at least not explicitly. Metadata is often given as strings or intervals. Since intervals were repeatedly represented as a single attribute or as interval boundaries (divided into two attributes), further changes are expected in the future.

4.4 Implementation

Even though the selection of our operators is based on the works [5] and [6], we have not implemented them in the PRISM++ system. We have decided for an implementation in a schema modification interface to MySQL which is already used at the IOW. All operators from Section 4.2 are implemented as described above. The auxiliary functions `funci` mentioned in Section 4.3 were implemented with appropriate `Update` operations, so we did not have any problems with the prototypical implementation here either.

5 Conclusion and Future Work

In this paper we classified – using CTD data as an example – the schema evolution operators relevant for research data management at the IOW. In particular, the addition of attributes using `ADD Column` as well as the merging and splitting of attributes were considered relevant. We have redefined the required operators `MERGE Column` and `SPLIT Column` based on the existing SMOs of [5].

To support reproducibility over time, we must combine (1) data analysis, (2) schema development steps, and (3) the process of reversing these steps using data prevention techniques. Since we have already developed formal techniques based on schema mappings for steps (1) and (3), the schema evolution steps must also be integrated by using schema mappings [1, 2]. Now, with our extended PRISM++ approach, we can apply the evolution operators expressed as s-t tgds against the given research database using the CHASE algorithm. And the IOW would thus be able to track the oxygen content of the Baltic Sea as well as other interesting parameters over a period of decades.

References

1. Auge, T: Extended Provenance Management for Data Science Applications. PhD@VLDB, CEUR Workshop Proceedings, CEUR-WS.org (2020)
2. Auge, T.; Heuer, A.: Combining Provenance Management and Schema Evolution. IPAW, 222–225 (2018)
3. Bock, S.; Feistel, F.; Jürgensmann, S.: Data Management at IOW. Poster (2014)
4. Bruder, I.; Klettke, M.; Möller, M. L.; Meyer, F.; Jürgensmann, S.; Feistel, S.: Daten wie Sand am Meer - Datenerhebung, -strukturierung, -management und Data Provenance für die Ostseeforschung. Datenbank-Spektrum **17**(2), 183–196 (2017)
5. Curino, C.; Moon, H. J.; Deutsch, A.; Zaniolo, D.: Update Rewriting and Integrity Constraint Maintenance in a Schema Evolution Support System: PRISM++. PVLDB **2**(4), 117–128 (2010)
6. Curino, C. A.; Moon, H. J.; Zaniolo, C.: Graceful Database Schema Evolution: the PRISM Workbench. PVLDB **1**(1), 761–772 (2008)
7. Curino, C. A.; Tanca, L.; Moon, H. J.; Zaniolo, C.: Schema Evolution in Wikipedia: Toward a Web Information System Benchmark. ICEIS(1), 323–332 (2008)
8. Fagin, R.; Kolaitis, P.G.; Popa, L.; Tan, W.C.: Schema Mapping Evolution Through Composition and Inversion. In: Schema Matching and Mapping, Springer, 191–222 (2011)
9. Manthey, E.: Beschreibung der Veränderungen von Schemata und Daten am IOW mit Schema-Evolutions-Operatoren. Bachelor Thesis, University of Rostock (2020)
10. Moon, H. J.; Curino, C.; Deutsch, A.; Hou, C.-Y.; Zaniolo, C.: Managing and Querying Transaction-time Databases under Schema Evolution. PVLDB **1**(1), 882–895 (2008)
11. Qiu, D.; Li, B.; Su, Z.: An Empirical Analysis of the Co-evolution of Schema and Code in Database Applications. ESEC/SIGSOFT FSE, ACM, 125–135 (2013)
12. Wilkinson, M.; Dumontier, M.; Aalbersberg, I. et al.: The FAIR Guiding Principles for scientific data management and stewardship. Sci Data **3**, 160018 (2016)
13. Wu, S.; Neamtiu, I.: Schema evolution analysis for embedded databases. ICDE Workshops, IEEE Computer Society, 151–156 (2011)

Future Fetch

Towards a ticket-based data access for secondary storage in database systems

Demian E. Vöhringer and Klaus Meyer-Wegener

FAU Erlangen-Nuremberg, Chair for Computer Science 6 (Data Management)
{firstname.surname}@fau.de, <https://www.cs6.tf.fau.de/>

Abstract. When accessing data from a database, a database management system has to accomplish many tasks. Checking the conformity of the query, generating a reasonably good query plan and executing it in a concurrent system are just a few of them. During query execution, there are some points at which the main execution thread must stop and wait for data. This synchronous waiting can have a major impact on the overall query performance. In this paper we introduce the idea of a ticket-based fetch, which supports the database management system by waiting for data asynchronously. This can improve the query-execution performance in database systems.

Keywords: Asynchronism · Data Flow · Data Access · Ticket · Database System · Data Warehouse · Big Data · Concurrent

1 Introduction

Large database systems (DBS) depend heavily on fast access to huge volumes of data, which can easily exceed the capacity of a server’s random-access memory (RAM) by orders of magnitude.

These data can be stored in three different storage spaces, the primary, secondary and archive storage space [9]. They all have different properties that allow them to be used for different tasks. Primary storage uses RAM, while solid-state and hard-disk drives are found in secondary storage, which is orders of magnitude slower. If the data a query is to be executed on is stored in the secondary storage space, they must be transferred to the much faster primary storage space to perform operations on them. It is therefore crucial that algorithms of the DBS depend as little as possible on data in slower storage spaces and try to rely only on data that has already been loaded into primary storage. However, if data are not in main memory, they still must be fetched from disk, which can cause long latencies and slows down the execution of queries.

G. Graefe [6] summarized ideas like indices, buffer management and parallelism in DBS, which allow to cope with these latencies. With the help of indices,

Copyright © 2020 by the paper’s authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

data can be found much faster on secondary and archival storage, while good buffer management helps keeping the right data in main memory. Parallelism in database systems is divided in three categories, namely interquery parallelism, horizontal interoperator parallelism, and vertical interoperator parallelism. With interquery parallelism, a DBS can execute several queries simultaneously and therefore use waiting times for data loading by switching to another query that already has its data. Horizontal interoperator parallelism performs the same operation on several partitions of the tuple set at the same time and can take advantage of modern multi-core CPUs. Vertical interoperator parallelism executes operators in pipelines, as shown in [10].

These ideas about parallelism focus more or less on data already accessible in primary storage, or on finding data in the storage spaces, ignoring the fact that the transfer of these data to primary storage adds latency to the execution of each single query. The question is whether we can support these parallelism ideas even better by providing an asynchronous data flow inside a DBS.

This can be useful in a query that connects many tuples from different tables to create one large result tuple, as it is done in star-shaped queries (e.g. Listing 1.1). A star-shaped query is similar to a star query, which uses a main table (fact table) that contains information about how to connect the other tables (dimension tables). The only difference is that star queries are bound to data-warehouse systems and use information from the dimension tables mostly for reducing the result-set size¹. With a limited number of values in the dimension tables, star queries can be optimized by using bitmap indices, while star-shaped queries cannot use this kind of optimization and are thus much harder to optimize.

In this paper we present some first steps of the idea of a ticket-based data loading from secondary storage. We do this by introducing a ticket system for database operators. It desynchronizes data and execution flow of query processing.

We begin our journey with a simple example. We describe the problem (Section 2), find a first solution (Section 2.1), explain that solution in depth (Section 2.2), and try to expand that solution, so it can be executed in a more general workflow of query execution (Section 2.3). After that we discuss related work (Section 3), draw a conclusion, and present future work (Section 4).

2 The Idea

To work with the data, a classical DBS must transfer it from secondary or archival storage to primary storage. This process is shown in Figure 1a for a star-shaped query. Because the data-transfer rate of secondary storage is limited, access to the data takes some time, so any query execution in need of these data must interrupt its execution flow and wait some time for the data to arrive.

This pause can become large depending on the exact location on disk and the access path for it, and the query execution must wait, even if the data is

¹ <http://www.orafaq.com/tuningguide/star%20query.html>

not needed immediately. To improve the search, indices have been introduced, which can help to find the required data faster on secondary storage. Once the data is collected, query execution proceeds to the next step, which may require some more data. These steps must be repeated for each tuple-fetch operation, until the result set is complete and can be returned to the caller. So waiting for data can take a large portion of the execution time of a single query.

To reduce this data latency, we present an initial idea of ticket-based data access, describe it in depth, and expand it towards a more general query execution.

2.1 A first step

To make the idea understandable, we simplify the problem a bit. We ignore some ideas on parallelism presented in [6] and only focus on one query that works on one large set of tuples. This way we still have vertical interoperator parallelism, but neither interquery parallelism nor horizontal interoperator parallelism. Additionally we assume that secondary indices (an index not containing data, but only information where to find the data in the storage spaces) can be kept in primary storage for fast access.

With our idea we want to intervene exactly where the query execution must wait for data and try to improve the execution time with asynchronous data access. As shown in Figure 1c, our first step towards asynchronous data access focuses on an index scan (a scan operator using an index) combined with a special *Data Access Manager*, our abstraction for access to secondary storage and moving data to primary storage.

Traditionally, an index scan performs three steps in accessing the data from secondary storage. First it uses a secondary index and checks whether and where the corresponding tuple can be found. Because of our assumption, this is done in main memory and thus very quickly. So the location of the tuple on secondary storage is known. In the second step, this tuple is loaded to primary storage, where the third step continues and returns the tuple to the calling operator. In our example, this is a join operator.

If we now rework this index-scan operator to use the Data Access Manager, we obtain two operators. We get a “check for existence, get the storage-space address and start loading” (\exists) and a “wait for loading and combine the loaded tuple with the existing tuple” (ω) operator, as seen in Figure 1b. We describe these operators and the Data Access Manager in detail in Section 2.2. You can easily see that the second step has been delegated to the Data Access Manager.

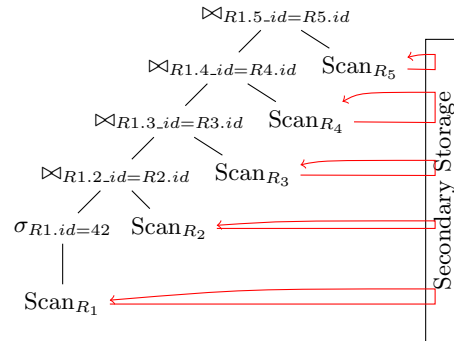
Now that we have refactored our index scan into two operators, we move them in our query-execution tree to optimize it, as shown in Figure 1c. The \exists operator remains in the same place where the scan was, while the ω operator is moved to where the data is actually needed. In our example, this is the end of query execution just before returning the result tuple, where we insert the “wait and combine all data” operator. Because the data from R_1 are needed right now, we can use the old scan operator for simplicity.

```

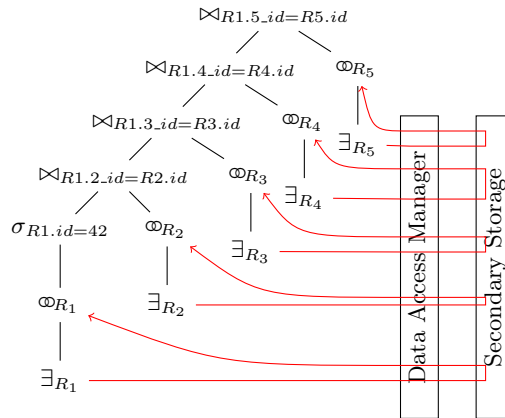
SELECT *
FROM R1, R2, R3, R4, R5
WHERE R1.id = 42
AND R1.2_id = R2.id
AND R1.3_id = R3.id
AND R1.4_id = R4.id
AND R1.5_id = R5.id;

```

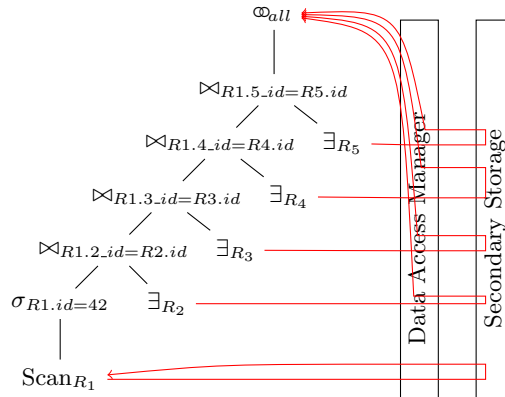
Listing 1.1: Exemplary SQL for a star-shaped query



(a) Traditional access approach



(b) Traditional access approach with our newly designed operators and the Data Access Manager



(c) The complete ticket-based system approach

Fig. 1: Execution trees of the star-shaped query presented in Listing 1.1. Access to data from secondary storage is presented in red.

We can now easily see that the main execution path just informs the Data Access Manager which tuples need to be moved from secondary storage to primary storage for easy access. This has several advantages, as described in the details (Section 2.2).

We expect that this change in strategy can already improve the execution time of quite small star-shaped queries in huge DBS, and we hope to show in the future that this also holds for other query types.

2.2 The operators in detail

Here we explain in more detail our thoughts on the two operators and the Data Access Manager mentioned above.

The \exists operator is designed to be called with a table and an expression (e. g. simple search argument). It queries a secondary index structure for the position of the tuple in secondary storage space. With this address and a certain priority, it calls the Data Access Manager and receives a *ticket* (e. g. a pointer or “future object”²) with information later allowing access to the fetched tuple. This ticket is then forwarded to the next operator. Now the data flow is separated from the execution flow and the current tuple is much smaller, since data not currently needed are not yet part of the tuple. This may have the additional advantage of being able to store the tuple closer to the CPU.

The ticket is now part of the tuple and is resolved by the ∞ operator when the data are needed. The ∞ operator looks at the ticket and determines whether the data is already stored in primary storage. If this is not the case, it recalls the Data Access Manager with the ticket, requests a higher priority and waits for the data to be delivered. This results in a delay that should not be greater than the normal access delay in a synchronous call. Hopefully the data can already be found and can be accessed directly without any delay. Now the data can be combined with the tuple and can be presented to the next operator. If the ∞ operator needs to get the data from multiple tickets at the same time, it can prioritize the already loaded data and can give the Data Access Manager more time for completing the other tickets.

The Data Access Manager has a list of all tuples that must be moved to primary storage and can therefore improve throughput by taking advantage of the secondary storage device’s properties, reorganizing the tuples’ access order accordingly³. If a tuple is needed right now, the access order can be rescheduled to make that data accessible sooner.

2.3 The next steps

The first step exploits some simplifications, like reducing the parallelism and keeping the secondary indices in primary storage. We see no problem in bringing

² <https://docs.oracle.com/javase/7/docs/api/java/util/concurrent/Future.html>

³ <http://www.cs.iit.edu/~cs561/cs450/disksched/disksched.html>

the two other parallelism ideas back. The secondary indices in primary storage gives us the advantage of knowing how many tuples to expect without needing to access the secondary storage. Additionally we have dropped the idea of batch execution, where at the same time multiple tuples are given to an operator as input.

We now like to expand the idea to overcome these simplifications.

For querying a secondary index that is not in primary storage, we move the querying of the secondary index into the Data Access Manager. The \exists operator then does not give an address to the Data Access Manager, but just the table and the expression. By limiting the information, we force the Data Access Manager to do a normal index scan, where the index might not be stored in primary storage. This way we cannot predict how many tuples are going to be returned. Based on our ticket system, the Data Access Manager now creates a data structure that stores from zero to multiple tuples in it. The ω operator reacts to this by either dropping the given tuple (it has no join partner), or by iterating through the data structure and creating a combination with each of the retrieved tuples, resulting in multiple tuples for the next operator. Depending on the query and the expected drop rate, the query optimization may place the ω operator closer to or farther away from the \exists operator to balance between reduced unnecessary workload and increased parallel execution.

With this change, we can use arbitrary scan operators with the Data Access Manager, and may also be able to include basic operations such as a projection or selection to increase the amount of parallel execution and to reduce the amount of space on primary storage needed to store the tuples.

To handle batch execution, we expect the \exists operator to open a ticket for each of the tuples in the batch. No further adjustments are needed, because the remainder of the execution can be kept as if there was only one tuple, only that we now have to do this for each tuple of the batch. The ω operator has another small advantage in being able to check tickets from multiple tuples at once, rather than waiting for the data of a particular tuple to be accessible right now.

With these steps we expect to be able to handle arbitrary queries in current DBS.

3 State of the Art

We found the idea of increasing the speed of slower devices already in the work of Sarawagi [12], where the optimization of access to tertiary memory (archival storage) is discussed. Here the author suggest using a scheduler to first bring all needed data in the secondary storage space and then start executing the query on it. Even if you do not consider that we are talking about secondary to primary storage data transfer and not archival to secondary, we still differ in the idea, because we try to make data accessible while running the query and not to load the data into a faster storage space before executing the query, which may not need all the data.

Codd described a restriction in [4] that was later used in the concept of semi-joins [2]. This operator exploits the idea of reducing a relation to the tuples needed before joining it with another relation. With this reduction all unneeded tuples are dropped early and the join can be performed much faster. In particular if working over a network, the data to be transferred can be reduced substantially. The idea seems similar to ours, but semi-joins are not using the potential of asynchronous data flow, and the relations are reduced in the execution of a join and not before.

For asynchronous data access there are some implementation concepts like asynchronous I/O⁴ or future objects⁵. These are some basic techniques for adding asynchronism to software systems and we plan to use them for implementing the Data Access Manager.

The idea of splitting an operator into smaller bits and looking at these bits was recently proposed by Dittrich and Nix [5]. This paper focuses on the idea of creating physical operators that fit the data better, while we focus on establishing a new asynchronous data flow that ignores the borders of scan operators.

Gurumurthy et al. [7] gave an overview of constructing operators from smaller parts and adapting more easily to new specialized hardware like GPUs [8] and FPGAs [1]. We share the idea of having smaller parts in operators, but focus on data access and not on adaptability to new hardware.

There have been some ideas for operators that can take advantage of better disk-scheduling algorithms, e.g. [3]. Here the authors present a physical scan operator that can adapt to errors in cardinality estimation by prefetching more blocks with tuples to tune between an index and a table scan. We, in contrast, are not defining a new physical operator to replace a logical operator, but change the size and borders of existing scan operators to better optimize the data flow in the query execution.

We found the idea of optimizing star queries inside the Oracle Documentation [11], but this is very specific for star queries in data warehouses. The data from the dimension tables is only used for comparison with constraints and is dropped afterwards. This way they use bitmap indices for the dimension tables to say whether a tuple fits the constraint, which can increase the speed of star queries in data warehouses. We just use star-shaped queries as an easy example and must return the data from the “dimension” tables.

In HyPer [10] a data-aware computing method was introduced by switching the execution direction inside of a pipe from pulling to pushing. By doing this, the data can be kept in CPU registers and need not be stored and loaded from slower storage areas. We do not focus on keeping the data close to the CPU, but instead try to minimize the waiting time when loading the data into a faster storage space. We think it might be possible to combine these two ideas.

⁴ <https://man7.org/linux/man-pages/man7/aio.7.html>

⁵ <https://en.cppreference.com/w/cpp/thread/future>

4 Conclusion and Future Work

In this paper we present our idea of desynchronizing the data and execution flow in query processing. We do this by introducing a Data Access Manager and by splitting scan operators into new operators focused on data access. This way we are able to place them independently in the query-operator tree. We expect to reduce the latency for data access in a DBS and the overall query-execution time.

We are fully aware that this idea increases the search space of query optimization even more, but we rather see this as a challenge for research than a problem.

As future work we plan to implement and test our idea. Because we expect the rework of the software to be quite large, we have decided to use a lightweight, relatively simple, and open-source database management system, namely Apache Derby⁶.

We plan to do this by dividing the task into subtasks, which are: the implementation and evaluation of the Data Access Manager with the ticket system, the splitting of the scan operator, and an additional optimizer step for the physical execution plan that replaces the standard operators with our operators and moves them to fit our idea. Afterwards, we would like to extend the implementation with ideas from our next steps (Section 2.3). For evaluation we plan to run some queries from the TCP-H benchmark on datasets magnitudes larger than the RAM of the specific machine. Furthermore, we think about adding the idea to distributed or federated DBS, to reduce the data-access latency of an access to a remote system.

References

1. Becher, A., B.G., L., Broneske, D., Drewes, T., Gurumurthy, B., Meyer-Wegener, K., Pionteck, T., Saake, G., Teich, J., Wildermann, S.: Integration of fpgas in database management systems: Challenges and opportunities. *Datenbank-Spektrum* **18**(3), 145–156 (Nov 2018). <https://doi.org/10.1007/s13222-018-0294-9>
2. Bernstein, P.A., Chiu, D.M.W.: Using semi-joins to solve relational queries. *J. ACM* **28**(1), 25–40 (Jan 1981). <https://doi.org/10.1145/322234.322238>
3. Borovica-Gajic, R., Idreos, S., Ailamaki, A., Zukowski, M., Fraser, C.: Smooth scan: robust access path selection without cardinality estimation. *The VLDB Journal* **27**(4), 521–545 (2018). <https://doi.org/10.1007/s00778-018-0507-8>
4. Codd, E.F.: A relational model of data for large shared data banks. *Commun. ACM* **13**(6), 377–387 (Jun 1970). <https://doi.org/10.1145/362384.362685>
5. Dittrich, J., Nix, J.: The case for deep query optimisation. In: *CIDR 2020, 10th Conference on Innovative Data Systems Research*, Amsterdam, The Netherlands, January 12-15, 2020, Online Proceedings. [www.cidrdb.org \(2020\), http://cidrdb.org/cidr2020/papers/p3-dittrich-cidr20.pdf](http://cidrdb.org/cidr2020/papers/p3-dittrich-cidr20.pdf)
6. Graefe, G.: Query evaluation techniques for large databases. *ACM Comput. Surv.* **25**(2), 73–169 (Jun 1993). <https://doi.org/10.1145/152610.152611>

⁶ <https://db.apache.org/derby/>

7. Gurumurthy, B., Broneske, D., Drewes, T., Pionteck, T., Saake, G.: Cooking dbms operations using granular primitives. *Datenbank-Spektrum* **18**(3), 183–193 (Nov 2018). <https://doi.org/10.1007/s13222-018-0295-8>
8. He, B., Lu, M., Yang, K., Fang, R., Govindaraju, N.K., Luo, Q., Sander, P.V.: Relational query coprocessing on graphics processors. *ACM Trans. Database Syst.* **34**(4) (Dec 2009). <https://doi.org/10.1145/1620585.1620588>
9. Kemper, A., Eickler, A.: Speichermedien. In: *Datenbanksysteme*, pp. 211–212. Oldenbourg Wissenschaftsverlag GmbH (2013), 9th edition
10. Neumann, T.: Efficiently compiling efficient query plans for modern hardware. *Proc. VLDB Endow.* **4**(9), 539–550 (Jun 2011). <https://doi.org/10.14778/2002938.2002940>
11. Rich, B.: Oracle database reference, 12c release 1 (12.1) e17615-20 (2017), <https://docs.oracle.com/database/121/DWHSG/schemas.htm#DWHSG9069>
12. Sarawagi, S.: Database systems for efficient access to tertiary memory. In: *Proceedings of IEEE 14th Symposium on Mass Storage Systems*. pp. 120–126 (1995). <https://doi.org/10.1109/MASS.1995.528222>

Modeling Interdependent Preferences over Incomplete Knowledge Graph Query Answers

Till Affeldt¹[0000-0001-6440-5654], Stephan Mennicke²[0000-0002-3293-2940], and
Wolf-Tilo Balke¹[0000-0002-5443-1215]

¹ Institute for Information Systems, TU Braunschweig, Braunschweig, Germany
² Knowledge-Based Systems Group, TU Dresden, Dresden, Germany

Abstract. We study the properties of optimal patterns, our novel extension of SPARQL, that allows for a fine-grained control over incompleteness of query answers in knowledge graphs. Optimal patterns encode preferences over the completeness of query answers. In this paper, we add language constructs for expressing dependencies between conflicting preferences. Furthermore, we provide discussions and proofs of fundamental results concerning SPARQL with optimal patterns.

Keywords: knowledge graphs · structural preferences · SPARQL.

1 Introduction

One of the major difficulties in working with knowledge graphs is that entities may vastly differ in their structural representation, even if they are of the same kind. Indeed, due to limitations of the underlying sources and knowledge extraction processes it is quite common that in practical instances two entities (represented as nodes or resources) share a type, but may be characterized in totally different ways regarding their properties. The consequences are unexpectedly small or even empty result sets when querying basic graph patterns (conjunctive queries), usually followed by a lot of manual query refinements in the retrieval process. Thus, operators for handling such heterogeneity are mandatory when designing a robust query language for knowledge graphs. SPARQL [11] already offers *optional patterns*, which handle incompleteness in RDF graphs by returning complete matches (i. e., mandatory plus optional parts) and incomplete matches. Optional patterns and well-behaving subclasses of them have been heavily discussed over the last decade [12,13,2,6,4]. However, the diversity of result types remains unchanged. Suppose we query for three patterns A , B , and C :

A OPTIONAL B OPTIONAL C .

Then we may get results satisfying A alone, results that additionally satisfy B or C , and results that satisfy all three patterns. In fact, the number of result

Copyright © 2020 by the paper's authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

schemas can be exponential in the number of OPTIONAL operators in a query. We recently proposed *optimal patterns* [1], which come with a built-in preference semantics [8], so that only maximal results are returned. In the example above, we would not return results satisfying *A* alone when there are results that satisfy more patterns.

The new OPTIMAL operator for SPARQL enables the expression of structural as well as some value preferences. These preferences can be used to satisfy a user's information need if no perfect results can be found. For instance, a user may ask for a car with sunroof and air conditioner that costs less than 10,000€. If such configurations are not available in the data, cars with fewer features or higher prices are still relevant. Evaluating queries that use our OPTIMAL operator automatically retrieves the best possible answers. Thus, a user can receive useful information instead of having to manually tweak her search settings until a result can be found.

OPTIMAL positions itself between the rather restrictive query conjunction and the loose optional patterns of SPARQL. In our previous work [1] we have shown that the operator can be used in real-world scenarios to achieve fine-grained control over inherent incompleteness of RDF knowledge bases. It can also be used to model multiple preferences simultaneously, both of similar as well as prioritized relevance. Moreover, using optimal patterns results in readable and maintainable query representations. After we introduce some notational conventions in Sect. 2, we briefly recap syntax and semantics of optimal patterns in Sect. 3. In addition to what has been shown in our previous work, we discuss fundamental design decisions and properties of optimal patterns (Sections 3.1 and 3.2). In particular, we discuss how optimal patterns behave in open world knowledge bases like RDF.

Optimal patterns alone lack the ability to describe complex dependencies between preference patterns. As we will argue with concrete query examples in Sect. 4.1, we cannot express what we call *positive* and *negative dependencies*. For instance, the preferred color of a car may only be relevant for a certain type of car but irrelevant for others. Similarly, optimal patterns do not allow for modeling the next-best preference if one preference cannot be satisfied. Therefore, we extend our earlier work introducing structural preferences into SPARQL with a focus on retrieving *reasonably sized* result sets over incomplete data sets. We analyze how multiple preferences can affect each other leading to a characterization of different types of dependencies (Sect. 4). In order to cope with these different types, we then introduce a suitable set of operators to be used in conjunction with OPTIMAL.

Modeling preferences with SPARQL has been the goal of several language extensions [5,10,14,15]. However, all of these works focus on a preference model based on preferred data values, i. e., they allow for expressing certain preferences over the attribute values such as the lowest price for a car or a specific set of brands. But, while these extensions work quite well for value-based ranking and selection operations, users cannot express preferences with respect to the completeness of query results. Indeed, expressing a simple query such as *I am*

looking for cars, preferably including information on prices and/or brands will result in rather cumbersome expressions. And even if a user's preferences are not completely satisfiable in some given knowledge graph, he/she is still looking for cars and expects that cars are returned whenever possible. Another difference is that value preferences do usually not consider structural preference dependencies. This is because no structural dependencies can exist if the returned results are guaranteed to be structurally complete. In this case also the need for non-structural dependencies is greatly reduced: The same results can often be achieved by defining a priority order for preferences, in particular by evaluating independent preferences first. For a detailed discussion of other related approaches see [1].

2 Preliminaries

Since SPARQL is the recommended query language for the *Resource Description Framework* (RDF) [11], we base our notions on that of RDF. Therefore, assume infinite and disjoint universes of IRIs \mathbf{I} and literals \mathbf{L} . The core construct of RDF is that of an *RDF triple* $(s, p, o) \in \mathbf{I} \times \mathbf{I} \times \mathbf{IL}^3$. In such a triple, predicates (p) connect resources (s) to other resources or actual data values (o). Sets of RDF triples form *RDF graphs* \mathbf{G} .

SPARQL offers expressions for querying information from RDF graphs. We restrict our presentation to pattern matching capabilities, e. g., leaving out path queries. Abstract syntax and semantics of SPARQL follow the standard notation [11,12]. Let \mathbf{V} denote the universe of variables, e. g., $x, y, z \in \mathbf{V}$. A *triple pattern* is a triple $(u, a, v) \in \mathbf{VI} \times \mathbf{VI} \times \mathbf{VIL}$, i. e., variables $x \in \mathbf{V}$ may occur in every component of triple patterns. Sets of triple patterns \mathbb{G} also relate to graph-like structures, then called *basic graph patterns* (BGPs).

Let $t = (u, a, v)$ be a triple pattern and \mathbf{G} an RDF graph. A partial function $\mu : \mathbf{VIL} \rightarrow \mathbf{IL}$ is called a *match of t in \mathbf{G}* iff (a) $\mu(c) = c$ for all $c \in \mathbf{IL}$, (b) $\{u, a, v\} \subseteq \text{dom}(\mu)^4$, and (c) $(\mu(u), \mu(a), \mu(v)) \in \mathbf{G}$. By $\emptyset_{\mathbf{IL}}$ we denote the *empty match*, being the partial function that respects (a) but is otherwise undefined. The notion of a match naturally extends to BGPs \mathbb{G} by requiring that μ is a match of all triple patterns $t \in \mathbb{G}$. We denote the set of all matches of t (\mathbb{G} , resp.) in \mathbf{G} by $\llbracket t \rrbracket_{\mathbf{G}}$ ($\llbracket \mathbb{G} \rrbracket_{\mathbf{G}}$, resp.).

SPARQL further supports complex queries in terms of *unions*, *query conjunctions*, *optional patterns*, and *filter conditions*. The semantics of such queries \mathcal{Q} (w. r. t. RDF graphs \mathbb{G}) can be found in [12] and is denoted by $\llbracket \mathcal{Q} \rrbracket_{\mathbb{G}}$. The core notion of *compatibility* for query conjunctions and optional patterns is also required for a formal account of OPTIMAL. In general, two partial functions $\mu_1, \mu_2 : \mathbf{VIL} \rightarrow \mathbf{IL}$ are *compatible*, denoted $\mu_1 \asymp \mu_2$, iff for all $x \in \text{dom}(\mu_1) \cap \text{dom}(\mu_2)$, $\mu_1(x) = \mu_2(x)$, i. e., μ_1 and μ_2 agree on the variables they share. In conjunctions and optional patterns, only compatible matches to subpatterns are joined in the result sets.

³ We use \mathbf{IL} as shorthand for $\mathbf{I} \cup \mathbf{L}$.

⁴ $\text{dom}(\mu)$ refers to the set of all elements of \mathbf{VIL} for which μ is defined.

3 SPARQL with OPTIMAL

Throughout this section, we present our extension of SPARQL by *optimal patterns*. Additional examples, an encoding by standard SPARQL, as well as an initial evaluation may be found in [1]. Here, we give a brief account of the syntax and semantics. Furthermore, we discuss the properties of optimal patterns formally (cf. Sections 3.1 and 3.2).

As with the other SPARQL structures, OPTIMAL combines queries Q_1 and Q_2 to an *optimal pattern* Q_1 OPTIMAL Q_2 . The intuition of optimal patterns is that we state a preference for matches that are complete w. r. t. Q_1 and Q_2 , i. e., in the *optimal case*, we get the matches of Q_1 AND Q_2 . Only if this goal is impossible to reach, i. e., there could be matches of Q_1 but no compatible ones of Q_2 , we expect the matches of Q_1 . The semantics of optimal patterns will adhere to the following property:

$$\llbracket Q_1 \text{ AND } Q_2 \rrbracket_{\mathbf{G}} \subseteq \llbracket Q_1 \text{ OPTIMAL } Q_2 \rrbracket_{\mathbf{G}} \subseteq \llbracket Q_1 \text{ OPTIONAL } Q_2 \rrbracket_{\mathbf{G}} \quad (1)$$

More concretely, if there are matches of Q_1 AND Q_2 in \mathbf{G} , then $\llbracket Q_1 \text{ OPTIMAL } Q_2 \rrbracket_{\mathbf{G}} = \llbracket Q_1 \text{ AND } Q_2 \rrbracket_{\mathbf{G}}$. Unlike the respective optional pattern Q_1 OPTIONAL Q_2 , we do not include a single match of only Q_1 in this case. This matching behavior allows us to express preferences over the completeness of the matches w. r. t. a given RDF graph. Only if $\llbracket Q_1 \text{ AND } Q_2 \rrbracket_{\mathbf{G}} = \emptyset$, the respective semantics of optimal patterns and optional patterns align.

As long as Q_1 and Q_2 are free of optimal patterns, the previous paragraph tells us the whole story of the new operator. The combination of (several) optimal patterns with filter constraints allows for formulating meaningful preference patterns on data values and structure.

Example 1. OPTIMAL can be used to model preferences over attribute values. A simple price preference could be: *I prefer cheap cars under 15,000€ over other cars under 20,000€ over more expensive cars.* It can be adequately modeled in the form of: `?car rdf:typeof Car`

```
OPTIMAL { ?car ex:price ?price FILTER( ?price < 15000 ) }
```

```
OPTIMAL { ?car ex:price ?price FILTER( ?price < 20000 ) }
```

The query will return all cars that are cheaper than 15,000€ if any exist. If not, then it will return all cars that are cheaper than 20,000€ instead. If no such car exists either, then the query will return all cars in the data set that are more expensive or have no available price information.

Example 2. A user can pose structural preferences as well. Such a query could be *I prefer to see cars with available price and brand information.* Such a query could be adequately modeled as: `?car rdf:typeof Car`

```
OPTIMAL { ?car ex:price ?price }
```

```
OPTIMAL { ?car ex:brand ?brand }
```

The query will return all cars with information for price and brand. If no such cars can be retrieved, it will instead return all cars with price information. If that also results in an empty set, then the query will return all cars with brand

information instead. Only if all of these attempts yield empty results, then the query will return all cars instead.

The last example shows how optimal patterns may be used to impose an ordering on the preferred structure. To allow for the formulation of preferences of equal importance, we extended the syntax of optimal patterns Q_1 OPTIMAL Q_2 , allowing Q_2 to be a finite list of queries $Q_2^1, Q_2^2, \dots, Q_2^n$, i. e., the syntax of these optimal patterns is

$$Q_1 \text{ OPTIMAL } (Q_2^1, Q_2^2, \dots, Q_2^n).$$

The intuition is still that Q_1 is necessary to be matched. However, a simple correspondence as in Equation (1) cannot be derived. In the *optimal case*, we find matches of all of the queries in the optional pattern, i. e.,

$$\llbracket Q_1 \text{ AND } Q_2^1 \text{ AND } Q_2^2 \text{ AND } \dots \text{ AND } Q_2^n \rrbracket_{\mathbf{G}} \subseteq \llbracket Q_1 \text{ OPTIMAL } (Q_2^1, Q_2^2, \dots, Q_2^n) \rrbracket_{\mathbf{G}}.$$

Whenever we cannot reach the optimal case, we will obtain *dominant matches* as answers.

Example 3. Given the query from Example 2, a possible answer could be a car for a price of 17,000€ but of no known brand. This result contains more information than one with completely unknown attributes. Thus, the first match dominates the second one. A match concerning a third car with a price of 18,000€ and a known brand contains even more information on the other hand. Thus, the first match is dominated by the third one. If we consider price and brand to be of equal importance, then the first match would not dominate a fourth one that only has brand information. The two matches would be incomparable. If we consider the price to be more important instead, then the fourth match does not appear in the result set. The just sketched process is like asking for a skyline over the price (which does not include the fourth match) and then construct a skyline over the brand for the previous results.

Hence, the new construct requires a Pareto-style semantic, which entails a *skyline of matches*. Subsequently, we order candidate matches by *Pareto dominance*. Only the maximal candidates (w. r. t. Pareto dominance) are considered matches of optimal patterns.

Definition 1 (Candidate Matches). *Let Q be an optimal pattern, i. e.,*

$$Q = Q_0 \text{ OPTIMAL } (Q_1, Q_2, \dots, Q_k)$$

for some integer $k > 0$. A partial function $\mu : \mathbf{VIL} \rightarrow \mathbf{IL}$ is called a candidate match of Q in \mathbf{G} iff there are $\mu_0, \mu_1, \mu_2, \dots, \mu_k$, such that 1. $\mu_0 \in \llbracket Q_0 \rrbracket_{\mathbf{G}}$, 2. $\mu_i \in \llbracket Q_i \rrbracket_{\mathbf{G}} \cup \{\emptyset_{\mathbf{IL}}\}$ ($0 < i \leq k$), 3. $\mu_i \rightleftharpoons \mu_j$ for all $i, j \in \{0, 1, 2, \dots, k\}$, and 4. $\mu = \mu_0 \cup \mu_1 \cup \mu_2 \cup \dots \cup \mu_k$.

The first requirement accounts for the fact that Q_0 is the *necessary pattern* to be matched. Note, every other part of μ may be the empty match $\emptyset_{\mathbf{IL}}$. Using the separation of candidate matches μ of Q (in \mathbf{G}) into $\mu_0, \mu_1, \mu_2, \dots, \mu_k$, we say that μ covers Q_i ($0 \leq i \leq k$) iff $\mu_i \neq \emptyset_{\mathbf{IL}}$. Note, Q_0 must be covered by all candidate matches. We denote by $\text{cover}_Q(\mu)$ the set of all sub-queries

$\mathcal{Q}_j \in \{\mathcal{Q}_0, \mathcal{Q}_1, \mathcal{Q}_2, \dots, \mathcal{Q}_k\}$ covered by μ . Then the semantics of optimal patterns boils down to the maximal matches w.r.t. inclusion of the sets of covered sub-queries.

Definition 2 (Semantics of optimal patterns). *Let \mathcal{Q} be an optimal pattern, \mathbf{G} an RDF graph, and μ, μ' two candidate matches of \mathcal{Q} in \mathbf{G} . μ' dominates μ w.r.t. \mathcal{Q} , denoted by $\mu \prec_{\mathcal{Q}} \mu'$, iff $\text{cover}_{\mathcal{Q}}(\mu) \subsetneq \text{cover}_{\mathcal{Q}}(\mu')$.*

Candidate match μ of \mathcal{Q} in \mathbf{G} is called a match of \mathcal{Q} in \mathbf{G} iff there is no candidate match μ' of \mathcal{Q} in \mathbf{G} with $\mu \prec_{\mathcal{Q}} \mu'$. The set of all matches of \mathcal{Q} in \mathbf{G} is denoted by $\llbracket \mathcal{Q} \rrbracket_{\mathbf{G}}$.

This formalization allows us to derive correctness of (1) for the special case of optimal patterns with a single query as the right-hand side.

Proposition 1. *For all SPARQL queries \mathcal{Q}_1 and \mathcal{Q}_2 , (1) holds.*

Proof. Let μ be a match of \mathcal{Q}_1 AND \mathcal{Q}_2 . Then there are matches $\mu_i \in \llbracket \mathcal{Q}_i \rrbracket_{\mathbf{G}}$ ($i = 1, 2$) with $\mu = \mu_1 \cup \mu_2$ and $\mu_1 \sqsubseteq \mu_2$. Thus, μ is a candidate match of $\mathcal{Q}' = \mathcal{Q}_1$ OPTIMAL \mathcal{Q}_2 . Furthermore, every match of \mathcal{Q}_1 AND \mathcal{Q}_2 covers \mathcal{Q}_1 and \mathcal{Q}_2 in \mathcal{Q}' . Hence, μ cannot be dominated by any other match of \mathcal{Q}' in \mathbf{G} .

Let μ' be a match of $\mathcal{Q}' = \mathcal{Q}_1$ OPTIMAL \mathcal{Q}_2 . We need to show that μ' is a match of \mathcal{Q}_1 OPTIONAL \mathcal{Q}_2 . Towards a contradiction assume, μ' is not a match of \mathcal{Q}_1 OPTIONAL \mathcal{Q}_2 . Then $\mu' \in \llbracket \mathcal{Q}_1 \rrbracket_{\mathbf{G}}$ but there is a match $\mu'' \in \llbracket \mathcal{Q}_2 \rrbracket_{\mathbf{G}}$, such that $\mu' \sqsubseteq \mu''$. But then $\mu' \cup \mu''$ is a candidate match of \mathcal{Q}' and

$$\text{cover}_{\mathcal{Q}'}(\mu' \cup \mu'') = \{\mathcal{Q}_1, \mathcal{Q}_2\} \supsetneq \{\mathcal{Q}_1\} = \text{cover}_{\mathcal{Q}'}(\mu'),$$

which contradicts the assumption that μ' is a match of \mathcal{Q}' , i.e., μ' is not dominated by any other candidate match. \square

Subsequently, we justify the syntax style of query lists in optional patterns and we consider RDF's *open world assumption* in the context of optimal patterns.

3.1 Justifying Optimal Patterns

Our divergence from the standard binary operator structure of SPARQL may seem peculiar at first. However, this was a necessary step to allow for expressing preferences of equal importance. Suppose, we want to express a preference of query \mathcal{Q} over \mathcal{Q}_1 and \mathcal{Q}_2 , but \mathcal{Q}_1 and \mathcal{Q}_2 are equally important. Without the query list notation, we have several candidates using optimal patterns and standard SPARQL operators:

- \mathcal{Q} OPTIMAL (\mathcal{Q}_1 UNION \mathcal{Q}_2): This query returns all the matches of \mathcal{Q} , preferably joined with matches of \mathcal{Q}_1 or \mathcal{Q}_2 . However, as soon as \mathcal{Q}_1 and \mathcal{Q}_2 have a shared variable, we will hardly see any matches fulfilling all three sub-queries.
- \mathcal{Q} OPTIMAL (\mathcal{Q}_1 AND \mathcal{Q}_2): From this query we may expect only matches that fulfill all three sub-queries or only \mathcal{Q} . Hence, if \mathcal{Q}_1 cannot be matched, we will also not see matches that fulfill \mathcal{Q} and \mathcal{Q}_2 .

- (\mathcal{Q} OPTIMAL \mathcal{Q}_1) AND (\mathcal{Q} OPTIMAL \mathcal{Q}_2): This query is quite close to the desired Pareto query. Here, \mathcal{Q}_1 and \mathcal{Q}_2 are seen as independent through the use of query conjunction. For queries of that shape we may actually observe matches fulfilling all three sub-queries or only a subset of these. However, this query also accommodates some weird matching behavior. Suppose, \mathcal{Q}_1 and \mathcal{Q}_2 share a variable x , that does not occur in \mathcal{Q} , but \mathcal{Q}_1 and \mathcal{Q}_2 cannot simultaneously be matched in some RDF graph \mathbf{G} . Then the query above may easily yield no results at all, although \mathcal{Q} may be matched together with \mathcal{Q}_1 or \mathcal{Q}_2 separately.

Besides the possibilities above, it is always possible to unfold the Pareto query exponentially. However, such queries are hardly readable and maintainable. Therefore, we decided to expand SPARQL’s operator set in this respect.

3.2 Optimal Patterns and Certain Answers

How well do optimal patterns cope with RDF’s *open world assumption* (OWA). The OWA influences the way the query semantics is actually executed. We often see query semantics following the *certain answers* perspective [2]. A match is a *certain answer of a query \mathcal{Q} in some OWA database* iff the match is a match in every possible interpretation of the OWA database. Regarding RDF, every superset RDF graph \mathbf{H} of \mathbf{G} is a possible interpretation of \mathbf{G} . According to Arenas and Pèrez, the certain answers perspective for SPARQL is manifested by

$$\text{CERTAINANSWERS}(\mathcal{Q}, \mathbf{G}) := \bigcap_{\mathbf{H} \supseteq \mathbf{G}} \llbracket \mathcal{Q} \rrbracket_{\mathbf{H}}.$$

Iterating over all infinitely many extensions \mathbf{H} of \mathbf{G} is infeasible in practice. Therefore, handy characterizations can be proven. One such characterization is concerned with the *monotonicity* of the given query \mathcal{Q} . A query \mathcal{Q} is *monotone* iff for all RDF graph $\mathbf{G} \subseteq \mathbf{H}$, $\llbracket \mathcal{Q} \rrbracket_{\mathbf{G}} \subseteq \llbracket \mathcal{Q} \rrbracket_{\mathbf{H}}$, i. e., adding more information may only lead to more query results. Since more information may lead to different matches that cover more sub-queries, Arenas and Pèrez introduced the notion of *weak monotonicity*. A query \mathcal{Q} is *weakly monotone* iff for all RDF graphs $\mathbf{G} \subseteq \mathbf{H}$. $\mu \in \llbracket \mathcal{Q} \rrbracket_{\mathbf{G}}$ implies the existence of a $\mu' \in \llbracket \mathcal{Q} \rrbracket_{\mathbf{H}}$ with $\mu \subseteq \mu'$. It can easily be shown that in the case of (weakly) monotone queries \mathcal{Q} , $\text{CERTAINANSWERS}(\mathcal{Q}, \mathbf{G})$ coincides with $\llbracket \mathcal{Q} \rrbracket_{\mathbf{G}}$.

Unfortunately, not all queries containing optimal patterns are weakly monotone. For instance, the third query shape in Sect. 3.1 is not monotone. Consider \mathcal{Q}_1 and \mathcal{Q}_2 to share a variable x , that is not shared with \mathcal{Q} . Now construct \mathbf{G} in such a way that it fulfills \mathcal{Q} and \mathcal{Q}_1 but not \mathcal{Q}_2 . Disjointly add a minimal structure to \mathbf{G} that fulfills \mathcal{Q} and \mathcal{Q}_2 but not \mathcal{Q}_1 . Then we simultaneously match \mathcal{Q} and \mathcal{Q}_1 as well as \mathcal{Q} and \mathcal{Q}_2 , but the respective matches are necessarily in conflict in x . Therefore, the result set will easily be empty, where it used to be non-empty when \mathcal{Q}_2 was violated.

Hence, optimal patterns are neither better nor worse than optional patterns, at least w. r. t. certain answers. It is simply two different styles of querying and we have to let the user decide which style is appropriate in which scenarios.

4 Adding Dependencies over Preferences

In this section we propose a solution for modeling dependencies by introducing an additional set of operators that complement `OPTIMAL`. Preferences may exist in various different types and combinations. Being able to model such dependencies is necessary to adequately model complex user requests.

Sometimes a subquery preference simply cannot return useful results for a given preference unless another part of the query also returns valid matches. This is the case when a query asks for a possibly missing node that is only indirectly related to an enforced match. For example, a query may ask for a person’s car as well as their car’s model. The person’s car can be trivially retrieved. However, if that person does not own a car, it is impossible to give an adequate answer regarding the car’s model. We call this kind of dependency a *structural dependency* because these directly depend on the (in-)completeness in the knowledge graph structure. In contrast, we call all dependencies that relate to data values within the modeled preference as *non-structural dependencies*, e. g., a user may prefer the color red for sports cars but not necessarily for other cars as well.

If one preference cannot be answered unless the found answers satisfy a specified condition, i. e., also cover a more relevant preference, we speak of a *positive dependency*. The previously given examples fall into this category. We speak of negative dependencies if one preference should only be answered if a specified condition remains unsatisfied, i. e., a specific and more relevant preference is not covered. For example, a user may prefer the color blue for any car that is not a sports car with either no color preference, or a different one for sports cars. Negative dependencies also model different alternatives of unequal importance, e. g., a user interested in buying a car needs to know about the price of the object. Most likely, that user will prefer offers made in his own currency. If no suitable matches can be found, offers in foreign currencies may still be relevant.

Dependencies are not guaranteed to be 1:1 relations. Especially structurally dependent preferences often rely on the same condition, implying a 1:m relation. For example, asking for a person’s car and its properties will result in a structural dependency of all properties towards the retrieval of a matching car. When modeling multiple alternatives of unequal importance, the opposite behavior is true. For example, a car manual should only be displayed in an unknown language if it is not available in any language the user is fluent in. This yields an n:1 relationship between preferences. Mixed types of questions can also lead to general n:m relations between preferences which need to be accounted for.

4.1 Limitations of `OPTIMAL`

Expressing interdependent preferences is already possible using existing SPARQL operators which requires significant effort, even more so than individual preferences. By the new `OPTIMAL` operator we have developed a tool to ease this process. We have also shown how it can be used for individual preferences. However, it is still not ideal for modeling dependencies.

Let us assume a user is looking for a car, preferably a sports car. If it happens to be a sports car, then the user prefers it to be red – otherwise any color is acceptable. Simultaneously, the user also prefers cars with available price information. Using multiple optimal patterns in a naive way (as shown in Example 4) does not represent these requirements.

Example 4. The following query will look for all cars, preferably sports cars. Among those, it will look for red ones. This works well if the database contains any sports car.

```
?car rdf:typeof Car
OPTIMAL { ?car rdf:typeof SportsCar, ?car ex:hasPrice ?price }
OPTIMAL { ?car ex:hasColor "red" }
```

Additionally, the query also describes a dominance of red cars over non-red cars.

A more accurate model can be achieved by nesting one optimal pattern into another one, being a technique also commonly used when modeling interdependent optional patterns.

Example 5. The following query looks for sports cars with red color first and then combines it with other preferences. As a result, red sports cars will be preferred over other sports cars, and sports cars in general will be preferred over other cars.

```
?car rdf:typeof Car OPTIMAL {
  ?car rdf:typeof SportsCar OPTIMAL { ?car ex:hasColor "red" },
  ?car ex:hasPrice ?price }
```

This time, red cars that are no sports cars will no longer be preferred over other cars. So we managed to fix the underlying problem in our model. However, the model is still not correct. Because the color preference is evaluated first, red sports cars will always be ranked better than non-red sports cars. This also means that a red sports car without price will be ranked higher than a non-red sports car with a price. In our use case, these two preferences are meant to be incomparable.

Modeling preferences with optimal patterns alone causes side effects by introducing unwanted ranking of actually incomparable query answers. For sure, it is always possible to model them correctly using complex filter conditions (cf. encodings in [1]). However, our goal is to ease the modeling process. The required use of unintuitive filters is very similar to the original motivation. Thus, we would like to have a simple set of operators that lets a user define dependencies between preferences directly.

4.2 Syntax and Semantics of THEN and OTHERWISE

In order to enable an easy way of modeling dependencies, we introduce two new operators called THEN and OTHERWISE. Both operators can be used on the right-hand side of an OPTIMAL operator in order to fine-tune requirements of a preference's matching behavior. A THEN B defines a positive dependency of

B towards A, meaning that preference B should only be answered if preference A also yields a non-empty set of answers. A OTHERWISE B, on the other hand, defines a negative dependency of B towards A, meaning that preference B is only answered if preference A does return an empty set of answers.

For our OPTIMAL operator we have already introduced lists of preferences for simultaneous evaluation (cf. Sect. 3). We apply the same concept to both sides of the new operators. For a preference A and a set of dependent preferences B towards A, A THEN (b_1, b_2, \dots, b_m) applies a matching restriction for all dependent preferences $b_j \in B$. They will only be answered if preference A is covered as well. Similarly, A OTHERWISE (b_1, b_2, \dots, b_m) applies a negative dependency restriction to all $b_j \in B$. For a set of independent preferences A and dependent preference B towards A, (a_1, a_2, \dots, a_n) THEN B applies a matching restriction on B. It will only be answered if at least one $a_i \in A$ is covered. When modeling a restriction towards all preferences in A (instead of any member) a simple conjunction can be used in order to turn the conditions into a single one. This method is best suited for handling alternative routes that lead to similar objects. Likewise, (a_1, a_2, \dots, a_n) OTHERWISE B applies a matching restriction on B. It will only be answered if none of the preferences $a_i \in A$ are covered. If both A and B are a set of preferences then both the respective restrictions apply for all $b_j \in B$ towards all $a_i \in A$.

Just like the OPTIMAL operator, we define THEN and OTHERWISE as left-associative. That way, the syntax for all following preferences $B_i \in \{B_1, \dots, B_n\}$ of the form $A \odot_1 B_1 \odot_2 \dots \odot_n B_n$ (with $\odot_i \in \{\text{THEN}, \text{OTHERWISE}\}$) will always imply a relation towards A, making the query easier to read. Query groups and preference list separators can be used to model the order of execution explicitly.

Example 6. The following query adequately models the query we were looking for in Sect. 4.1 and thereby resolves our issues with optimal patterns.

```
?car rdf:typeof Car OPTIMAL
({ ?car rdf:typeof SportsCar } THEN { ?car ex:color "red" },
?car ex:price ?price )
```

Example 7. In the following query (with negative dependencies), we consider the price in US-Dollar as a fallback if no offers in Euros can be found. Finding both currencies is not better than only Euros, though.

```
?car rdf:typeof Car OPTIMAL
({ ?car ex:price_EUR ?p_eur } OTHERWISE
{ ?car ex:price_USD ?p_usd })
```

4.3 Encoding Dependencies

In our previous work [1], we have demonstrated that optimal patterns can be encoded using existing query operators in order to work under standard SPARQL 1.0 semantics. The same holds for the new set of operators. We have found two different styles of encodings – one using UNION and one using OPTIONAL. Since

the use of optional patterns follows more or less standard procedure, we restrict our presentation to the union-style encoding.

The encoding using UNION has proven to be more stable regarding performance. For this style we first construct a superset of desired results by combining all possible preference combinations. For every such combination we bind a fresh variable in the respective subquery to mark which combination an answer candidate is generated from. As a next step, we retrieve the results for every subquery a second time (but with changed variable names) inside an EXISTS-operator. That way, we can determine which combinations yield at least one candidate match. Lastly, we use FILTER-conditions to remove any answers that are dominated by at least one combination with a non-empty set of matches.

This method is easily changed to adapt for the new operators. When constructing the superset, we have to redefine what a *possible combination* is. Every THEN or OTHERWISE operation results in additional constraints on this set. A preference A that is dependent on B is not supposed to be matched unless a mapping for B also appears in the answer. Thus, any combination including mappings for A but not B has to be removed from the superset. Likewise, a preference A that has a negative dependency on B is only supposed to match if no mapping for B appears in the answer. Thus, any combination including mappings for both A and B has to be removed from the superset. Our prototypical implementation⁵ can easily be extended.

In terms of performance, we leave out an extensive analysis here. Usage of the new operators only results in removal of answer candidates. Thus, any encoded query using THEN or OTHERWISE will be shorter and most likely slightly faster than an independent preference.

5 Conclusion

We have presented a new set of operators for expressing structural preferences over the completeness of knowledge graph query results. The primary extension consists of so-called *optimal patterns*, which we introduced in [1]. Here, we added the possibility of stating semantic dependencies between conflicting preferences. Furthermore, we discussed fundamental design decisions.

Although our discussion regarding certain answers (cf. Sect. 3.2) attests optimal patterns to have a similar behavior as optional patterns, a detailed discussion about the expressive power/complexity may still reveal important differences. For future work, we plan to overcome the limitations we observed in [1] with our encoding approach of evaluating optimal patterns by implementing custom evaluation strategies that may even use Pareto-specific optimizations [3,9,7].

Acknowledgements. This work is partly supported by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) in project number 389792660 (TRR 248, Center for Perspicuous Systems) and Emmy Noether grant KR 4381/1-1 (DIAMOND).

⁵ available at Github: <https://github.com/ifis-tu-bs/optisparql>

References

1. Affeldt, T., Mennicke, S., Balke, W.T.: Preference-driven Control over Incompleteness of Knowledge Graph Query Answers. In: 12th ACM Web Science Conference 2020. WebSci'20, Southampton, United Kingdom, Association for Computing Machinery, New York, NY, USA (July 2020)
2. Arenas, M., Pérez, J.: Querying Semantic Web Data with SPARQL. In: Proceedings of the Thirtieth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems. pp. 305–316. PODS '11, ACM, New York, NY, USA (2011), event-place: Athens, Greece
3. Balke, W.T., Güntzer, U., Zheng, J.X.: Efficient Distributed Skylining for Web Information Systems. In: Bertino, E., Christodoulakis, S., Plexousakis, D., Christophides, V., Koubarakis, M., Böhm, K., Ferrari, E. (eds.) *Advances in Database Technology - EDBT 2004*. pp. 256–273. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg (2004)
4. Cheng, S., Hartig, O.: OPT+: A Monotonic Alternative to OPTIONAL in SPARQL. *Journal of Web Engineering* **18**(1), 169–206 (2019)
5. Guerousova, M., Polleres, A., McIlraith, S.A.: Sparql with qualitative and quantitative preferences. In: *OrdRing@ ISWC*. pp. 2–8 (October 2013)
6. Kaminski, M., Kostylev, E.V.: Beyond Well-designed SPARQL. In: Martens, W., Zeume, T. (eds.) *19th International Conference on Database Theory (ICDT 2016)*. Leibniz International Proceedings in Informatics (LIPIcs), vol. 48, pp. 5:1–5:18. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany (2016)
7. Keles, I., Hose, K.: Skyline Queries over Knowledge Graphs. In: Ghidini, C., Hartig, O., Maleshkova, M., Svátek, V., Cruz, I., Hogan, A., Song, J., Lefrançois, M., Gandon, F. (eds.) *The Semantic Web – ISWC 2019*. pp. 293–310. Lecture Notes in Computer Science, Springer International Publishing, Cham (2019)
8. Kießling, W.: Foundations of preferences in database systems. In: Proceedings of the 28th international conference on Very Large Data Bases. pp. 311–322. VLDB '02, VLDB Endowment, Hong Kong, China (2002)
9. Morse, M., Patel, J.M., Jagadish, H.V.: Efficient skyline computation over low-cardinality domains. In: Proceedings of the 33rd international conference on Very large data bases. pp. 267–278. VLDB '07, VLDB Endowment, Vienna, Austria (Sep 2007)
10. Pivert, O., Slama, O., Thion, V.: Sparql extensions with preferences: A survey. In: Proceedings of the 31st Annual ACM Symposium on Applied Computing. p. 1015–1020. SAC '16, Association for Computing Machinery, New York, NY, USA (2016). <https://doi.org/10.1145/2851613.2851690>
11. Prud'hommeaux, E., Seaborne, Andy: SPARQL Query Language for RDF. Tech. rep., W3C (2008), <https://www.w3.org/TR/rdf-sparql-query/>
12. Pérez, J., Arenas, M., Gutierrez, C.: Semantics and Complexity of SPARQL. *ACM Trans. Database Syst.* **34**(3), 16:1–16:45 (Sep 2009)
13. Schmidt, M., Meier, M., Lausen, G.: Foundations of SPARQL Query Optimization. In: Proceedings of the 13th International Conference on Database Theory. pp. 4–33. ICDT '10, ACM, New York, NY, USA (2010), event-place: Lausanne, Switzerland
14. Siberski, W., Pan, J.Z., Thaden, U.: Querying the semantic web with preferences. In: *International Semantic Web Conference*. pp. 612–624. Springer (2006)
15. Troumpoukis, A., Konstantopoulos, S., Charalambidis, A.: An extension of sparql for expressing qualitative preferences. In: *International Semantic Web Conference*. pp. 711–727. Springer (July 2017)

Towards Evolutionary, Domain-Specific Query Classification Based on Policy Rules

Peter K. Schwab and Klaus Meyer-Wegener

Friedrich-Alexander-Universität Erlangen-Nürnberg
{firstname.lastname}@fau.de, <https://www.cs6.tf.fau.eu/>

Abstract. Many devices like smart sensors produce a vast amount of data that are still commonly stored in relational databases and are being processed using SQL queries. This data is only useable if it is processed in a fashion that results in applicable information for the users posing these queries. Thus, it can be very supportive for them to assess other queries that have already processed the targeted data. This is not a simple exercise, as SQL allows alias names and various syntactic structures to express equivalent queries. A manual assessment is also hard to accomplish due to the amount of qualified queries. We present a framework for evolutionary SQL query classification. Based on the analysis of query logs, query metadata like schema lineage and result statistics are automatically derived. Our framework enables users to define domain-specific policy rules for automatic query classification based on the query metadata. Classification is done according to domain-specific, contextual attributes that can be defined evolutionary at runtime, together with the policy rules. The classification results enrich the query metadata.

1 Introduction

Hoarding vast amounts of data is no longer a big thing to undertake, for example by smart sensors in the context of Industry 4.0. Instead, the key task is to extract the desired information from the data for a particular purpose at a certain point in time [3]. Data is only useable when accessed in a fashion that results in applicable information for the users posing the accessing queries. Thus, it can be very supportive for them to assess other queries that already have accessed the targeted data set. Most data are still commonly stored in relational databases (DBs) and are being processed using SQL queries. Analyzing these queries regarding their kind of data access is not a simple exercise, as SQL allows alias names and various syntactic structures for equivalent queries (e. g. subquery instead of join). Query assessment can be supported by considering query metadata (QM). A manual assessment is often hard to accomplish because of the vast amount of qualified queries. In addition, most query-assessment results are not commonly accessible but only available as tacit knowledge in the heads of the resp. users.

Copyright © 2020 by the paper's authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

Problem Statement. We require novel approaches that support query assessment according to a certain processing context. They must ease the analysis of the SQL queries’ syntactical variety and support automation of the query classification. Users must be enabled to store their assessment results linked with the underlying queries in order to share them with other users.

Contribution. We present a framework for evolutionary, domain-specific SQL query classification based on policy rules. QM like schema lineage and result statistics are automatically derived based on the analysis of query logs. Our framework enables users to externalize their tacit knowledge into domain-specific policy rules for automatic query classification based on the QM. Classification results are stored in contextual QM attributes (QMAs). They can be used for further classification in other policy rules. The contextual QMAs as well as the policy rules can be defined evolutionary at runtime.

2 Policy-Based Query Classification

We provide a policy-based, automatic query classification [8] based on relational and graph-based data models holding QM [7].

Evolutionary Definition of Contextual Query Metadata. Examples for contextual QM are a query’s purpose, its compliance according to data-privacy directives, or its aptitude for hardware acceleration. So far, this type of QM was mapped to our relational model. We extend our multi-relational property-graph model [7] to enable the evolutionary definition of contextual QM at runtime. For every query, a graph with a root vertex holding a UID is modeled. This root can have several edges of type `hasContextualAttr` to vertices of type `contextualAttribute`. A vertex has two properties holding the `name` and the `value` of the resp. contextual QMA. In addition, a `contextualAttribute` vertex has exactly one edge `hasDataType` to a vertex of type `dataType`. To support a slender, user-centric set of data types that can be selected for contextual QMA, we orientate towards the requirements interchange format (ReqIF) as a standard for user-centric types that fulfill an end-user’s plain idea of data types [2] and provide *Boolean*, *String*, *Integer*, *Float*, *Timestamp*, and *Enumeration*, and lists of these data types.

Domain-Specific Policy Rules. Our policies are based on conditional rules. The Boolean expression in their antecedent part describes a query-processing pattern. We provide a domain-specific language to write it down [8]. Our query representation is independent from SQL syntax using the queries’ corresponding trees of relational algebra operators. A single tree covers many syntactical variants of semantically equivalent queries. This syntax-independent representation enables a more generic definition of patterns. A *basic* pattern p_{basic} is related to a single QM attribute and covers for example an accessed relation or schema attribute, a certain filter predicate, the related DB user, or a query’s runtime or number of result tuples. Schema lineage is resolved automatically. Up to now,

basic patterns have been combined by logical conjunction to a *complex* pattern p_{complex} . We extend the combination possibilities by adding Boolean operators (NOT, OR, XOR) and nesting via parentheses to create richer complex patterns.

Queries matching a complex pattern will be classified according to the rule’s consequent part. So far, this part was fixed on the contextual QMA *data-privacy compliance*. Based on our data-privacy use case [6], any policy rule classified a matching query q always as *non-compliant* (cf. List. 1, line 2).

We extend our policy rules’ consequent part and allow classification to arbitrary contextual QMAs. Now both contextual QMAs and policy rules can be defined at runtime. The assigned value v has to match the data domain of the related contextual QMA q_{mac} (cf. List. 1, line 5). When q is classified, its contextual QM is enriched with the classification result and the related policy IDs of all matching rules. This enables traceability of the classification process.

```

1  /* Status Quo of Policy-Rule Definition */
2  IF q.match( p_complex ) THEN q.classify( 'non-compliant' )
3
4  /* Evolutionary Policy Rules */
5  IF q.match( p_complex ) THEN q.classify( q_mac, v )

```

Listing 1. Status quo of our policy rules and the proposed extension.

3 Exemplary Classification Use Case

Up to now, our approach was tailored towards the use case of data-privacy compliance [6,7,8]. Examples for policy rules in this context are the prohibition of filters on certain personal data or the requirement of a minimum result size in order to prevent users from drawing conclusions on individuals by queries. We will motivate now another use case that is totally different to the present one in order to demonstrate that by enabling evolutionary contextual QMAs, policy-based query classification can be applied in arbitrary scenarios without the need of adapting our implementation.

To enable hardware-based acceleration of DB query processing, the project “Reconfigurable Data Provider (ReProVide)” provides a sophisticated storage solution based on field-programmable gate arrays (FPGAs) [1]. Its query optimization techniques consider the capabilities of the hardware for a scalable and highly performant near-data processing of Big Data [5]. ReProVide’s generic FPGA architecture offers a library of query-processing modules, which can be configured onto the FPGAs. So far, queries apt for near-data processing are selected manually based on tacit expert knowledge. The ReProVide system is a system on a chip (SoC) with its own storage [4]. Only queries accessing data that are stored there can be accelerated by the FPGAs. Filter operators, for example, can be accelerated at line rate. But there are different query-processing modules for filters, depending on the involved data type. Thus, assuming that the date dimension of the TPC-DS benchmark suite¹ is located on the RePro-

¹ <http://www.tpc.org/tpcds/>

Vide storage, a responsible DB administrator could first create a new contextual QMA at runtime with name `hardware-acceleration aptitude` and data type `List<Enumeration>` with the elements `{'filter (float)', 'filter (int)', 'filter (uint)', 'filter (boolean)', 'filter (string)', 'filter (date)', 'filter (timestamp)'}`. Then, the admin could create the policy rule shown in List. 2 which triggers automatic classification of queries containing filter operations on integers. The admin accordingly creates further policy rules covering filter operations on other data types. This means, a query containing several filter operations on different data types can be classified by different policy rules. Therefore, our contextual QMA was defined as a List. For example, the query in List. 3 will finally be classified as `hardware-acceleration aptitude = {'filter (int)', 'filter (string)'}`.

```

1  IF q.match(
2      restrictsOn('date_dim', 'd_year') OR
3      restrictsOn('date_dim', 'd_dow') OR
4      ...
5      restrictsOn('date_dim', 'd_last_dom')
6  )
7  THEN q.classify('hardware-acceleration aptitude',
8                  'filter (int)')
9  )

```

Listing 2. Example rule to classify queries apt for hardware acceleration.

```

1  SELECT d_year, d_dow
2  FROM   date_dim
3  WHERE  d_day_name="Monday" AND (d_year > 1900 OR d_moy > 4)

```

Listing 3. Example query that filters on data stored within the ReProVide system.

As ReProVide also allows hardware acceleration of projections and semi-joins, the enumeration’s elements of our contextual QMA can be extended by respective elements and new policy rules could be defined to enable classification of queries containing these operations. All of this can happen at runtime, without adapting our implementation. Furthermore, the authors of ReProVide also aim query-sequence optimization [4]. Our framework can also support this aim by defining additional contextual QMAs and policy rules – again at runtime.

4 Next Steps

We will elaborate the use case for hardware acceleration in more detail concerning our approach. Furthermore, we have to solve the problem of contradicting classification results based on conflicting policy rules. A prototypic implementation will give further information about the applicability of our evolutionary approach for arbitrary scenarios.

Acknowledgement: The authors would like to thank the anonymous reviewers for their valuable remarks.

References

1. Becher, et al.: Reprovide: Towards utilizing heterogeneous partially reconfigurable architectures for near-memory data processing. In: BTW, 18. Fachtagung des GI-Fachbereichs DBIS, Workshopband. LNI, vol. P-290, pp. 51–70. GI, Bonn (2019)
2. Ebert, et al.: ReqIF: Seamless requirements interchange format between business partners. *IEEE Softw.* **29**(5) (2012)
3. Lee, et al.: Recent advances and trends in predictive manufacturing systems in big data environment. *Manufacturing letters* **1**(1) (2013)
4. Lekshmi B. G., et al.: The ReProVide query-sequence optimization in a hardware-accelerated DBMS. In: 16th Int. Workshop DaMoN. pp. 17:1–17:3. ACM (2020)
5. Lekshmi B. G., et al.: SQL query processing using an integrated FPGA-based near-data accelerator in ReProVide. In: Proc. 23rd Int. Conf. EDBT. pp. 639–642. Open-Proceedings.org (2020)
6. Schwab, et al.: Query-driven enforcement of rule-based policies for data-privacy compliance. In: Proc. LWDA (2019)
7. Schwab, et al.: A framework for DSL-based query classification using relational and graph-based data models. In: Proc. Joint Wksh. GRADES-NDA. ACM (2020)
8. Schwab, et al.: We know what you did last session – policy-based query classification for data-privacy compliance with the DataEconomist. In: Proc. SSDBM (2020)

Author index

- Affeldt, Till 279
Althoff, Klaus-Dieter 130,142,154,174
Arif, Mofassir Ul Islam 47
Aßmann, Jule 203
Auge, Tanja 258
Balke, Wolf-Tilo 279
Balzereit, Kaja 180
Bartels, Jobst-Julius 130
Basoglu, Cem 70
Baumeister, Joachim 118
Beckh, Katharina 88
Beer, Anna 11
Behrens, Grit 70
Bergmann, Ralph 162, 192
Biertz, Manuel 23
Brell, Claus 234
Busch, Julian 11
Cichonczyk, Mario 218
Diehl, Matthias 70
Dietrich, Clarissa 192
Dumani, Lorik 23
Ehmüller, Jan 246
Eisenstadt, Viktor 174
Feistel, Susanne 258
Fischer, Raphael 6
Fullen, Marta 180
Giesselbach, Sven 76
Gips, Carsten 100,105,218
Grabocka, Josif 47
Grüger, Joscha 162
Haller, David 241
Harth, Maximilian 59
Heinz, Marcel 142
Heuer, Andreas 258
Jakobs, Matthias 6
Jameel, Mohsan 47
Jürgensmann, Susanne 258
Kazempour, Daniyal 11
Kazik, Yavuz 162
Kindermann, Jörg 88
Kirsch, Birgit 76
Kohlmeyer, Lasse 246
Korger, Andreas 118
Krestel, Ralf 246
Kreutz, Christin Katharina 23
Krieger, Rolf 59
Kuhn, Martin 162
Kühne, Maurus 35
Kuron, Ralf 234
Langenhan, Christoph 174
Lenz, Richard 241
Manthey, Erik 258
McKee, Holly 246
Mennicke, Stephan 279
Mertes, Konrad 70
Meyer-Wegener, Klaus 270,291
Morik, Katharina 6
Mücke, Sascha 6
Mülder, Wilhelm 234
Naumann, Felix 246
Niggemann, Oliver 180
Paeschke, Daniel 246
Raue, Benjamin 192
Repke, Tim 246
Reuss, Pascal 130
Rostalski, Frauke 76
Rüping, Stefan 76
Sauer, Joachim 203
Schenkel, Ralf 23
Schmidt-Thieme, Lars 47
Schmude, Timothée 76
Schoenborn, Jakob Michael 142,154
Schorr, Christian 59
Schriml, Sebastian 192
Schulz, Michael 203
Schwab, Peter K. 291
Seidl, Thomas 11
Steinkamp, Louis 100,105
Tekles, Alexander 11
Tödtli, Beat 35
Tymann, Karsten 100,105
Viefhaus, Sebastian 130
Vöhringer, Demian E. 270
Völkening, Malte 76
Witry, Alex 23
Yasrebi-Soppa, Philipp 130
Zhurakovskaya, Oxana 100,105