



GI-Edition



**Lecture Notes
in Informatics**

Gesellschaft für Informatik (Hrsg.)

SKILL 2019

**Studierendenkonferenz
Informatik**

**25. und 26. September 2019
Kassel**

GESELLSCHAFT
FÜR INFORMATIK



Gesellschaft für Informatik (Hrsg.)

SKILL – Studierendenkonferenz Informatik 2019

25./ 26. September 2019
Kassel

Gesellschaft für Informatik e.V. (GI)

Lecture Notes in Informatics (LNI) - Seminars

Series of the Gesellschaft für Informatik (GI)

Volume S 15

ISSN 1614-3213

ISBN 978-3-88579-449-3

Volume Editors

Gesellschaft für Informatik e.V.

Ahrstraße 45

53175 Bonn

E-Mail: bonn@gi.de

Redaktion: Michael Becker

E-Mail: mbecker@informatik.uni-leipzig.de

Series Editorial Board

Heinrich C. Mayr, Alpen-Adria-Universität Klagenfurt, Austria
(Chairman, mayr@ifit.uni-klu.ac.at)

Torsten Brinda, Universität Duisburg-Essen, Germany

Dieter Fellner, Technische Universität Darmstadt, Germany

Ulrich Flegel, Infineon, Germany

Ulrich Frank, Universität Duisburg-Essen, Germany

Michael Goedicke, Universität Duisburg-Essen, Germany

Ralf Hofestädt, Universität Bielefeld, Germany

Wolfgang Karl, KIT Karlsruhe, Germany

Michael Koch, Universität der Bundeswehr München, Germany

Thomas Roth-Berghofer, University of West London, Great Britain

Peter Sanders, Karlsruher Institut für Technologie (KIT), Germany

Andreas Thor, HFT Leipzig, Germany

Ingo Timm, Universität Trier, Germany

Karin Vosseberg, Hochschule Bremerhaven, Germany

Maria Wimmer, Universität Koblenz-Landau, Germany

Dissertations

Steffen Hölldobler, Technische Universität Dresden, Germany

Seminars & Thematics

Andreas Oberweis, Karlsruher Institut für Technologie (KIT), Germany

© Gesellschaft für Informatik, Bonn 2019

printed by Köllen Druck+Verlag GmbH, Bonn



This book is licensed under a Creative Commons BY-SA 4.0 licence.

Vorwort

Die Studierendenkonferenz Informatik (SKILL) ist eine jährliche Konferenz für Studentinnen und Studenten der Informatik sowie angrenzender Disziplinen aus ganz D-A-CH. Die Intention des SKILL-Konferenzformats ist es, sehr guten studentischen Arbeiten eine öffentliche Plattform zur Diskussion zu bieten. Die Studierenden können Erfahrungen zum wissenschaftlichen Publizieren sammeln und ihre Ergebnisse vor einem breiten Publikum vorstellen.

In diesem Jahr wurden insgesamt 21 Beiträge als Full oder Short Paper eingereicht und wissenschaftlich begutachtet. In diesem Band erscheinen zwölf Beiträge, die auf zwei Konferenztagen durch die Studierenden präsentiert wurden. Sie bilden eine breite Vielfalt an Themen und aktuellen wissenschaftlichen Diskussionen ab und beschäftigen sich u.a. mit natürlicher Sprachverarbeitung und Sicherheitsthemen bis hin zu Fragestellungen aus der Lehre.

Mit der SKILL 2019 wurde die seit 2014 bestehende Zusammenarbeit mit der Gesellschaft für Informatik e.V. (GI) weiter vertieft. Die Konferenz fand am 25. und 26. September 2019 im Rahmen der größten deutschsprachigen Informatik-Konferenz, der Jahrestagung der GI (INFORMATIK 2019), in Kassel statt. Die Teilnehmerinnen und Teilnehmer der SKILL waren durch Besuche der verschiedenen INFORMATIK-Keynotes sehr stark mit der Hauptkonferenz vernetzt.

Die Mitglieder des Organisationskomitees der SKILL 2019 bedanken sich zunächst bei den Autorinnen und Autoren, ohne deren qualitativ hochwertige Beiträge die Konferenz nicht möglich wäre. Wir freuen uns darüber hinaus, dass wir auch in diesem Jahr wieder namhafte Gutachterinnen und Gutachter gewinnen konnten, die den Studierenden mit hilfreichen und ausführlichen Kommentaren zu ihren Arbeiten zur Seite standen. Durch ihre Hilfe konnten alle eingereichten Beiträge von jeweils zwei ausgewiesenen Expertinnen und Experten begutachtet werden.

Die Mitglieder des Organisationskomitees bedanken sich für das große Interesse an der SKILL und laden alle Interessierten sehr herzlich zum anstehenden zehnjährigen Jubiläum 2020 ein.

Kassel, 25. September 2019

Organisationskomitee der SKILL 2019

- Michael Becker, Institut für Angewandte Informatik e.V.
- Judith Michael, Rheinisch-Westfälische Technische Hochschule Aachen
- Thomas Riechert, Hochschule für Technik, Wirtschaft und Kultur Leipzig
- Johannes Schmidt, Universität Leipzig
- Ludger Porada, Gesellschaft für Informatik e.V.

Gutachterinnen und Gutachter der SKILL 2019

- Michael Bader, Technische Universität München
- Matthias Book, University of Iceland
- Addis Dittebrandt, Karlsruhe Institut für Technologie
- Thomas Efer, Universität Leipzig
- Julia Friedrich, Institute for Applied Informatics (InfAI)
- Rüdiger Grimm, Universität Koblenz-Landau
- Volker Gruhn, Universität Duisburg-Essen
- Otthein Herzog , Tongji University, Shanghai
- Walter Hower, Hochschule Albstadt-Sigmaringen
- Martin Hübner, Hochschule für Angewandte Wissenschaften Hamburg
- Felix Hüning , FH Aachen
- Paul Jähne, Fraunhofer-Institut für Zelltherapie und Immunologie
- Christine Jakobs, Technische Universität Chemnitz
- Dietmar Jannach, Alpen-Adria-Universität Klagenfurt
- Friedbert Kaspar, Hochschule Furtwangen
- Stephan Klingner, Institut für Angewandte Informatik e.V.
- Jörn Kreutel, Beuth Hochschule für Technik Berlin
- André Langer, Technische Universität Chemnitz
- Jan-Patrick Lehr, Technische Universität Darmstadt
- Ludger Martin, Hochschule RheinMain Wiesbaden
- Alexander Mehler, Goethe-Universität Frankfurt am Main
- Lars-Peter Meyer, Universität Leipzig
- Helmut Neukirchen, University of Iceland
- Billy Naumann, Technische Universität Chemnitz
- Felix Neumeister, Karlsruhe Institut für Technologie
- Petra Sauer, Beuth Hochschule für Technik Berlin
- Peter Schartner, Alpen-Adria-Universität Klagenfurt
- Thomas Schmid, Universität Leipzig
- Andreas Schmidt, Universität des Saarlandes (Saarbrücken)
- Katja Wengler, Duale Hochschule Baden-Württemberg Karlsruhe
- Antonia Reißner, Technische Universität Chemnitz
- Juliane Krämer, Technische Universität Darmstadt
- Stefan Wagner, Universität Stuttgart
- Frank Grimm, FH Zwickau
- Patrick Struck, Technische Universität Darmstadt

Inhaltsverzeichnis

Natural Language Processing

Markus Glas

Social Relation Extraction from Chatbot Conversation 11

Marco Lehner, Anna Sauer, Christopher Schmidt, Lukas Schwarz

*Open Information Extraction gestützte Pipeline für einen
deutschsprachigen Wissensgraphen* 23

Dennis Debeye, Tim Eder, Paul Vincent Guigas, Viktoria Schuberth

*Improvement of automated social media sentiment analysis methods - a
context-based approach* 33

Informatik in der Lehre

Leonhard Applis

*Theoretical evaluation of the potential advantages of cloud IDEs for
research and didactics* 47

Timo P. Gros, Pascal L. Held, Pascal Lauer, Niklas O. Metzger, Kallistos Weis

aMAZEing Programming — Providing SKILLS to Fellow Students 59

Carolin Gümpel

Vorhersage der Ballbewegung in der RoboCup Standard Platform League 71

Algorithmik

Felix Welter

Datenbasierter Vergleich von statistischen Tests mithilfe von Simulationen 85

Iona Kuhn

<i>Heuristische Optimierung durch menschliche Intuition – Das Beste aus zwei Welten</i>	97
---	----

Maximilian Benedikt Schier, Niclas Wüstenbecker

<i>Adversarial N-player Search using Locality for the Game of Battlesnake .</i>	109
---	-----

Lukas Grätz

<i>Baumautomaten als notationelle Variante logischer Matrix-Semantik . . .</i>	121
--	-----

Security

Lisa Jasmin Moos

<i>Evaluation von Anti-Spoofing-Methoden mittels der Face Liveness Detection für die Verwendung in Web-Applikationen</i>	131
--	-----

Dominik Wagner

<i>Latent representations of transaction network graphs in continuous vector spaces as features for money laundering detection</i>	143
--	-----

Natural Language Processing

Social Relation Extraction from Chatbot Conversations

A Shortest Dependency Path Approach

Markus Glas¹

Abstract: Digital dialog systems, also known as chatbots, often lack in the sense of a human-like and individualized interaction. The ability to learn someone's social relations during conversations can lead to more personal responses and therefore to a more human-like and diverse conversation. In this work we present S-REX, a comparison method for extracting social relations from chatbot conversations. The implemented approach uses information from the shortest dependency path in combination with state-of-the-art natural language processing models for entity recognition and semantic word vectors. The method is evaluated on two conversational datasets and achieves results close to more complex neural network methods without the need of extensive training.

Keywords: Relation Extraction; Information Extraction; Chatbots; Information Retrieval; Text Mining; Natural Language Processing

1 Introduction

Digital dialog systems, also known as chatbots, have gained popularity through their usage in smart phones and smart speakers in recent years. Nevertheless, conversations with chatbots often lack in terms of natural human-like conversations and are often based on question and answering dialogs. The ability to learn someone's social relationships during conversations can help to create personalized responses and lead to a more human-like and diverse conversation.

The field of relation extraction (RE), as a subfield of information extraction (IE), has become an important task to automatically discern relationships among detected entities within unstructured text. Within the field of information extraction, several applications, most notably for optimizing web search engines, have pushed the development and research of new approaches during the last two decades. Nevertheless, most RE approaches have essentially focused on textual data in form of books or web site documents. A less considered area is the field of natural language conversations, used within chat messages between people, or humans and chatbots. The ambiguous, informal and often erroneous characteristic of natural language conversations requires specific adaptations on common approaches in

¹ Technische Hochschule Nürnberg Georg Simon Ohm, Fakultät Informatik, Hohlfederstraße 40, 90489 Nürnberg, Germany, glasma60952@th-nuernberg.de

RE. Therefore, it is necessary to evaluate new techniques upon the well-known approaches for extracting relationships. This work evaluates current approaches in RE and their usage within the domain of natural language conversations. Based on that, we present a new approach, called *S-REX*, for extracting social relations within chat conversations.

2 Relation Extraction

Relation extraction (RE), also known as relationship extraction or relation detection, describes the task of detecting and classifying semantic relations among named entities in texts [JM14]. These are commonly represented as binary relations between two entities such as *part-of*, *child-of*, *employed-at* or *located-in*. Relation extraction faces many challenges as relations are inherently ambiguous and the range of relations can vary from one domain to another. As other computational linguistic tasks, RE is also language-dependant.

Approaches for semantic relation extraction can be roughly assigned to one of the following four categories [Ba16, JM14]: Pattern-based, Supervised, Semi-Supervised or Distant Supervision, which are being discussed throughout the following sections.

2.1 Pattern-based Relation Extraction

Early and some of the commonly used algorithms use hand-written patterns for relation extraction, also known as pattern-based RE. First developed by Hearst [He92], these patterns try to infer the relationships between entities based on syntactic rules using regular expressions or similar. Consider the following sentence from Hearst's publication:

“The bow lute, such as Bambara ndang, is plucked and has individual curved neck for each string.”

Even without knowing the meaning of *Bambara ndang*, a human can infer that it is a kind of *bow lute*. This is true, even if the reader has only few conception about a *bow lute*.

A lexico-syntactic pattern like NP_0 *such as* $\{NP_1, NP_2 \dots, (and|or)\}$ NP_n can indicate this relation by recognizing noun phrase (NP) followed by the words *such as*, and one or more noun phrases, connected by *and* or *or*. Noun phrases are commonly composed of a noun (NN), with an optional preceding determiner (DT) and adjective (JJ). A regular expression for extracting noun phrases, based on part-of-speech (POS) tags, could be defined as follows: $<DT>?<JJ>*<NN>$.

Even if this method can reach a high precision on specific domains, it requires deep domain knowledge and lacks with respect to robustness and portability to other domains. Nevertheless, it is still a commonly used approach in some enterprise systems [CLR13].

2.2 Supervised Relation Extraction

Supervised relation extraction, based on supervised machine learning, follows the method of hand-annotating a text with a fixed set of relations as training corpus, and use them to train a classifier. In recent approaches, two classifiers are used to indicate if two entities are related to each other or not, before the final decision about the entity type is made. This allows to speed up the final classification and allows the use of distinct features. For each of the classifiers, standard classification techniques like Support Vector Machines, logistic regression, decision trees or similar, can be used [JM14].

Supervised relation extraction can achieve high accuracy, if enough hand-labelled data is available. But labelling large training sets is extremely time consuming and error-prone. Besides that, supervised models are very domain-specific and can not generalize to different text genres, that were not part of the training set [JM14]. Most research has therefore focused on semi-supervised approaches in recent years.

2.3 Semi-supervised Relation Extraction

Semi-supervised or bootstrapping approaches for RE try to tackle the problem of hand-labelling large training sets. Starting with a set of known relationships, bootstrapping algorithms try to generate training data by iterating over large texts and extracting training sentences based on predefined seeds.

One of the first published system using this technique was DIPRE, developed by Brin [Br98] specifically for the purpose of extracting books and their authors from web site documents (i.e. *[author, book]* pairs). It uses a 7 tuple *<author, book, order, url, prefix, suffix, middle>* as feature for every co-occurrence of a seed example, where *order* represents a boolean if the author occurs before the book, *prefix* and *suffix* the 10 words before and after the matched entities and *middle* all words in between them. All tuples are grouped by the *middle* and *order* features and DIPRE verifies if new tuples are the same. The new learned pattern is used to search the corpus again and extract new relationships.

Agichtein & Gravano [AG00] extended Brin's approach to a system called *Snowball*, which presents a more general approach for extracting further entity types from texts, than only books and authors. Similar to DIPRE, this approach forms tuples of the entities (*e1, e2*) and the words before (BEF), in between (BET) and after (AFT) them as context tuple: *<BEF, e1, BET, e2, AFT>*. Each context tuple is then represented by a TF-IDF vector, and similar contexts are clustered by a single-pass clustering, using the cosine similarity between the vectors.

Batista et al. [BMS15] updated the *Snowball* system with state-of-the-art word vector representations using word embeddings. Together with a prior selection of words between

the entities using hand-written patterns, this approach achieves a higher accuracy compared to previous systems.

The advantage of bootstrapping approaches lies in the omitted necessity of hand labelling large datasets to train a classifier. One downside of the examined bootstrapping approaches is the often limited usability to specific relation types, such as hypernyms (is-a) or meronyms (part-of) [GBM03, SJN05]. Another drawback of semi-supervised RE is the manual creation of seed examples, which requires domain specific knowledge.

2.4 Distant Supervision

Distant Supervision methods use knowledge bases, like *DBPedia*², *Wikidata*³ or formerly *Freebase*⁴ (offline since May 2016), containing semantic relations to automatically collect large amounts of seed examples. These seed examples are then used to extract training examples on large amounts of texts, similar to bootstrapping approaches.

Considering a system which tries to learn the relation type *place-of-birth*, instead of manually defining a small number of seed examples, *DBPedia* is used, containing over 100,000 examples for the relation type *place-of-birth*, like *<Albert Einstein, Ulm>*. Distantly supervised approaches then run a named entity tagger over large coherent texts, like Wikipedia Articles, and extract sentences containing both entities (e.g. “*Albert Einstein was born in Ulm...*”). The extracted sentences can be used to train a supervised classifier, similar to the supervised approaches described above.

Mintz et al. [Mi09] use *Freebase* as source for their seeds, which is a former semantic database trained on Wikipedia info boxes and other sources. They use several syntactic and lexical features, like sequence of words between the entities and part-of-speech tags, to train a multi-class logistic classifier.

Aljamel et al. [AOA15] used a combination of *DBPedia* and *Freebase* for extracting their seed tuples. They also used *JENA'S SPARQL* engine for extracting the mentioned relations within both databases through a common interface. For the classification task, they compared three methods: Support Vector Machine (SVM), Perceptron Algorithm Uneven Margin (PAUN) and K-Nearest-Neighbour (KNN). The SVM outperformed in both Person-Location (Per-Loc), Person-Organisation (Per-Org) relations and had an equal accuracy to PAUN for Location-Organisation (Loc-Org) relations.

² <https://wiki.dbpedia.org/>

³ <https://www.wikidata.org>

⁴ <https://developers.google.com/freebase/>

3 Social Relation Extraction

Social Relations describe relationships between two or more persons. Within this work, social relations are denoted as a finite number of relation types between two entities within a sentence. The entities are assigned to one of the following two categories: *Person* (*PER*) or *User* (*USR*). The type *Person* (*PER*) describes a person, recognized by a named entity recognition (NER) model. The entity type *User* (*USR*) describes the user interacting with the dialog system. This applies when the user mentions him- or herself through a possessive pronoun, like *my* or *I* (e.g. “<*USR*>My</*USR*> brother <*PER*>Michael</*PER*> is flying to London next week.”).

Entity pairs must occur in one of the following orders within a sentence, to be successfully extracted: *Person-to-Person* (PER-PER) or *User-to-Person* (USR-PER). Table 1 shows an example for both combinations.

No.	Example	Relations	Entities
1	“ Peter is the father of Tom .”	<Peter, father-of, Tom>	PER-PER
2	“ My daughter Lisa is moving to London next month.”	<Lisa, daughter-of, USER>	USR-PER

Tab. 1: Social relationships are being described as a relation between two specific types of entities, which can be related to each other in two different ways shown as rows in the table.

The relations between the two entities can be of one of the following types: *father-of*, *mother-of*, *son-of*, *daughter-of*, *brother-of*, *sister-of*, *grandfather-of*, *grandmother-of*, *grandson-of*, *granddaughter-of*, *husband-of*, *wife-of*, *uncle-of*, *aunt-of* or *friend-of*.

Extracted relations are denoted as a triple $\langle e_1, rel, e_2 \rangle$, where e_1 and e_2 represent the extracted entities, and rel describes a specific relationship type that connects the two entities.

3.1 Shortest Dependency Path Relation Extraction Approach

The method for shortest path RE is based on the feature extraction process, used by Bunescu et al. [BM05] to train a multi-class classifier. Other than in [BM05], the proposed method does not train a classifier due to the lack of conversational training datasets. It will rather use a distance measurement and compare the extracted features with predefined vectors of known relationship types. Besides that, it uses word embeddings instead of combined features made of POS-tags, entity types and other syntactic features.

3.2 Shortest Dependency Path Hypothesis

As proposed by Bunescu et al. [BM05], the contribution of the dependency path for establishing the relation between two entities, is almost exclusively concentrated in the

shortest path between these in the undirected version of the dependency path. If two entities are arguments of the same predicate (verb) and belong to the same predicate-argument structure, then the shortest path will pass through this predicate. Figure 1 shows an example representing this case, where the two entities represent the subject (nsubj - *protesters*) and object (dobj - *stations*), both related to the same predicate.

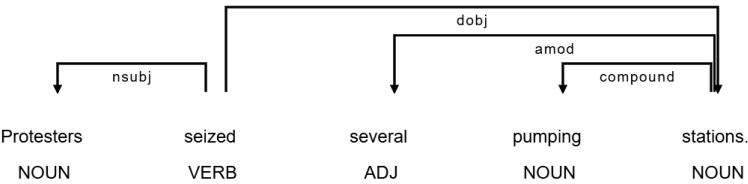


Fig. 1: Sentences with POS-Tags and dependency path

Sentence	Shortest path in undirected dependency graph
Protesters seized several pumping stations.	protesters ← seized → stations

Tab. 2: Shortest path representations of the sentence

Within the sentence shown in figure 1, the shortest dependency path (SDP) between the two noun entities *protesters* and *stations* passes through the predicate (verb) *seized*. In this basic examples, the predicate represents the only word on the shortest path, which might not be the case in more complex sentences. Nevertheless, even with more words on the shortest path, the predicate is always a part of it.

3.3 Shortest Path Hypothesis for Social Relations

The shortest path hypothesis can be used to extract social relationships between two entities, as the shortest path commonly passes through the mentioned relation. This is also true, if the relation is not mentioned between two entities or if other entities appear in between them. The sentence shown in the figure 2, holds a social relation between the entities *Monica* and *Ross*. Even though the relation is not in between these entities, it is possible to extract the relation by following the shortest path in the undirected dependency graph.

3.4 S-REX - Social Relationship Extraction

S-REX combines steps from different RE techniques, as described in section 2, and consists of eight steps: Recognize entities, extract dependency path, transform dependency into an undirected graph, search shortest path, get word embeddings for each word, sum up word embeddings, measure cosine similarity, extract relationships. For visualization purposes the

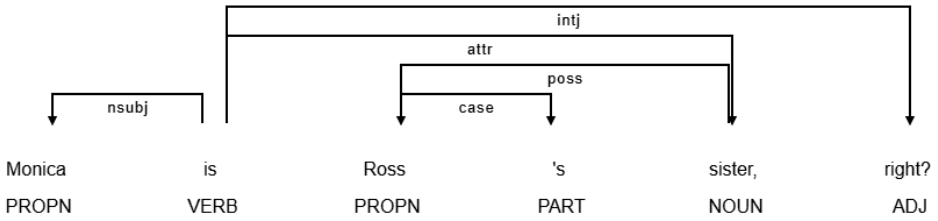


Fig. 2: Sentence containing social relations with POS-Tags and dependency path

sentence “*Monica is Ross’s sister, right?*”, as illustrated in figure 2, is used as an example throughout this section.

The first step consists of finding entities (*PER*, *USR*) within a sentence. *Person* (*PER*) entities are recognized by using a pre-trained sequence tagger model, provided by the open source library *Flair*⁵. The model is trained over the *CoNLL-03* dataset and offers a 4-class named entity recognition. The architecture of the sequence labelling follows a BiLSTM-CRF approach, as described in [ABV18]. *USER* (*USR*) entities are recognized by using a predefined list.

For creation of the dependency path (step 2), the pre-trained model *en_core_web_md* for English, provided by the open source NLP library *spaCy*⁶, is used. The model is trained on the sources *CommonCraw*⁷ and *OneNotes 5*⁸, and consists of 20,000 unique word vectors and 685,000 keys.

The extracted dependency path is then transformed into an undirected graph containing the words as nodes and dependencies as edges (see Figure 3). As the entities are already known from the first step, the next step is to search for the shortest path between each entity pair using Dijkstra’s algorithm. The entity pairs are constructed, following their appearance within a sentence from left to right, so that the leftmost entity is connected to each of following entities to the right. For instance, considering three entities e_1, e_2, e_3 within a sentence, the three entity pairs $\langle e_1 - e_2 \rangle$, $\langle e_1 - e_3 \rangle$ and $\langle e_2 - e_3 \rangle$ are being constructed.

Following the sentence of the example, two named entities are recognized, resulting in one entity pair $\langle \textit{Monica} - \textit{Ross} \rangle$. Using the shortest path algorithm, the words *is* and *sister* are extracted (step 4; see Table 3).

In the fifth step, for each word on the shortest path the corresponding word embedding representation is looked-up in a word embedding dictionary. The dictionary used here,

⁵ <https://github.com/zalandoresearch/flair>

⁶ <https://spacy.io/>

⁷ <http://commoncrawl.org/>

⁸ <https://catalog.ldc.upenn.edu/LDC2013T19>

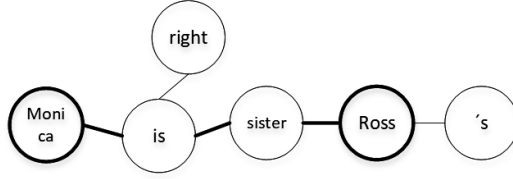


Fig. 3: Example of a dependency path within a sentence containing two entities.

Sentence	Shortest path in undirected dependency graph
Monica is Ross's sister, right?	Monica \leftarrow is \rightarrow sister \rightarrow Ross

Tab. 3: Shortest path representations of the sentence

contains vector representations of *GloVe*⁹ word embeddings, trained over Wikipedia provided by the *FastText*¹⁰ library. Each word is represented by a 100-dimensional word vector. Words which are not part of the dictionary, will be represented by a zero vector.

All word vectors of the shortest path are summed up into a single 100-dimensional vector (step 6), further denoted as *shortest path vector* (\vec{sp}). This vector is compared (step 7) to a set of stored *relation vectors* (\vec{rel}), each representing one of the predefined relation types. This results in a word vector for each of the following 15 words: *father*, *mother*, *sister*, *brother*, *son*, *daughter*, *husband*, *wife*, *grandson*, *granddaughter*, *grandmother*, *grandfather*, *uncle*, *aunt* and *friend*.

The *relation vector* with the highest cosine similarity, defines the relation type between the entities (step 8). The similarity is depicted by a single float value and needs to exceed a predefined distance threshold of 0.6 for successful RE. Similarities below this threshold are not extracted as social relationships. The similarity is measured by using the cosine similarity, as denoted in Equation 1, where \vec{sp} represents the shortest path vector and $\vec{rel}(n)$ a relation vector ($1 \leq n \leq 15$).

$$similarity = \cos(\theta) = \frac{\vec{sp} \cdot \vec{rel}(n)}{\|\vec{sp}\| \cdot \|\vec{rel}(n)\|} \quad (1)$$

Figure 4 illustrates a two-dimensional plot of *relation vectors* (blue) and the *shortest path vector* (red) between the entities *Monica* and *Ross* from the example. The closest vector in this case is the word vector for representing *sister*.

⁹ <https://nlp.stanford.edu/projects/glove/>

¹⁰ <https://fasttext.cc/docs/en/crawl-vectors.html>

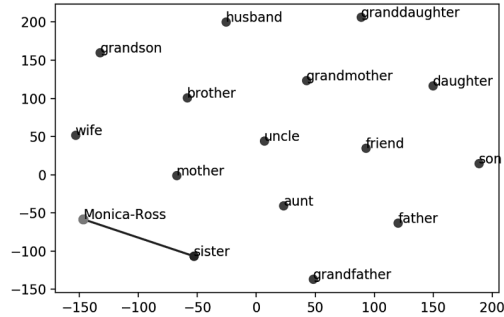


Fig. 4: Word Vector Space of 100-dimensional word embeddings projected to a 2-dimensional space using t-Distributed Stochastic Neighbour Embedding (t-SNE). The closest distance from vector *Monica-Ross* is the vector for representing the word *sister*.

4 Experimental Evaluation

The experiments are conducted on a dataset, built as a combination from utterances of the *Persona-Chat*¹¹ dataset and the *Friends TV Corpus*¹².

4.1 Datasets

The *Persona-Chat* project [Zh18] was created by collecting 162,064 utterances between crowdworkers via *Amazon Mechanical Turk*. The crowdworkers were randomly paired and told to chat naturally and get to know each other during conversation. This dataset is also known as a training set for the Conversational Intelligence Challenge 2 (ConvAI2).

The *Friends TV Corpus*, first introduced by Nio et al. [Ni14], was constructed by using subtitles of the popular American *Friends*¹³ sitcom. It contains subtitles from five seasons, with a total of 112 episodes. Each episode contains several scenes with several dialog turns. The dataset consists of 60,849 dialogs in total.

Both datasets contain sentences in a natural non-upscale language. The *Persona-Chat* corpus mainly contains *chat-style* utterances with an unclear spelling, including typing errors, chat-slang notations and missing punctuation. The *Friends TV Corpus* contains less spelling and grammatical errors, but, different from the *Persona-Chat* corpus, also contains mentions of persons. To evaluate the extraction of all types of defined entities (*PER* and

¹¹ <https://github.com/facebookresearch/ParlAI/tree/master/parlai/tasks/convai2>

¹² <https://github.com/npow/friends-chatbot>

¹³ <https://en.wikipedia.org/wiki/Friends>

USR), a combination of both datasets is used containing 500 randomly chosen utterances from each dataset. The combined dataset consists of 2313 sentences, which is an appropriate amount compared to other relation extraction datasets, such as the popular *SemEval-2010 Task 8* [He10], which has 2717 utterances in its test set.

4.2 Results

The 2313 sentences of the dataset contain 47 sentences with social relations. Within this category, seven sentences returned no relation and 4 returned a false or incomplete relation by the relation extraction system. This results in 11 false and 36 correct extracted sentences resulting in an accuracy of 76.6% and an F1 score of 79.6.

Sentences like “*my dad flies airplanes.*” or “*That’s what my mom said.*” could be correctly extracted, even though the relation types *dad* and *mom* are not explicitly defined inside the relationship list. This is possible due to the use of semantic word vectors. Other sentences like “*I exercise at home with my gf*” or “*i actually have four sisses too !*” could not be extracted, due to unrecognized short notation or the use of slang. Besides that, the named entity recognizer might not recognize every person as such and sometimes falsely identifies other words as a person.

4.3 Comparison with other methods

An exact comparison with other methods is not possible, as most academic research on relation extraction focuses on general relation extraction tasks using supervised methods, trained on annotated corpora. State-of-the-art approaches use different methods including distant supervision, dependency path or end-to-end models. We compare our model with other models, trained on the *SemEval-2010 Task 8* [He10] as it is one of the most popular in current research. Current state-of-the-art results are achieved by Wu et al. [WH19] with an F1 of 89.25 by leveraging a pre-trained end-to-end model from the BERT [De18] language model. Cai et al. [CZW16] use a combination of convolutional neural networks with a two-channel recurrent neural network using long-short term memory (LSTM) and reach an F1 of 86.3. Xu et al. [Xu16] use deep recurrent neural networks for relation classification with a F1 of 86.1. Both of the latter approaches make use of information along the SDP.

All of the mentioned supervised neural network approaches reach a F1 score of over 86 and outperform S-REX. However, it should be noted that S-REX does not train a model and therefore can be instantly used without the need of a large annotated training set. Beyond that, it is evaluated on a different training set including different (only social) relation types.

5 Conclusion

In this work we present S-REX, a method for extracting social relationships from natural language conversations. The proposed method shows robustness in extracting social relations from conversational data, even if the grammatical structure is not correct or spelling errors are present. Additionally, different notations of relations can be extracted, due to the use of semantic word vectors. S-REX is also able to extract relations mentioned after two entities within a sentence. In future work, the presented method can be improved by expanding the number of features and taking relations across sentence boundaries into account, using coreference resolution.

References

- [ABV18] Akbik, A.; Blythe, D.; Vollgraf, R.: Contextual String Embeddings for Sequence Labeling. In: Proceedings of the 27th International Conference on Computational Linguistics. Association for Computational Linguistics, Santa Fe, New Mexico, USA, pp. 1638–1649, August 2018.
- [AG00] Agichtein, E.; Gravano, L.: Snowball: extracting relations from large plain-text collections. In: Proceedings of the fifth ACM conference on Digital libraries - DL '00. ACM Press, San Antonio, Texas, United States, pp. 85–94, 2000.
- [AOA15] Aljamel, A.; Osman, T.; Acampora, G.: Domain-Specific Relation Extraction - Using Distant Supervision Machine Learning:. In: Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management. SCITEPRESS - Science and Technology Publications, Lisbon, Portugal, pp. 92–103, 2015.
- [Ba16] Batista, D. S.: Large-Scale Semantic Relationship Extraction for Information Discovery. PhD Thesis, INSTITUTO SUPERIOR TÉCNICO, 2016.
- [BM05] Bunescu, R. C.; Mooney, R. J.: A Shortest Path Dependency Kernel for Relation Extraction. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing. HLT '05, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 724–731, 2005.
- [BMS15] Batista, D. S.; Martins, B.; Silva, M. J.: Semi-Supervised Bootstrapping of Relationship Extractors with Distributional Semantics. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Lisbon, Portugal, pp. 499–504, September 2015.
- [Br98] Brin, S.: Extracting Patterns and Relations from the World Wide Web. In: WebDB. pp. 172–183, 1998.
- [CLR13] Chiticariu, L.; Li, Y.; Reiss, F. R.: Rule-Based Information Extraction is Dead! Long Live Rule-Based Information Extraction Systems! In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Seattle, Washington, USA, pp. 827–832, October 2013.

- [CZW16] Cai, R.; Zhang, X.; Wang, H.: Bidirectional Recurrent Convolutional Neural Network for Relation Classification. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Berlin, Germany, pp. 756–765, 2016.
- [De18] Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 [cs], October 2018. arXiv: 1810.04805.
- [GBM03] Girju, R.; Badulescu, A.; Moldovan, D.: Learning semantic constraints for the automatic discovery of part-whole relations. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1. Association for Computational Linguistics, pp. 1–8, 2003.
- [He92] Hearst, M. A.: Automatic Acquisition of Hyponyms from Large Text Corpora. In: COLING 1992 Volume 2: The 15th International Conference on Computational Linguistics. Association for Computational Linguistics, 1992.
- [He10] Hendrickx, I.; Kim, S. N.; Kozareva, Z.; Nakov, P.; Seaghdha, D. O.; Pado, S.; Pennacchiotti, M.; Romano, L.; Szpakowicz, S.: SemEval-2010 Task 8: Multi-way Classification of Semantic Relations Between Pairs of Nominals. In: Proceedings of the 5th International Workshop on Semantic Evaluation. SemEval '10, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 33–38, 2010. event-place: Los Angeles, California.
- [JM14] Jurafsky, D.; Martin, J. H.: Speech and language processing, volume 3. Pearson Education International, Upper Saddle River, NJ, 2 edition, 2014.
- [Mi09] Mintz, M.; Bills, S.; Snow, R.; Jurafsky, D.: Distant supervision for relation extraction without labeled data. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - ACL-IJCNLP '09. volume 2, Association for Computational Linguistics, Suntec, Singapore, p. 1003, 2009.
- [Ni14] Nio, L.; Sakti, S.; Neubig, G.; Toda, T.; Adriani, M.; Nakamura, S.: Developing non-goal dialog system based on examples of drama television. In: Natural Interaction with Robots, Knowbots and Smartphones, pp. 355–361. Springer, 2014.
- [SJN05] Snow, R.; Jurafsky, D.; Ng, A. Y.: Learning syntactic patterns for automatic hypernym discovery. In: Advances in neural information processing systems. pp. 1297–1304, 2005.
- [WH19] Wu, S.; He, Y.: Enriching Pre-trained Language Model with Entity Information for Relation Classification. CoRR, abs/1905.08284, 2019.
- [Xu16] Xu, Y.; Jia, R.; Mou, L.; Li, G.; Chen, Y.; Lu, Y.; Jin, Z.: Improved Relation Classification by Deep Recurrent Neural Networks with Data Augmentation. arXiv:1601.03651 [cs], January 2016. arXiv: 1601.03651.
- [Zh18] Zhang, S.; Dinan, E.; Urbanek, J.; Szlam, A.; Kiela, D.; Weston, J.: Personalizing Dialogue Agents: I have a dog, do you have pets too? arXiv:1801.07243 [cs], January 2018.

Open Information Extraction gestützte Pipeline für einen deutschsprachigen Wissensgraphen

Marco Lehner,¹ Anna Sauer,² Christopher Schmidt,³ Lukas Schwarz⁴

Abstract: Eine zentrale Herausforderung bei der Erstellung von Wissensgraphen aus natürlich-sprachigen Texten besteht darin, geeignete Werkzeuge für unterschiedliche Sprachen zu entwickeln. Besonders abseits des Englischen sind einsatzfähige Architekturen Mangelware. In diesem Paper stellen wir eine mögliche Pipeline vor, die auf Basis von Open Information Extraction (OIE) einen RDF/OWL-Wissensgraphen aus deutschen Texten extrahiert. Dabei verbinden wir verschiedene bestehende Werkzeuge zur Natürlichen Sprachverarbeitung miteinander, die eigens für die deutsche Sprache konstruiert wurden. Während die Relation Extraction zum Großteil auf Dependency Parsing basiert, konzentrieren wir uns bei der Entity Extraction mithilfe von Named Entity Recognition auf Eigennamen, vor allem von Personen.

Keywords: Wissensgraph; Open Information Extraction; Natürliche Sprachverarbeitung; Semantic Web

1 Motivation

Wissensgraphen wie Googles Knowledge Graph, YAGO, Wikidata oder DBpedia stellen Versuche dar, für das Semantic Web und seine Anwendungen eine große Wissensbasis über Entitäten (Personen, Orte, Organisationen) und ihren Beziehungen aufzubauen [EW16]. Eine der Methoden zur Erstellung von Wissensgraphen ist die Open Information Extraction (OIE): Dabei werden Informationen mithilfe Natürlicher Sprachverarbeitung automatisiert aus Texten extrahiert. Die OIE benötigt dabei im Gegensatz zur klassischen Information Extraction keine manuell annotierten Trainingsdaten [Ba07].

Allerdings muss die Vorgehensweise bei der OIE an die Sprache der gewählten Textquellen angepasst werden [Et11]. Während für das Englische [Ga17] und weitere Sprachen wie Chinesisch [Wa15] oder Arabisch [KL17] bereits Ansätze existieren, um auf OIE basierend Wissensgraphen zu generieren, bestehen für das Deutsche noch keine uns bekannten Versuche in dieser Richtung. In diesem Paper stellen wir daher die Pipeline zum Aufbau eines Wissensgraphen aus deutschsprachigen Texten vor, die wir im Rahmen eines einsemestrigen Projekts an der Universität Bamberg entwickelt haben.

¹ Universität Bamberg, marco.lehner@posteo.net

² Universität Bamberg, auersanna@googlemail.com

³ Universität Bamberg, c.schmidt914@gmail.com

⁴ Universität Bamberg, lukas.schwarz@posteo.de

2 Verwandte Arbeiten

	Unsere Pipeline	FRED [Ga17]	KParser [Sh15]	AWAKE [Bo14]
Extraktionsart	Erstellung von OIE-Tupeln mithilfe von PropsDE [Fa16]	Ermitteln von Diskursrepräsentationsstrukturen mit Boxer [Bo08]	Ableitung semantischer Rollen aus Dependency Trees	Extraktion von wenigen Entitätstypen, Relationen und Quantitäten mittels BBN SERIF [Ra11]
Pipelineaufbau	<ul style="list-style-type: none">• Part of Speech Tagging• Coreference Resolution• Named Entity Recognition• Entity Linking• Relation Extraction	<ul style="list-style-type: none">• Named Entity Recognition• Coreference Resolution• Einführen einer Taxonomie• Relation Extraction• Semantic Role Labeling• Frame/Situation Extraction• Entity Linking• Validierung	<ul style="list-style-type: none">• Dependency Parsing• Mapping auf Relationen der Knowledge Machine Ontologie [CPW04]• Entity Linking• Semantic Role Labeling	<ul style="list-style-type: none">• Dependency Parsing• Mention Detection• Coreference Resolution• Relation Extraction
Art des Graphen	Entitätszentrisch	Semantikorientiert	Semantikorientiert	Entitätszentrisch
Sprachunabhängigkeit	Nur Deutsch	Nur Englisch, anwendbar auf andere Sprachen dank BING-Übersetzung	Nur Englisch	Nur Englisch

Tab. 1: Vergleich bestehender Pipelines zur Wissensextraktion mit unserer Eigenentwicklung.

In Tabelle 1 werden vier Systeme zur Informationsextraktion nebeneinandergestellt. Die Art des Graphen unterteilen wir dabei in die Kategorien entitätszentriert und semantikorientiert. Entitätszentrierte Graphen fokussieren sich auf in den Texten enthaltene Entitäten und verbinden diese mit einer vergleichsweise geringen Anzahl an Prädikaten. Semantikorientierte Graphen versuchen die semantischen Rollen in einem Satz möglichst granular darzustellen und so den einzelnen Satz zu repräsentieren. Hier werden wenige Entitäten über vielen Kanten verbunden.

3 Lösungsansatz

Unsere Pipeline besteht aus folgenden Verarbeitungsschritten:

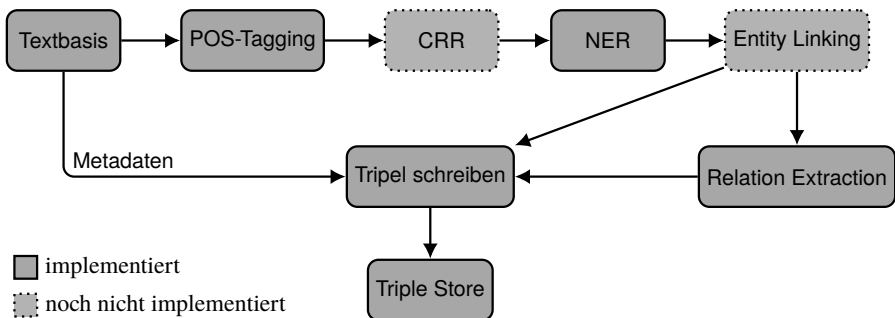


Abb. 1: Gesamtstruktur der Pipeline

Zunächst muss eine **Textdatenbasis** geschaffen werden. Je nach späterem Anwendungszweck des Wissensgraphen kann es dabei nützlich sein, **Metadaten zu den einzelnen Texten zu extrahieren**, um jedes verarbeitete Dokument im Wissensgraphen zu repräsentieren und mit den jeweils extrahierten Tripeln zu verknüpfen. Dies gewährleistet zudem die spätere Verifizierbarkeit der Tripel über einen Rückgriff auf die Textquelle.

Im nächsten Schritt wird durch **Part of Speech (POS) Tagging** jedem Wort der Texte seine jeweilige Wortart zugewiesen. Auf diesem sehr grundlegenden Verarbeitungsschritt bauen viele der weiteren Pipeline-Etappen auf.

Anschließend werden für die **Entity Extraction** mithilfe von **Named Entity Recognition (NER)** Eigennamen in den Texten identifiziert, die auf Personen, Organisationen, Orte etc. referieren und sich daher für die Aufnahme als Entitäten in den Graphen eignen. Wenn man zuvor noch durch **Coreference Resolution (CRR)** Verweise auf Named Entities z. B. von Pronomina ausgehend auflöst (z. B. „*Andreas Starke* ist Bürgermeister von Bamberg. *Er* gehört der SPD an.“), erschließt dies eine große Anzahl an zusätzlichen Informationen.

Als nächstes wird beim **Entity Linking** jeder Entität ein eindeutiger Identifikator zugeordnet und geprüft, ob sich der gleiche Name auf verschiedene Entitäten bezieht. Um den

Wissensgraphen im Sinne des Linked Open Data Prinzips mit anderen Ressourcen des Semantic Web zu verknüpfen, ergibt es Sinn, sich bei der Wahl der Identifikatoren auf bereits bestehende, umfangreiche Ontologien wie Babelnet oder Wikidata zu stützen.

Bei der anschließenden **Relation Extraction** werden durch Dependency Parsing Relationen zwischen den Entitäten in einem Satz identifiziert. Diese Relationen liegen dann in Tupelform mit Subjekt, Prädikat und Objekt(en) sowie eventuellen Modifikatoren vor. Ein ausführliches Beispiel dazu folgt im Abschnitt zur Implementierung.

Zum Abschluss müssen aus jedem Tupel noch ein oder mehrere **RDF-Tripel** gebildet werden. Jedes dieser Tripel besteht aus einem Prädikat, einem Subjekt und einem Objekt (z. B. in informeller Notation `pred:besuchen`, `subj:Angela Merkel`, `obj:Emmanuel Macron`). Zudem werden die Tupel der dem Graphen zugrundeliegenden Ontologie angepasst, die unter anderem festlegt, welche Typen von Relationen konkret im Graphen modelliert werden. Das jeweilige Tupelprädikat wird dabei in eine der in der Ontologie vorhandenen generischen Relationen umgewandelt. Beispielsweise wird „wohnen/leben in“ zu `dbo:residence` [Ab]. Die fertigen RDF-Tripel werden im Terse RDF Triple Language (Turtle) Format in einem Triple Store gespeichert. Der resultierende Wissensgraph kann dann mit SPARQL-Queries abgefragt werden.

4 Implementierung

Nach dieser allgemeinen Beschreibung der Grundarchitektur folgt nun eine eingehende Erläuterung der Implementierung in Python und der dabei eingesetzten Werkzeuge.

Unsere **Textdatenbasis** bilden lokaljournalistische Texte, die uns die Mediengruppe Oberfranken in Form eines XML-Dumps von 160.000 Artikeln aus dem Zeitraum ab 2005 zur Verfügung stellte. Zum Auslesen und Bereinigen des Dumps verwenden wir einen DOM-Parser. Wir extrahieren **Metadaten** wie Titel, Autor*in und Erscheinungsdatum zu den einzelnen Artikeln und speichern diese, wie alle anderen Zwischenergebnisse der Pipeline, zunächst in einer SQL-Datenbank.

Für das **POS-Tagging** kommt *spaCy* zum Einsatz, eine Python-basierte Open-Source-Bibliothek für Natürliche Sprachverarbeitung [HM].

Die **Named Entity Recognition** übernimmt ebenfalls *spaCy*. Für jede erkannte Entität wird ein RDF-Tripel geschrieben, das dieser einen Typ (Person, Organisation, Ort) zuweist. Für unseren ersten Testlauf entschieden wir uns, zunächst einen Fokus auf Personen und ihre Beziehungen zu legen. Deshalb werden die Sätze in diesem Schritt danach gefiltert, ob sie eine Named Entity der Kategorie Person enthalten.

Da das **Entity Linking** während der Erstellung der Tripel zu sehr hohen Laufzeiten führen würde, findet das Linking vorläufig erst bei Abruf der jeweiligen Entität in der später noch beschriebenen Weboberfläche statt. Dabei werden gleichlautende Einträge in

Wikidata gesucht und gegebenenfalls einige Eckdaten zur Person, wie Geburtsdatum und Parteizugehörigkeit, über den SPARQL-Endpoint abgefragt.

Bei der **Relation Extraction** bildet der von uns in Python implementierte *Restrictive Apposition Handler (RAH)* zunächst Tripel aus engen Appositionen vor Personennamen. „die Aschaffenerin Ruth Weiss“ ergibt hierbei etwa `pred:from_rah, subj:Ruth Weiss, obj:Aschaffenerin`. Enge Appositionen, d. h. substantivische Beiwörter vor anderen Substantiven, eignen sich besonders für die Wissensextraktion, da sie vor allem in der journalistischen Sprache häufig auftreten und in den meisten Fällen vor Personennamen einen eng definierten semantischen Gehalt in Bezug auf die Person besitzen (Berufsfeld, geographische Herkunft). Um den RAH zu schreiben, setzten wir uns mit der dependenzgrammatischen Struktur von Sätzen mit engen Appositionen auseinander und leiteten daraus eine Anzahl an Regeln ab, um entsprechende Tripel zu extrahieren. Wir berücksichtigten auch Fälle, in denen eine Apposition sich auf mehrere Personen bezieht („die Informatikerinnen Constanze Kurz und Rena Tangens“) oder mehrere Appositionen für eine Person vorhanden sind („die Informatikerin und Datenschutzaktivistin Constanze Kurz“).

Das OIE-Tool *PropsDE* [Fa16] analysiert daraufhin die syntaktische Struktur jedes Satzes und gibt eine Reihe von Tupeln zurück. In den meisten Fällen bilden diese wie bereits beschrieben die dependenzgrammatische Baumstruktur des Satzes ab:

„2019 konstruierte Hedwig Zuckerl für die deutsche Sprache einen Wissensgraphen.“

```
konstruieren:(subj:Hedwig Zuckerl , dobj:einen Wissensgraphen ,
               prep_für:die deutsche Sprache , mod:2019 )
```

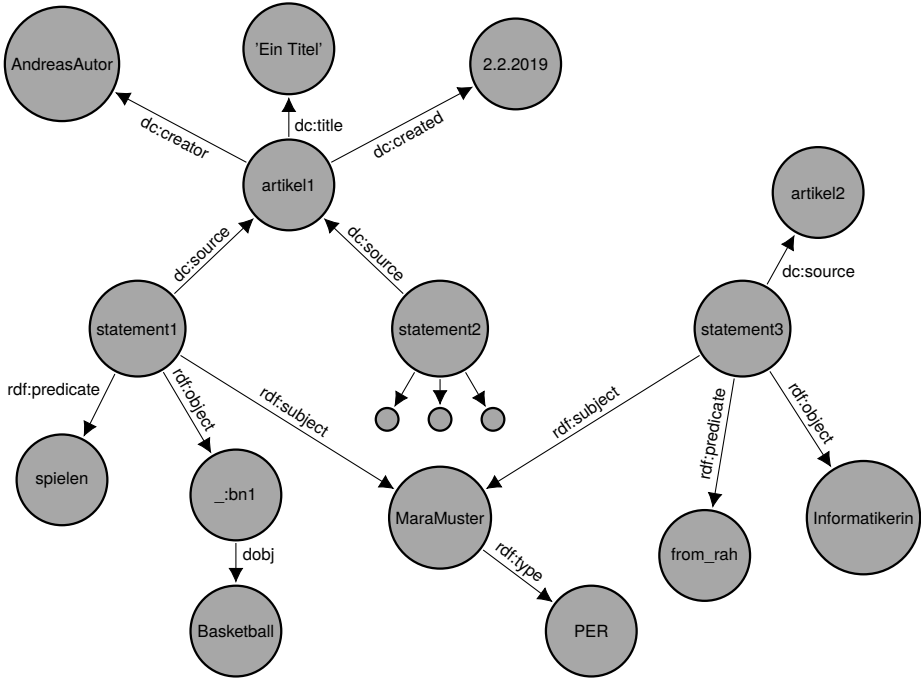
Andere syntaktische Muster werden dagegen als logische Aussagen modelliert. Weite Appositionen wie „Andreas Starke, Bürgermeister von Bamberg“ werden etwa in „SameAs“-Aussagen umgewandelt: `SameAs(subj:Andreas Starke , obj:Bürgermeister von Bamberg)`.

Bei der Umwandlung der Tupel in RDF-Tripel trafen wir eine Reihe von Modellierungsentscheidungen zur Gestaltung unseres Wissensgraphen (vgl. Abb. ??⁵): Viele der Prädikate im Wissensgraphen entsprechen natürlichsprachigen Verben und können deshalb potentiell mehr als ein Objekt besitzen. Dies lässt sich in der klassischen RDF-Tripelform jedoch schwer abbilden. Deshalb wird als Objekt bei solchen Tripeln ein Blank Node eingefügt. An diesen Knoten können dann alle zum Prädikat gehörenden Objekte, aber auch Modifikatoren und adverbiale Bestimmungen gehängt werden.

Auch die zuvor erwähnten Metadaten zur Textquelle ordnen wir den Tripeln jeweils mithilfe von Reifikation zu, indem wir an jedes Tripel eine Statement-ID vergeben und diese dann als Subjekt eines weiteren Tripels mit der ID des Herkunftsartikels verknüpfen.

⁵ dc: Dublin Core Prädikate für bibliographische Angaben [Du12]

Derzeit wird das Mapping von Prädikaten aus den Texten auf Relationen unserer Ontologie über einfache Zuordnungslisten vorgenommen, die beinhalten, welche synonymen Prädikate zu einer Relation passen. Beispielsweise werden über eine solche Liste Verben des Sprechens wie „sagen“ oder „erzählen“ der Relation sagen zugeordnet.



Die fertigen RDF-Tripel werden in einen Apache Fuseki Triple Store überführt, so dass der Zugriff über SPARQL möglich wird. Zusätzlich gibt es ein einfaches Query-Frontend als Django-App, mit dem Endanwender*innen auch ohne die komplexe SPARQL-Syntax Anfragen stellen können und das neben einer Freitextsuche einige Filterfunktionen bietet.

5 Ergebnisdiskussion

Der von uns erzeugte Graph hat eine andere Struktur als die der meisten anderen Implementierungen, da wir nicht versuchen, die semantische Struktur eines oder mehrerer Sätze bis ins Detail abzubilden. Im hier beschriebenen Graphen stehen die Entitäten im Mittelpunkt und der Graph wächst dadurch, dass bei der Verarbeitung der Textbasis ein immer dichteres Netz aus Kanten zwischen diesen Entitäten entsteht. Wir halten diesen Ansatz für zielführender, da dadurch eine konsistente Struktur gewährleistet wird. So vereinfachen wir die Abfrage für die Anwender*innen.

Gleichzeitig müssen wir feststellen, dass OIE-Tupel ihrer Art nach nicht optimal beschaffen sind, um konsistente Einträge im Graphen zu erreichen. Häufig werden ganze Nominalphrasen extrahiert (z. B. `schwimmen(subj: der Lokalpolitiker Tizian Rieth, der für den Bamberger Stadtrat kandidiert, prep_in:Fluss, mod:täglich)`). So stehen Knoten nicht mehr wie gewünscht für eine Entität, sondern bilden eine semantische Einheit ab, die nicht sinnvoll abgefragt werden kann. Um dieses Problem zu lösen ist es nötig, über Alternativen zu PropsDE nachzudenken und möglicherweise sogar ein eigenes OIE-Werkzeug zu implementieren, um einen geeigneten reduzierten Output zu erzielen.

Was das Laufzeitverhalten unserer Pipeline betrifft, können wir aufgrund der Aneinanderreihung von teils bereits existierenden Tools keine detaillierte Gesamteinschätzung etwa in Form einer O-Notation geben. Im Einzelnen fiel bei spaCy ein verhältnismäßig geringer Zeitaufwand an, sobald das zugrundeliegende Sprachmodell für das Deutsche nur einmal geladen wurde. Der selbstgeschriebene RAH schlug ebenfalls nur mit geringem Aufwand zu Buche, da sich die Extraktion auf einige wenige zentrale Regeln konzentriert. PropsDE dagegen verursacht höhere Laufzeitkosten durch eine größere Anzahl an Regeln, allerdings lässt die Verarbeitung, etwa durch satzweises Einlesen, gut parallelisieren. Beim Entity Linking fällt durch den Rückgriff auf die Web-API von Wikidata ebenfalls ein relativ hoher Aufwand bei jeder Anfrage an.

6 Ausblick

Um in Zukunft die Zahl der extrahierten Tripel zu erhöhen und so einen dichteren Graphen zu generieren, ist eine Implementierung der oben beschriebenen Coreference Resolution unerlässlich, etwa durch Einbindung des Tools CorZu [Tu16]. Auch eine elaboriertere Disambiguierung der Entitäten stellt eine weitere Herausforderung für die Zukunft dar.

Im Bereich des Predicate Mapping erfordert unsere derzeitige Vorgehensweise eine händische Identifikation von Zuordnungen zwischen Prädikaten und Relationen der Ontologie bereits zum Zeitpunkt, zu dem die Ontologie festgelegt wird. In Zukunft wäre ein flexiblerer Ansatz wünschenswert, das auch zum anfänglichen Modellierungszeitpunkt noch nicht mitbedachte Prädikate einer Relation in der Ontologie zuordnen kann. Dafür könnte eine lexikalische Datenbank für das Deutsche wie Open German WordNet [Op19] eingesetzt werden, die Aussagen zum Grad semantischer Nähe zwischen zwei Begriffen erlaubt. Mit einer solchen Datenbasis könnte man bei der Eingabe eines Prädikats herausfinden, ob dieses einer bestehenden Ontologierelation ähnlich genug ist, um auf sie gemappt zu werden.

Um einen dichteren und aussagekräftigeren Wissensgraphen zu erzielen, ist es außerdem notwendig, dass wir unsere Ontologie auch auf Entitäten wie Organisationen und Orte ausdehnen, nachdem wir zunächst auf die Repräsentation von Informationen zu Personen konzentrierten. Nicht zuletzt wäre es sinnvoll, Inferenzmechanismen zu entwickeln, um aus den bereits gefundenen Tripeln durch logisches Schließen automatisiert neue Tripel zu generieren.

Literaturverzeichnis

- [Ab] About: residence. <http://dbpedia.org/ontology/residence>. Letzter Zugriff: 01.05.2019.
- [Ba07] Banko, Michele; Cafarella, Michael J; Soderland, Stephen; Broadhead, Matthew; Etzioni, Oren: Open Information Extraction from the Web. In (Veloso, Manuela M., Hrsg.): Proceedings of the 20th International Joint Conference on Artificial Intelligence. S. 2670–2676, 2007.
- [Bo08] Bos, Johan: Wide-coverage Semantic Analysis with Boxer. In: Proceedings of the 2008 Conference on Semantics in Text Processing. STEP '08, Association for Computational Linguistics, Stroudsburg, PA, USA, S. 277–286, 2008.
- [Bo14] Boschee, Elizabeth; Freedman, Marjorie; Khanwalkar, Saurabh; Kumar, Anoop; Srivastava, Amit; Weischedel, Ralph: Researching persons & organizations: AWAKE: From Text to an Entity-centric Knowledge Base. In: 2014 IEEE International Conference on Big Data (Big Data). IEEE, S. 1030–1039, 2014.
- [CPW04] Clark, Peter; Porter, Bruce; Works, Boeing Phantom: Km – The Knowledge Machine 2.0: Users Manual. Department of Computer Science, University of Texas at Austin, 2(5), 2004.
- [Du12] Dublin Core Metadata Initiative (DCMI) Metadata Terms. <http://dublincore.org/specifications/dublin-core/dcmi-terms/>, 2012. Letzter Zugriff: 01.05.2019.
- [Et11] Etzioni, Oren; Fader, Anthony; Christensen, Janara; Soderland, Stephen et al.: Open Information Extraction: The Second Generation. In: Twenty-Second International Joint Conference on Artificial Intelligence. 2011.
- [EW16] Ehrlinger, Lisa; Wöß, Wolfram: Towards a Definition of Knowledge Graphs. SEMANTiCS (Posters, Demos, SuCCESS), 48, 2016.
- [Fa16] Falke, Tobias; Stanovsky, Gabriel; Gurevych, Iryna; Dagan, Ido: Porting an Open Information Extraction System from English to German. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. The Association for Computational Linguistics, S. 892–898, 2016.
- [Ga17] Gangemi, Aldo; Presutti, Valentina; Recupero, Diego Reforgiato; Nuzzolese, Andrea Giovanni; Draicchio, Francesco; Mongiovì, Misael: Semantic Web Machine Reading with FRED. Semantic Web, 8:873–893, 2017.
- [HM] Honnibal, Matthew; Montani, Ines: , spaCy. <https://spacy.io/>. Letzter Zugriff: 31.01.2019.
- [KL17] Ktob, Ahmed; Li, Zhoujun: The Arabic Knowledge Graph: Opportunities and Challenges. In: 2017 IEEE 11th International Conference on Semantic Computing (ICSC). S. 48–52, 2017.
- [Op19] Open German WordNet. <https://github.com/hdaSprachtechnologie/odenet>, 2019. Letzter Zugriff: 23.06.2019.
- [Ra11] Ramshaw, Lance; Boschee, Elizabeth; Friedman, Marjorie; MacBride, Jessica; Weischedel, Ralph; Zamanian, Alex: SERIF Language Processing — Effective Trainable Language Understanding. In (Olive, Joseph; Christianson, Caitlin; McCary, John, Hrsg.): Handbook of Natural Language Processing and Machine Translation, S. 626–631. Springer, 2011.

- [Sh15] Sharma, Arpit; Vo, Nguyen Ha; Aditya, Somak; Baral, Chitta: Towards Addressing the Winograd Schema Challenge - Building and Using a Semantic Parser and a Knowledge Hunting Module. In (Yang, Qiang; Wooldridge, Michael, Hrsg.): Proceedings of the 24th International Joint Conference of Artificial Intelligence. S. 1319–1325, 2015.
- [Tu16] Tuggener, Don: Incremental Coreference Resolution for German. Dissertation, 2016.
- [Wa15] Wang, C.; Gao, M.; He, X.; Zhang, R.: Challenges in Chinese Knowledge Graph Construction. In: 2015 31st IEEE International Conference on Data Engineering Workshops (ICDEW). S. 59–61, 2015.

Improvement of automated social media sentiment analysis methods - a context-based approach

Dennis Debeye,¹ Tim Eder,² Paul Vincent Guigas,³ Viktoria Schuberth⁴

Abstract: The sentiment analysis of social media data increasingly gains importance in business and research. But still, topical algorithms cope with problems, since it is reasonably manageable to extract the tonality of a social media post, but not the authors attitude towards a given topic. However, in most cases, this is the relevant information users of social media analysis tools are looking for. To tackle this problem, we propose a context-based algorithm that not only focuses on isolated postings, but also takes the authors' earlier postings and their interactions with other users' posts into account to derive their actual opinion on a subject. To evaluate this approach, we implemented a test system and compared the algorithm's results to manually assessed sentiments.

Keywords: sentiment analysis; social media analysis; opinion mining

1 Introduction

At the present time, we find ourselves midway of a revolution of the way people communicate with each other: Instant messaging and social media increasingly gain importance, not only for private communication, but also for businesses and organizations. On social media, everybody is a potential content creator; every user has a channel to broadcast their opinion and thoughts on certain topics. This challenges marketing departments. They are forced to step away from traditional monodirectional campaigns and towards running social or even cross media campaigns, that involve continuous exchange with their digital communities. The communication has become bilateral.

Not only do users of social media talk *with*, but also *about* the organizations, their products, offered services or activities. And this makes social media a great opportunity for such organizations, since an analysis of these publicly available communication data from social media networks can help to improve marketing strategies and achieve better follower/customer engagement, better follower/customer service, better reputation management and brand awareness, product innovation, business process improvement and even the discovery

¹ HTWK Leipzig , IMN, Karl-Liebknecht-Str. 132, 04277 Leipzig, Germany dennis.debeye@stud.htwk-leipzig.de

² HTWK Leipzig , IMN, Karl-Liebknecht-Str. 132, 04277 Leipzig, Germany tim.eder@stud.htwk-leipzig.de

³ HTWK Leipzig , IMN, Karl-Liebknecht-Str. 132, 04277 Leipzig, Germany paul_vincent.guigas@stud.htwk-leipzig.de

⁴ HTWK Leipzig , IMN, Karl-Liebknecht-Str. 132, 04277 Leipzig, Germany viktoriaschuberth@stud.htwk-leipzig.de

of new business opportunities (cf. [HPH14]). Already today, these potentials are utilized in a wide range of application fields, e.g. in political election campaigns [AG17], the banking sector [Co15] or even on the stock market [HPH14]

There are many thinkable metrics and indicators that provide valuable information for organizations to improve the understanding for their communities, e.g. term frequency or demographic information. One of the key indicants surely is the *sentiment*, i.e. the information about how people feel about a certain topic. To find a communities sentiment, social media analysis (SMA) tools run *automated sentiment analyses* on a vast amount of original posts and comments.

This paper focuses exactly on these automated, computational sentiment analysis (SEA) procedures. Automated SEA usually utilizes methods of the fields *information retrieval*, *text mining*, *web mining* and *natural language processing* (NLP). The common problems most of these approaches have, is that they focus on the bare text of a single post to derive a sentiment from it. No matter how good these approaches are, they will always only be able to reflect the sentiment of an isolated post. However, the information most data scientists and sociologists are looking for will most likely not be the sentiment of a *post* but the sentiment of a *user*. Therefore, in this paper, we will present an approach to a technique to enhance the accuracy of an existing automated SEA tools by taking into account the context (i.e. the users other interactions with the social community) of the analyzed social media postings. As application example, we chose the micro blogging platform *Twitter*. In the remainder of this paper, we will first examine the current state of the art regarding SEA in general, then depict the motivation for a new, context based approach and afterwards present our solution and evaluate and discuss our findings.

2 State of the Art

Long before the World Wide Web was invented, we have let our decisions be influenced by others point of view. *Can somebody recommend this washing machine? Can someone recommend a local restaurant?* The internet gives us a place to share opinions with others and spread it around the world. Depending on the platform and the count of potential readers, these viewpoints could reach thousands and millions of people. But not only recommendations about products and services can be shared, also political statements and information. Especially Twitter, the most popular microblogging platform, has a widespread community. Each day 500 million tweets are sent. 326 million people use twitter monthly [Ho19]. Sentiment Analysis is one of the first steps to extract the opinion out of an unstructured text with relatively low investment of time and effort [Pa18].

Over the time sentiment analysis became a very important topic for researchers and firms for evaluating opinions over a large user population. The number of papers related to this topic increases rapidly: according to [Mä18], almost 7000 papers have been published until 2016. We have also seen a massive increase in the number of papers focusing on sentiment analysis and opinion mining during the recent years. The most commonly used target classes

for the sentiment classification are *positive* and *negative*, sometimes extended by a third *neutral* class.

2.1 Lexical Approach

A tremendous amount of papers rely on the lexical approach. A lexicon of sentiments represents a list of words or lexical attributes and are labeled with their semantic polarization [Li12]. Manually creating such lexicons is a very costly - but robust - technique, so most of the approaches resort to preexisting lexicons [HG14].

Another way to generate opinion lexicons is the corpus-based approach. It's focused on conjunctions, like AND, OR, EITHER-OR and NEITHER-NOR. The conjoined word after AND has usually the same orientation like it's predecessor. People usually express the same opinion on both sides of the conjunction [Li12]. After BUT, we can expect a contrary opinion. In practice this approach is not always consistent. [Ho13] introduces a way to improve the lexical-based sentiment classification. They analyzed how emoticons provide sentiments and based on that, they made a manually created emoticon sentiment lexicon. A relevant approach for opinion mining and especially for this paper is described in [HG14]: they created a rule-based model for sentiment analysis. In a micro-blogging context they first construct and empirically validate a *goldstandard* list of lexical features combined with their sentiment value. Adding five general rules that contain grammatical and syntactical conventions to the lexical features, the sentiment intensity becomes even clearer.

Our approach relinquishes the costly work of creating a lexicon beforehand and we do not have to deal with grammatical or syntactical rules. Lexical approaches are not always consistent. In case of inconsistency, we use deepSEA to take a closer look on ambiguous tweets. We focus on the author of the tweet and his opinion towards the tweet's topic and not only on the occurring words and phrases.

2.2 Machine Learning

Manually creating lexicons is a time-consuming process. Since machine learning is becoming more and more relevant in the field of natural language processing, it's also a state-of-the-art approach in sentiment analysis. The Naive Bayes (NB) is a classifier based on the Bayes rule and the naive assumption that features' probabilities are independent of each other. This works well on text categorization.

Maximum Entropy models should prefer the most uniform models that satisfy a given constraint. These classifiers are used in the approach described in [MKP02]. In a sentiment analysis context this is useful for recognition of patterns in word usage between different classes to put it into these categories.

Support Vector Machines are non-probability classifiers. SVMs work by separating data points in space using hyperplanes [GBH09]. [PP10] presented a way by collecting a corpus of text with emoticons. These collected corpora were used to train a classifier to identify

positive and negative sentiments. Each of the described machine learning approaches require large sets of training data and always depend on them. It's also expensive in terms of memory consumption, CPU processing and classification time. With machine learning the sentiment assumptions are made by correlating large sets of data. This does not imply causation [Le19]. To prevent this possibility of errors, deepSEA takes a closer look at the user and her or his specific opinion to make the sentiment less ambiguous. It also does not have the need of a large training set or special hardware requirements.

2.3 Semantic Approach

There is also a way to extract the sentiment out of an unstructured text with semantic web technologies. One approach is introduced in the book [SHA12]. The semantic values extracted from tweets can be used to measure the overall correlation of a group of entities (e.g. all APPLE products) with a given sentiment polarity. For example products of brands can be used as vocabulary, which are listed in the training data with occurrences in positive and negative tweets. The entities *iPad*, *iPod* and *Mac Book Pro* appeared preferably in tweets with positive polarity and they are all mapped to the semantic concept PRODUCT/APPLE. As a result, the tweet from the test set *Finally, I got my iPhone. What a product!* is more likely to have a positive polarity because it contains the entity *iPhone* which is also mapped to the concept PRODUCT/APPLE. It is also possible to give a statement about products, which are not mentioned in the training set (e.g. they are not released yet), because they are mapped to their brands. ArsEmotica - an application software for associating the predominant emotions to artistic resources of a social tagging platform - also uses the semantic approach. This is done by exploiting and combining available computational and sentiment lexicons with an ontology of emotional categories. Words with a relevant sentiment meaning get ranked by users feedback. So the emotional ontology hierarchy grows and is available within the linked data network [Ar12]. This approach calculates a sentiment relative to related topics, while the deepSEA approach tries to estimate a tweets sentiment using tweets relative to the same topic. This way we expect to get an even more accurate result

2.4 Context-based Approach

In a user-based context, we can extract additional information of the opinions about a specific topic. For example in conversations of users about this subject, the users attitude or likes and retweets. The approach described in [VCB14] goes hand in hand with the machine learning technologies: they are using kernel functions - integrated in SVMs - to capture specific aspects of sentiment connection between two tweets. In conversations on Twitter they disambiguate even very short messages and characterize them according to their authors. Their experimental evaluation proves that sequential tagging effectively embodies evidence about the contexts. They also improve detection accuracy to around 20%. In case of ambiguous phrases in tweets, deepSEA looks at the author and her or his

previous opinions on this topic. Strictly speaking, deepSEA does not analyze the sentiment of the tweet, it rather analyzes the sentiment of the user.

3 Problem motivation

Most previous approaches to sentiment analysis do not take a tweet's context into consideration, they merely look at the choice of words in a single tweet. While this is fine for tweets in a more literal style, it is more complicated to analyse tweets written in figurative language. Especially the detection of irony and sarcasm is a field, which faces a certain set of challenges. Key factors in detecting irony and sarcasm are aspects on the syntactic and textual level, the semantics and pragmatics as well as the discourse analysis [HR17]. While progress was made in the fields of the syntactical, lexical as well as the semantical level, the aspect of discourse analysis still is a widely unknown territory. Key factor therefore is the analysis of the discussion itself, i.e. how utterances relate to each other within the Twitter ecosystem. It is therefore necessary to investigate the broader context of a tweet by taking a closer look at the authors and their intentions. The main motivation of this paper is, to achieve this analysis by taking Twitter's own metrics (e.g. number of likes and retweets on a tweet) into account.

4 Solution

4.1 deepSEA Approach

Sentiments get determined by our underlying analysis tool *VADER* (cf.[HG14]) with a confidence. If this confidence is below a set threshold, the tweet's sentiment gets marked as uncertain. To specify an uncertain sentiment, the users opinion on the subject is put in broader context. Based on the hypothesis, that there is a confirmation bias in a Twitter users hashtag usage [KL18], the users interaction within the Twitter social ecosystem is being analyzed. In this context, the initial tweet is as important as the set of the original authors interactions on the same subject. Therefore, the deepSEA approach is based on the following assumptions:

- If a user posts multiple tweets about a certain topic(represented by the usage of a subject hashtag), these tweets reflect the same sentiment towards that topic.[KL18]
- A like (vgl. [Tw]) shows the users appreciation for a tweet . Knowing, that the users hashtag usage is biased, the assumption can be made, that a like on a tweet using a hashtag the user has tweeted about before, reflects an approval for the opinion of certian tweet.

The tweets the author interacts with are used to calculate a second sentiment which considers the authors feeling about the subject itself. In case of an uncertainty in the initial tweet's

sentiment rating, these derived tweets are used to adjust the sentiment in the authors initial tweet.

4.2 deepSEA Logic

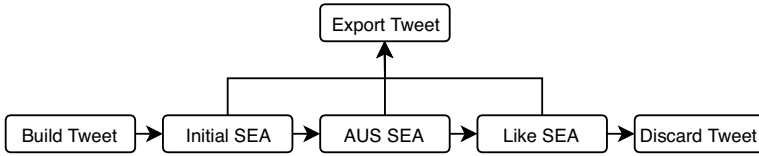


Fig. 1: The deepSEA rating process is controlled via the sentiment pipeline and is subdivided into rating steps.

The deepSEA logic is subdivided into four partial steps as seen in Fig. 1. The Twitter metadata is built into a readable data format and transported through a pipeline of different SEA steps. If the current step is successful, the tweet will get exported and the sentiment of the most recent SEA step will be regarded as the true or final sentiment of the tweet. The success of an analysis step depends on each step’s individual definition. In the *Initial Sentiment Analysis* (Initial SEA) the initial tweet’s sentiment gets determined by the sentiment analysis tool *VADER* [HG14]. If the confidence of the initial tweet’s lexical analysis is above a fixed threshold, the analysis step succeeded. Otherwise, the tweet is presented to the *Advanced User Subject Sentiment Analysis* (AUS SEA). The AUS SEA rates the users subject based on derived tweets. By analyzing the sentiment of the users previous tweets on that subject with *VADER*, a ratio of the most frequently occurring sentiment in relation to all derived tweets is drawn. The most frequently occurring sentiment becomes the new sentiment of the tweet, whereas the ratio represents its confidence. If this confidence is below a set threshold, the tweet is passed to the *Like Sentiment Analysis* (Like SEA). The Like SEA step uses all tweets related to the analyzed subject that the user has liked as a basis to draw a ratio analogous to the AUS SEA. If the confidence of this ratio is also under a set threshold, the tweet is marked as not ratable and therefore discarded. After the automated sentiment-analysis processed is finished, tweets are rated in *Twinder* (a subsystem designed to manually rate tweets through a user interface) for further analysis and evaluation.

4.3 Test Architecture

For gathering the necessary data, a system with the architecture presented in Fig.2 was implemented. A logging microservice was realized to keep track of every step of the processed data. Using the stream processing framework Apache Kafka, a Twitter stream for a certain topic is subscribed and incoming data conveyed to the *deepSEA Logic* microservice, where its sentiment gets determined through the process described in section 4.1. The overall process is triggered by a REST call against the *deepSEA Ingest* service. Also, with

this call, the subject of the analysis is set. When the rating process of a tweet is finished, Kafka transports the rated tweet to the so called *Twinder API*, which saves it to a Database. They are then manually rated by (human) test subjects, through a simple frontend in order to be able to evaluate the performance of our approach. After this processing step is completed, the tweets are evaluated in the *statistics service*, that assesses the sentiment rating quality of the different approaches.

4.4 Data Acquisition

The test data with the subject *Captain Marvel* was acquired on the 12th of March 2019. Over the timespan from 8:13pm to 8:55pm CET, 429 tweets with a textual mention of *Captain Marvel* were captured and analyzed with *deepSEA*. The corresponding thresholds of the SEA step were set as follows:

- init SEA: 0.6
- AUS SEA: 0.5
- Like SEA: 0.5

The procedure of finding the init SEA threshold consisted of a series of experiments roughly resembling the gradient descent method [Sc17]. Through iteratively adjusting the threshold, we found, that a value of 0.6 provided the best results.

The confidence of the *AUS* resp. *Like* sentiment is calculated by deviding the count of tweets that show the dominant sentiment by the count of all relevant tweets that were taken into account. Since the overall probability of occurrence of relevant tweets is relatively small, we decided that an absolute majority (i.e. a ratio $>50\%$) is the most sensible threshold to choose, since it *always* provides a usable result, if a dominant sentiment exists.

4.5 Manual Rating Process

Additionally, each tweet was rated manually by four test subjects. The subjects were asked to identify the attitude of the original author towards the topic (in this case the movie *Captain Marvel*) rather than the general tonality of the tweet. The subjects were presented each tweet exactly once and were supposed to rate it as positive, negative or neutral. The results of this manual rating served as reference sentiment for the automatic SEA methods.

5 Discussion

5.1 Data Collection and Evaluation

We chose the topic of *Captain Marvel*'s cinema release, because we expected controversial sentiments, which - as one can derive from the results presented in Table 1 - turned out to be mostly true. The true sentiment of the tweets was found via our rating frontend *Twinder*. We specifically did not want test subjects to rate the tonality of the language, like a sentiment analysis software would, but rather the author's true stance on the topic. This meant that they had to consider irony and disagreements with people of polar opinions. If three test subjects agreed on a sentiment, we took it as ground truth and compared our analysis to this value. Non-english tweets and spam was discarded. After this data cleansing, we ended up with a total of 429 tweets. Their ratings are summarized in table 1.

	True Sentiment	deepSEA	VADER
positive	303	388	367
neutral	92	6	18
negative	34	35	44

Tab. 1: Overview of sentiment values

Our algorithm rated 191 of these 429 tweets, since they did not match our specified certainty threshold for the initial sentiment analysis. Of these 191 tweets, 101 had a correctly detected final sentiment. This means deepsea could not manage to outperform VADER, which detected the sentiment of 106 tweets correctly (but - as explained - all with insufficient certainty). However, the vast majority of tweets was still classified correctly only by analysing the original posts context. This is an indicant, that our foundational assumptions were correct.

Nevertheless, the data shows that deepSEA failed to improve VADER ratings in some cases. It is noticable, that many tweets were falsely rated to be positive and hardly any neutral tweets were detected.

5.2 Further Thoughts

The deepSEA approach appears to be well suited in the case of a person with a positive opinion on the topic expressed in negative language, like in a discussion between two users. VADER can not detect these sentiments, whereas our context based approach takes a user's previous comments and interactions on the subject into account to find their true opinion. The opposite scenario however, causes a lot of false classifications: It seems appears more unlikely, that a user would post a negative opinion using positive words. This could explain why our system classifies too many tweets as positive. Some edge cases come to mind that could further explain the false ratings:

- A user has expressed a negative opinion even though their past statements were positive.
- They did not use a *like* as an agreement.
- They generally use negative language in all of their tweets.
- Etc.

Since we rely on VADER in our submethods, the original problem of not being able to detect context or irony persists even in the lower stages of our analysis. VADER also failed to classify a lot of neutral tweets, since the numeric sentiment range for a neutral rating is quite small. This means the software prefers to give positive or negative ratings, even though the manual classification determined a higher amount of neutral tweets for the analyzed topic.

5.3 Possible Improvements

Our sample data set was quite small and four participants are most likely not enough to construct a valid true sentiment. Hence our first goal in further research is to optimize our test data not only in regards to size, but also tweet content. In order to correctly identify the shortcomings of our algorithm, we need to test with an equal amount of handpicked positive, neutral and negative tweets. Our current data set was quite imbalanced, the most common sentiment was *neutral*. Increasing the threshold for neutral ratings could also improve results.

When we incorporated the context into our rating, we only looked at user context, i.e. other tweets from the same author. Another possibility for improvement could be to construct a rating from conversational context. By looking at replies in a comment thread, it might be easier to figure out a user's agreement or disagreement. We also did not take temporal changes into account: a user's opinion on a subject could change over time. This causes uncertain ratings in our algorithm.

Our sentiment analysis did not run, when VADER's initial rating had a high certainty. In that case we did not consider the possibility of the VADER sentiment being wrong. In future tests we will start our analysis on every tweet and compare it with the initial sentiment.

6 Conclusion

In this paper we have pointed out weaknesses of modern sentiment analysis algorithms. In order to overcome them, we came up a new analysis strategy that takes the context of a social media posting into account: Whenever the underlying lexicon approach failed to return a sufficiently certain sentiment, the algorithm considered older posts and liked posts by other users to derive the sentiment of the original one. We implemented a system to test

our approach with real time test data and statistically evaluated the results.

We found out that, despite being seemingly successful in enhancing the detection of tweets with a negative sentiment, our algorithm yet has issues in improving it for other sentiments as well. We discussed the most probable reasons for those findings and possible solutions, that might lead to better results in future tests.

References

- [AG17] Allcott, H.; Gentzkow, M.: Social Media and Fake News in the 2016 Election. In: *Journal of Economic Perspectives*. 2017.
- [Ar12] Arsmeteo, A. C.: Sentiment analysis in the planet art, <http://www.di.unito.it/~patti/arsemotica.htm>, 2012, visited on: 03/15/2019.
- [Co15] Costin, D.: Banking Business and Social Media – A Strategic Partnership. In: *Theoretical and Applied Economics*. 2015.
- [GBH09] Go, A.; Bhayani, R.; Huang, L.: Twitter sentiment classification using distant supervision. *Processing 150/*, Jan. 2009.
- [HG14] Hutto, C. J.; Gilbert, E.: Vader: A parsimonious rule-based model for sentiment analysis of social media text. *Proceedings of the Eighth International AAAI Conference on Weblogs and Social Media/May*, pp. 216–225, 2014, URL: <http://comp.social.gatech.edu/papers/icwsm14.vader.hutto.pdf>.
- [Ho13] Hogenboom, A.; Bal, D.; Frasincar, F.; Bal, M.; de Jong, F.; Kaymak, U.: Exploiting emoticons in sentiment analysis. In: *Proceedings of the ACM Symposium on Applied Computing*. Pp. 703–710, Mar. 2013.
- [Ho19] Hootsuite: 28 Twitter Statistics All Marketers Need to Know in 2019, <https://blog.hootsuite.com/twitter-statistics/>, 2019, visited on: 03/15/2019.
- [HPH14] Holsapple, C.; Pakath, R.; Hsiao, S.-H.: Business Social Media Analytics: Definition, Benefits, and Challenges. In: *Twentieth Americas Conference on Information Systems*, Savannah. 2014.
- [HR17] Hernandez Farias, D.; Rosso, P.: Irony, Sarcasm, and Sentiment Analysis. 2017.
- [KL18] Kowald, D.; Lex, E.: Studying Confirmation Bias in Hashtag Usage on Twitter./, pp. 2–4, 2018, URL: <http://arxiv.org/abs/1809.03203>.
- [Le19] Leetaru, K.: A Reminder That Machine Learning Is About Correlations Not Causation, 2019, URL: <https://www.forbes.com/sites/kalevleetaru/2019/01/15/a-reminder-that-machine-learning-is-about-correlations-not-causation>, visited on: 03/15/2019.
- [Li12] Liu, B.: *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers, 2012.

- [Mä18] Mäntylä Graziotin, K.: The evolution of sentiment analysis. In: A review of research topics, venues, and top cited papers. 2018.
- [MKP02] Mehra, N.; Khandelwal, S.; Patel, P.: Sentiment Identification Using Maximum Entropy Analysis of Movie Reviews./, Jan. 2002.
- [Pa18] ParallelDots, I.: Breakthrough Research Papers and Models for Sentiment Analysis, <https://blog.paralleldots.com/data-science/breakthrough-research-papers-and-models-for-sentiment-analysis/>, 2018, visited on: 03/15/2019.
- [PP10] Pak, A.; Paroubek, P.: Twitter as a Corpus for Sentiment Analysis and Opinion Mining. In: Proceedings of the Seventh conference on International Language Resources and Evaluation. European Languages Resources Association (ELRA), Valletta, Malta, May 2010.
- [Sc17] Schöni, M.: Was ist das Gradientenabstiegsverfahren?, <https://zkma.blog.com/2017/02/11/was-ist-das-gradientenabstiegsverfahren/>, July 2017, visited on: 07/04/2019.
- [SHA12] Saif, H.; He, Y.; Alani, H.: Semantic Sentiment Analysis of Twitter. In: The Semantic Web–ISWC 2012. Vol. 7649, pp. 508–524, Nov. 2012.
- [Tw] Twitter: Glossary, URL: <https://help.twitter.com/en/glossary>, visited on: 07/05/2019.
- [VCB14] Vanzo, A.; Croce, D.; Basili, R.: A context-based model for Sentiment Analysis in Twitter./, Aug. 2014.

Informatik in der Lehre

Theoretical evaluation of the potential advantages of cloud IDEs for research and didactics

on the example of Eclipse Che

Leonhard Applis ¹

Abstract: Eclipse Che is a open source cloud-based IDE, inheriting the approach of building cloud native software within a remote environment. Instead of developing on a local machine, a remote development-server is accessed via web-browser and the artifacts are compiled into docker-containers.

This paper shortly summarizes the required environment, introduces the concept and workflow of Eclipse Che and evaluates the benefits and downsides of this approach. Focus is set on a new point of view, enhancing computer studies and programming classes with this technology.

Keywords: Eclipse Che; Cloud Native; DevOps; Docker; Kubernetes; Higher Education; Software Development

1 Introduction

Children look into the clouds and count sheep. Developers look into the clouds and check their applications. Cloud technologies are already state of the art, enabling global players to run applications on a world-scale. The market-leaders stocked up their portfolios with a variety of different cloud technologies, ranging from cloud computing, cloud storage, database-as-a-service or rented authentication. Modern software needs to be scalable, self-contained, and should run on any device, as well as in the cloud.

As cloud-technology such technologies are referred which achieve five attributes: On-demand self-service, broad network-access, resource pooling, rapid elasticity and measured service [Me11]. Cloud technology usually splits into two components: infrastructure and services, where Docker and Kubernetes are considered infrastructure and provide a platform to host services. Most of the attributes are (pseudo-) achievable by the infrastructure alone – such as restarting containers. Software running as a service can be put into two categories: *Cloud ready* – meaning that it can be run in common cloud environments, such as Kubernetes. The term cloud ready does usually not include live-scalability. Cloud-readiness is mostly achieved by removing dependencies and either running on bare-linux or being ported into a docker-container. Lately the term *Cloud native* came up [Bi17], referring to software which

¹ Technische Hochschule Georg Simon Ohm, Leonhard.Applis@Protonmail.com

is built from the very first steps to run inside the cloud. This kind of software is usually running inside a docker-container and both resilient and elastic by design [To17]. Resilience refers to the ability of handling failure, both from external sources and restarting/configuring themselves. Elasticity is the ability to request more resources on demand, performing more computations when required, but also to free the resources if not needed. Cloud-ready software usually achieves only a certain degree of resilience. Cloud-Native applications are aware of their context and collect meaningful metrics for the platform that they cannot reach other services or need more capacities. The applications must also be able to use the gained resources in a productive and healthy way. Modern cloud-platforms can perform pseudo-scalability for cloud ready-services by starting multiple instances of the same service. The cloud-platform performs load-balancing between these instances. This naive approach can work out fine, but for example starting two services working on the same database will not yield any real scalability. Another common problem arises when two services are started on the same machine and need the same port, just to name two problems with cloud ready software in a cloud environment.

As Cloud is a central element of modern IT-development, it is strange to see that it is not yet widely taught across universities in Germany ². There are courses to teach either classic virtualization or docker-basics, but the knowledge of full-stack cloud-platforms and cloud native development is sealed behind corporate doors and conference workshops. Meanwhile, the annual StackOverflow developer-survey [St19] showed, that DevOps-engineers are not only among the most wanted and best paid jobs but also tend to be the happiest participants. Every student either wants to be happy or to be rich (some want both), making Cloud an interesting topic for their career.

There are many reasons why there are not many courses, but the most prominent is the complexity of the topic: To build a real cloud native application, the developer must understand virtualization, system-administration, development, infrastructure and networking. Additionally, the developer needs to use specific tools, know best practices, tests and work in a team. All these factors make a high stake for entrance, also represented in the developer-survey [St19] which states that most DevOps engineers have a decade of experience, usually in operations topics.

Luckily a new open-source software, Eclipse Che, is on the rise to enable everyone for cloud native development. Che is unlike a normal Eclipse-distribution: It is hosted on a server and developers get access with their browser. Instead of installing dependencies, compilers etc. locally, a common workspace is set up, which is shared among the programmers. The software build is run inside a docker-container, making it cloud native by default. With Version 7 upcoming and prominent support from RedHat, Eclipse Che is not a prototype anymore. The latest distributions of RedHats OpenShift are shipped with Che in default. Therefore, it is worth looking at this possible game-changer. With the first part of this paper, the structure and ideas of Che are explained in more detail, the second part of this paper

² <https://www.hochschulkompass.de/studium/studiengangsuche> lists currently only 10 german universities with courses on cloud computing, about 30 courses on virtualization

covers general arguments about this approach and gives a detailed overview of reachable benefits for education.

2 Eclipse Che

2.1 Environment and Requirements

Che is hosted as a server-application and is already cloud native. The only requirements are that either Docker, Kubernetes or OpenShift are available for the installation. The differences between these will shortly be summarized.

Docker³ is a container-platform, where a container is a standardized unit of software including the code, dependencies and core-functionalities. A container runs on an OS-Virtualization, originating from LXC, has declared interfaces and can be parameterized. One of the core features is to connect containers for bigger projects, e.g. one database container and one web-server-container, which are connected into a virtual network. Docker is mostly famous for this docker-engine, but enriches it with monitoring and logging, as well as the ability to move containers onto different machines.

Kubernetes⁴ is a container-orchestration-system based on Docker. Kubernetes picked up the growing problem in managing multiple docker-containers by simplifying resource-management. Additional to deployment and monitoring of container-groups it provides auto-restart mechanisms and scaling based on (custom) metrics. When run on Kubernetes, Che is hosted as a single docker container in the existing platform. The term pod, also later used in this paper, refers to a suite of connected containers. An example pod would be a simple two-tier web-application, where the database and web-server are two different, but connected containers.

OpenShift⁵ provides a software-suite around Kubernetes with the goal to automate cloud native development and delivery. Notable additions include Jenkins, Gitea, Sonarqube and lately Che. Regarding Che there are no notable changes in handling, as Che is simply run on the build in Kubernetes. When run on Kubernetes or Docker Che will require two additional containers for authentication with KeyCloack. For OpenShift the standard-openshift authentication is used.

Eclipse Che can be run on any base-technology on localhost – this is rather for demonstrations as well as for developers working on the Che-Code. When run (locally) on docker, the plugin CheDir enables a portable workspace, which can be used on any docker based Che.

³ <https://www.docker.com/>

⁴ <https://kubernetes.io/>

⁵ <https://www.openshift.com/>

2.2 Technology and Workflow

A normal IDE is software on a computer which makes programming easier than doing it in a plain text-editor and commandline. Common tasks for an IDE are dependency management, debugging, refactoring and auto-completion. It is important that an IDE is not *required* to develop - Vim can write Java files, the JDK can compile and the JRE can run everything manually from console. With that in mind, an IDE is just a tool-stack to make things (a lot) more comfortable – but it also has requirements such as the JDK, the .Jars needed in the classpath etc. as they are only utilized by the IDE. This stack can be considered a *workspace* as it contains everything required to work, with the IDE making the workspace comfortable to use. Eclipse Che does not work like a normal IDE. Che consists of three main components shown in Figure 1⁶.

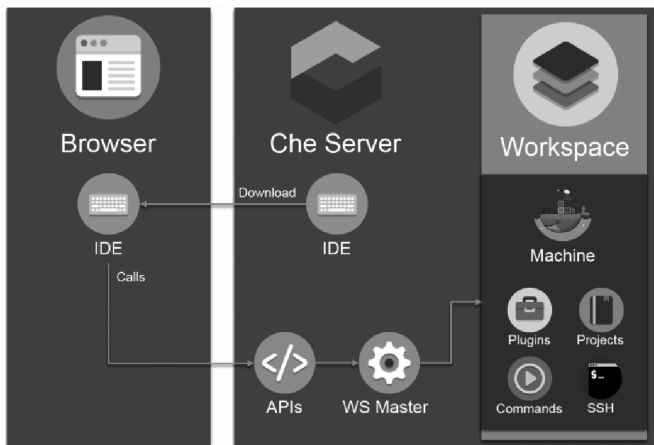


Fig. 1: Che Workflow

1. Workspaces, including Runtimes and IDEs
2. A browser-based IDE
3. An administrative server

A workspace is like the above mentioned common *workspace*. It is a single machine, as a container, containing the project files, compilers, package managers and an IDE-Interface. There are two big differences to a normal development-setup:

- The required items for the workspace are explicitly declared, making the workspace itself a docker image

⁶ source: https://en.wikipedia.org/wiki/Eclipse_Che#/media/File:Eclipse_Che_-_Workflow.PNG

- The IDE is not graphical – it is a REST-API performing actions like a normal IDE, such as build, writing to files and installing packages.

An additional distinction is about building software. Instead of building the .Jar in the remote-workspace, it is build inside a container in the remote-workspace. While this sounds a bit confusing, it makes sense to separate the actual runtime from the workspace. This is also what happens on a local machine – the executable is compiled and run separately. Having these runtimes inside a docker-container comes with great benefits, outlined later in this paper.

The second component is the web-based IDE, which is basically a web-page performing the required REST-calls and utility-tasks for the developer. It is sent by the administrative server when an authorized user accesses the workspace via browser. These two components can be enriched with plugins. Common plugins are language-extensions which support syntax-highlighting etc. or package managers with their regarding lifecycles. The difference to a normal IDE plugin is that it has two components: One in the workspace-API and one in the hosted web-IDE. Che offers a suite of common plugins dependent on the project. When starting a Maven-project, the workspace is initialized with Maven and Java.

The last component is the Che-administration server. In the administration-server workspaces are orchestrated and monitored. The authorization is also done at the admin-server and not by every workspace. The workspaces can be shipped onto multiple separate machines. The runtimes do not necessarily need to be on the same machines as their workspace. Instead of the web-based IDE, a desktop IDE can be connected to the workspace. This is done by mounting the workspaces-filesystem in the desktop IDE. This will grant access to auto-completion and other features, but will miss Che specific helpers the web-IDE offers. This feature looks to be rather a proof of concept.

A single workspace can contain multiple projects, whereof every project will have its own runtime. Projects can either share a versioning control or be separated. You can build whole pods in a single workspace with the single containers being the sub-projects. You can also compose applications inside the workspace, meaning that the resulting container of a project can be the base-image of another project. A workspace can be shared among multiple users. Both will work on the same remote filesystem, altering files simultaneous. Due to changes being transmitted as HTTP-requests, the current implementation when working on the same file is a *last write wins*-policy. For version 7 a multi-cursor file-editing will be provided, like it's common in OneDrive-documents. This is not to be mixed up with having multiple users work on the same project but on their own workspace. The code, and the workspace itself, can commonly be shared with any versioning tool such as Git. It's mostly dependent on the team which kind of cooperation they prefer.

Eclipse Che originates from the Eclipse-Theia project. Theia focussed (only) on the idea of having a remote workspace and access it via browser. Che uses Theia and enriches it with idea of capsulation and containerization. An alternative to Eclipse Che is Amazons

Cloud9⁷, which is free for students. Microsoft is currently working on a similar solution, the platform Coder⁸ is on a state similar to Eclipse-Theia. Che is the only completely free and open sourced Cloud-IDE.

3 General Benefits and Handicaps

The primary goals of Che are twofold: First it reduces failure when migrating software into the cloud and enhances the experience of working in a rich remote environment. Second is to set up people into existing projects faster and fail safe.

3.1 Reproducibility and Maintenance

The University of Arizona showed [CP16], that about 50% of software inspected was not able to build. Common reasons are missing, changed or faulty dependencies, leading to the common term of *dependency hell* as well as unclear documentation. Experienced programmers are usually able to overcome these issues, given enough time. This continues to be an issue [C114], and without a different approach probably always will be.

This problem is solved, if the software is built inside a docker-container and follows the principles of self-containment and image-immutability [Bi17]. The self-containment principle dictates that at build-time every required dependency is given, and a fully functional image is created. A containerized maven-application would include java, maven and every required maven package to run the application in the current state. The image-immutability means to forbid alternations to the environment. No installations at runtime, no change in ports, no creation of database-schemas. These two principles, while being optional, lead to always reproducible and functional artifacts. The main challenge is to migrate existing software and fulfill these properties. Typically, after making the first iterations of the software on a local machine, containerization will yield to 100 problems at once, not only being frustrating but taking tons of time.

With Che, this kind of Big-Bang-Migration is not possible, as from the very first step it is working inside of a docker-container. The properties above can be held with every change to the image and checked in a normal review-process. Problems will arise one at a time and be solved one at a time, whether they are about code or operations. Che therefore embraces the core-ideas of DevOps [Ja16] and agile software development [Eb16]. That being said, of course it's possible to big-bang-migrate into Che, causing the normal problem mentioned above. However, once the errors are resolved, every further development will work more agile by design.

⁷ <https://aws.amazon.com/cloud9/>

⁸ <https://coder.com/>

The artifacts (whether containers or pods) produced with Che are self-contained and immutable, therefore every result is reproducible. Any further development is incremental on the existing images and keeps old artifacts stable. As the code is fully build inside containers and are already run remotely, the famous “It works on my Machine!” is eliminated. This does not only enhance deployments, but also has a big impact on quality assurance and the motivation of developers. You can keep multiple projects in a single workspace and launch multiple pods at once, running the full environment. The behavior of this multi-pod is unified and testable, making a stable release of full environments possible. This is one of the primary goals of the DevOps approach [Ja16].

3.2 Sharing Results

The topic *docker for reproducible research* has already been promoted [Bo15] and successfully applied to complex topics [St13] such as machine learning. As Clark et al. [Cl14] have used code of several repositories provided from their university, most of the projects can be expected scientific or experimental, therefore 50% of the research results were not reproducible.

The idea to not only ship code, but ship the experiment as a whole makes reproduction much easier. With Che (and using best practices for containers) the experiments done will be containerized and therefore reproducible. Even more complex environments including multiple components can be made in the same paradigms forming a pod.

Che also offers a whole new way of sharing the results: If peers wants to inspect the code, they can be granted accounts and get hands on the experiment without any setup. In an further attempt, a common guest-account can be established for anyone interested with read-only rights. This does not only help cooperation and understanding, but is also a good way to verify the results.

3.3 Faster Setup

Onboarding new teammembers is made easy with Che. A newly joining member does not need to install IDEs, languages, tools and packages, they only needs access to their Che-Account and a singular workspace with every dependency build for them. Depending on the complexity of the project this required at most an hour. While this may seem long, the setup time is dependent on the complexity of the workspace. Setting up a complex workplace will also take more time manually. The work for this is done automatic and remote, so the user is free to dive into the documentation or have some coffee. His workspace will be able to launch the latest version immediately. It is also notable, that the (technical) onboarding of developers and operators is streamlined into one single process.

The combination of a quick setup and the usability of every enhancement makes Che especially interesting to recruit troubleshooters into projects, which solve some of the very specialized tasks. With the artifacts being reproducible, the expert can quickly inspect the subject and analyze problems. With their changes being also in docker, they will be visible and functional for the whole team. Therefore, the hardest common problems regarding *quick help* are addressed and solved. As an additional minor benefit, it is easy to hibernate resources using Che with Kubernetes or Openshift. The workspaces collect metrics like any other container and are therefore scalable on demand. The ability of down-scaling the development resources has a big impact on the overall workload. As a further example, the workspace could be shut down out of normal business-time. This attribute is crucial for a rented infrastructure, which is paid usually on both workload and storage.

3.4 Different Workflow

Additionally, the missing default “build and run” is a paradigm-shift for many developers, which will take a while to get used to. Clark et al. [C114] describe this rather as a change in the mindset, than in actual technology:

"The primary shift that's required is not one of new tooling, as most developers already have the basic tooling they need.

Rather, the needed shift is one of philosophy."

Intuitively this is true: Every developer needs to setup their own workspace – the skills to do this are already there. The only difference is to setup the workspace in a common, documented and descriptive way.

Another problem with Che is the uncommon debugging. As the code is running in a remote-runtime, the common breakpoints and value checks are missing. Instead, one either must be familiar with the languages commandline-debugging or write good logs and metrics. While this will slow down development, it also offers some benefits: The debugging and error-analysis inside the container is the same as it would be in the productive environment. Having meaningful logs with different loglevels and collecting valuable metrics is in general a good attribute for software in any circumstance. It just means additional effort and know-how.

4 Benefits for Education

In addition to the previous mentioned benefits, there are more points about Che that are especially important for educational purposes, such as courses in the university, professional school or research departments. For simplicity only universities will be addressed.

4.1 New motivational and intuitive Courses

Eclipse Che provides the possibility to learn primary cloud-technologies hands on and in a simple and supportive manner. Che makes it possible to teach students fundamentals of modern software development. Depending on the courses subject, one can either provide scalable software or a running environment. With suitable material provided, it's easy to learn cloud native development, DevOps or Cloud-Operations. Even complicated topics, such as micro-services and IoT-applications can easily be adapted inside Che [Iv18] and performed as for example as group-projects with a common goal. The student will need to understand cloud-components as well as team-work to succeed with Che.

The produced artifacts can simply be shipped as containers, giving comfortable access to the teaching staff. With both the product and the code inside the workspace easily accessible, the teaching staff can grant faster and better feedback for the students, which can have massive impact on learning [HSM85][HLK18]. Additionally, the teacher can provide automated end-to-end tests with tools such as Jenkins or TravisCI. Given such an environment, the student will immediately get objective feedback, whether their code fulfills the requirements. This immediate feedback is a big motivator and has a positive impact on the learning experience [Fi05]. With a clear reason on the usefulness of the task, the visible progress and the direct impact of personal effort should lead to intrinsic motivation, which studies show to have a strong effect on education [DR93].

As a requirement, it's necessary to provide a meaningful and fun task. Fortunately, such tasks are not that rare in development. Other studies have shown that only automated and unified testing is not beneficial to education in general [Cr88]. But the tools provided by Che and CI/CD slim down the overhead until both personal feedback and automated can be supplied. If the whole classroom utilizes this system, the educator can also monitor who is having trouble in an automated manner. With a single investment for the tests, feedback for every student is granted at every time. With the single investment of providing a running Che instance (and a short lecture how to use it), feedback on the artifacts and code can be immediately granted to every student, without having the overhead of setting up the workspace. These investments can be done before the course starts at any time and can be reused.

4.2 Equality and Accessibility

With access to a university Che workspace, every person can develop the same software in the same environment. A well known problem is the necessity of an IOS-build-device for IOS-appdevelopment. Students which are limited in their access to the university-laboratory or cannot afford a mac are therefore handicapped. Another rising problem are GPU-accelerated machine learning algorithms. Depending on the task, training neural networks can be 20 times faster if a CUDA-compatible graphics card is used. While these

technologies are great and important for many applications, the required technologies are exclusive due to their costs.

This hindrance for education is usually addressed with IOS-laboratories and/or cloud-resources paid for by university. The topic of socio-economic factors is beyond the scope of this work, but the impact has been analyzed by the PISA Studies [OE15b], which state that socio-economically handicapped students fail three times more likely to achieve a basic level of performance. According to PISA, this rate has not significantly changed since the studies started in 2006. Germany scored better on the impact of socioeconomic-factors on education in 2015 than it did in 2006 but is worse than the global average [OE15a]. The Mac-Laboratories and rented cloud space are technically a good attempt at inclusion. But students which are required to get to university to do their assignments are less likely to do so. Making assignments mandatory, providing extrinsic motivation [DR93] has usually a negative impact on the subjective experience.

A different problem comes with machine-learning, which is usually a process of trial and error to some extent. Long waiting times disconnect the action from the result being not beneficial for the learning process [HSM85][DR93].

With studies showing basics of good learning experiences [Fi05][Cr88] and the impact of motivation [HSM85][DR93], it is mandatory for institutions to provide the best possible environment. While many socio-economic factors cannot be grasped at the university or by the educators, those originating in the students' economic background can be addressed. Most universities already provide free services, such as software-suites, e-libraries or cloud storage.

Simply providing a technical correct solution (like laboratories) is not enough to provide a good environment. One student with a better socio-economic background who can afford a 300\$ GPU will have a better baseline for their education. This is simply put unfair. With Che, universities can provide a free, good and accessible development environment for every student.

5 Conclusion

Cloud technology will be dominant in the foreseeable future. Not *everything* will be best fit for cloud, but most of the everyday software benefits from the concepts. With more and more tools making it easier, it's just a matter of time until cloud-tec becomes standard instead of bonus.

With Eclipse Che many problems about teaching students these Cloud Skills and reducing the initial scope for learning, are addressed. While the impacts on motivation and the actual learning results are only hypothetical, most of the technical requirements are solved by Che. As Eclipse Che is cutting edge it whatsoever lacks many quality of life features, has some bugs, and the preferred design is undefined to some extend.

It somewhat is a chance: Instances such as a university can have massive impact on the design of Che and drive the project to some extend for their own needs. With faculties working on the open source project, either providing code and features, or just testing Che in educational environments, both quality issues and missing features can be solved. Now cloud development is only for corporations. There is hardly any chance to successfully learn it at university. With Che there is another chance to “get back into the game” – and provide corporation-free education.

While this sounds harsh towards companies, unbiased education for students is not a nice thing to have, but a core concept of education in general. Only if that is true, the students can make unbiased decisions and make unbiased research. Being dependent on Google for Universities is in a similar way disturbing, as if a biology faculty would be sponsored by Bayer. One can only imagine a grim future if the curriculum is influenced by profit.

Future Work

As this paper only argues about the possible improvement of education using Che, it should be therefore tested. A study could be hold on both the acceptance and effect of a CI/CD based development course with Che in comparison to a conventional course. These studies could be done with universities and corporations, which may yield very different results.

Additionally, it should be inspected whether the studies on general motivation and education hold for IT-students.

When moving the scope from student to educator, a study can be hold about the time spent of the teacher on which part of the course and whether a positive effect on effective feedback-time can be achieved.

Bibliography

- [Bi17] Principles of Container-Based Application Design. <https://www.redhat.com/cms/managed-files/cl-cloud-native-container-design-whitepaper-f8808kc-201710-v3-en.pdf>, last seen 2019-05-04.
- [Bo15] Boettiger, Carl: An Introduction to Docker for Reproducible Research. SIGOPS Oper. Syst. Rev., 49(1):71–79, January 2015.
- [Cl14] Clark, Dav; Culich, Aaron; Hamlin, Brian; Lovett, Ryan: BCE: Berkeley’s common scientific compute environment for research and education. In: Proceedings of the 13th Python in Science Conference (SciPy 2014). pp. 1–8, 2014.
- [CP16] Collberg, Christian; Proebsting, Todd A.: Repeatability in Computer Systems Research. Commun. ACM, 59(3):62–69, February 2016.
- [Cr88] Crooks, Terence J.: The Impact of Classroom Evaluation Practices on Students. Review of Educational Research, 58(4):438–481, 1988.
- [DR93] Deci, E.; Ryan, R.: Die Selbstbestimmungstheorie der Motivation und ihre Bedeutung für die Pädagogik. Zeitschrift für Pädagogik, 39(2):223–238, 1993.
- [Eb16] Ebert, C.; Gallardo, G.; Hernantes, G.; Serrano, N.: DevOps. IEEE Software, 33(3):94–100, May 2016.
- [Fi05] Fink, L.: Creating Significant Learning Experiences : An Integrated Approach to Designing College Courses / L.D. Fink. 01 2005.
- [HLK18] Hutfilter, A.; Lehmann, S.; Kim, E. J.: Improving skills and their use in Germany. (1516), 2018.
- [HSM85] Hughes, Billie; Sullivan, Howard J.; Mosley, Mary Lou: External Evaluation, Task Difficulty, and Continuing Motivation. The Journal of Educational Research, 78(4):210–215, 1985.
- [Iv18] Developing Multi-Pod Apps with Kubernetes and Che. <https://che.eclipse.org/using-eclipse-che-to-develop-multi-container-apps-eb45b32ffe5b>, last seen 2019-05-04.
- [Ja16] Jabbari, R.; bin Ali, N.; Petersen, K.; Tanveer, B.: What is devops?: A systematic mapping study on definitions and practices. In: Proceedings of the Scientific Workshop Proceedings of XP2016. ACM, p. 12, 2016.
- [Me11] Mell, P.; Grance, T. et al.: The NIST definition of cloud computing. 2011.
- [OE15a] Country Note Germany – Results from PISA. <http://www.oecd.org/pisa/PISA-2015-Germany.pdf>.
- [OE15b] PISA 2015 – Ergebnisse im Fokus. http://www.oecd.org/berlin/themen/pisa-studie/PISA_2015_Zusammenfassung.pdf.
- [St13] Stodden, V. et al: Reproducibility in computational and experimental mathematics - . Examining reproducibility in computer science. 2013.
- [St19] Stack Overflow Developer Survey 2019. <https://insights.stackoverflow.com/survey/2019>, last seen 2019-05-04.
- [To17] Toffetti, Giovanni; Brunner, Sandro; Blöchliger, Martin; Spillner, Josef; Bohnert, Thomas Michael: Self-managing cloud-native applications: Design, implementation, and experience. Future Generation Computer Systems, 72:165 – 179, 2017.

aMAZEing Programming — Providing SKILLs to Fellow Students

Timo P. Gros,¹ Pascal L. Held,¹ Pascal Lauer,¹ Niklas O. Metzger,¹ Kallistos Weis¹

Abstract: Learning programming can be hard, especially for inexperienced freshman. An aMAZEing team of 30 students decided to lighten that challenge for their fellow students. This paper will present the attempt of volunteers to prepare their fellow students for their programming course, resulting in a week full of lectures, projects, coachings and didactical effort.

Keywords: students preparatory course; tutor preparatory workshop; programming; didactics; domain specific language

1 Introduction

Learning how to code is an obligatory task for every computer scientist. Starting without any programming skills is the reality for many students in early semesters. At Saarland University, two mandatory courses are supposed to teach the basics of programming where the focus is on theoretical aspects and the pace of the lecture and required exercises is very high. Therefore, starting four years ago, the first of our team members had the vision to improve the basic lecture “Programmierung 2”. Enjoying the trust of Prof. Sebastian Hack, the teaching professor, we started to organize a one-day students preparatory course (SPC) and beforehand a four-day tutor preparatory workshop (TPW). The latter was meant to prepare the tutors for their tutoring job and to improve their teaching ability. The SPC had the intention to teach the basics of Unix and Git, such that during the following semester the students can concentrate on the programming tasks without having the need to waste their cognitive abilities on technical prerequisites. We observed a significant increase in the number of students passing the course.

During the last four years, the team of current and former tutors kept improving the SPC. Still, many students were overwhelmed by the task of learning multiple programming languages within one course. Therefore, we decided to try to ease that challenge. We wanted to organize a one week preparatory course, teaching the students not a specific programming language, but especially *how to learn* a new programming language. That was when the aMAZEing Programming Course (APC) was born. For the APC we developed a domain specific language to solve mazes. We decided to not use an existing learn-to-program project as we wanted the project to suit our intentions exactly. The project we developed for the APC can be found on <https://amazeing.de>.

¹ Affiliations including {timopgros, s9paheld, s8palaue, s8nimetz, s9ksweis}@stud.uni-saarland.de

Study regulations According to the study regulations at Saarland University, in their first semester the students participate in the lecture “Programmierung 1”. This may raise the question, why a course as the APC is needed between the first and the second semester. The Programmierung 1 lecture deals with functional programming only [Sm08]. In opposite, the lecture of the second semester, called Programmierung 2, deals with imperative programming languages, such as C and Java. Even though the students already had a different — functional — programming course, our experience shows that students in Programmierung 2 are overwhelmed by learning multiple programming languages, especially at the beginning of the course. This is where the APC can help: teaching how to learn new programming languages and technical skills efficiently.

Statistics The APC took place one week before the start of the lectures, i.e. from Monday, April 1st to Friday, April 5th. There were 316 students participating. In comparison, the Programmierung 2 course has 540 participants. Considering that the course traditionally has many students reparticipating in the Programmierung 2 course and thus not visiting a preparatory course, we reached a large majority of the target students.

Our team consisted of 13 tutors, 11 coaches that mainly are alumni tutors and 6 supervisors. Of the latter, 2 were holding the lectures, 2 organized the aMAZEing project (AP) and 2 held the TPW.

Paper outline In the following, the organization of the aMAZEing Programming course is presented in section 2. section 3 and section 4 show the lectures and the aMAZEing project, respectively. In section 5 we provide details about the latest iteration of the tutor preparatory workshop. Afterwards, section 6 evaluates the first iteration of the APC and section 7 concludes this paper.

2 Organization

In this section, we present the organization of the APC. The timetable for the students and the APC team is shown as well as the content of tutorials and coachings.

As mentioned in section 1, aMAZEing Programming was a week-long course for the preparation of students before their second semester at Saarland University. The timetable of this event is pictured in Figure 1a. The four main parts that were relevant to the students are lectures, coachings, office hours and tutorials. In parallel to the APC, the tutor preparatory workshop took place but is not shown in this timetable. The schedule of the TPW is pictured in section 5. After the start of the day with the lecture, the daily project together with its documentation was published. In the first coaching phase, the students worked on the project by themselves, unassisted except for a small group of coaches that were supposed to help with technical challenges. After the lunch break, the office hour took place with tutors

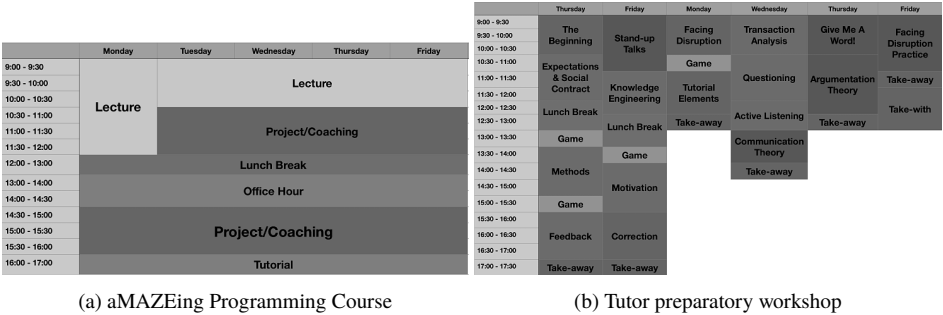


Fig. 1: Timetables

actively supporting the students and answering questions. After a second coaching period, the solution to the project combined with additional exercises was presented in a tutorial.

To maximize the content as well as the knowledge of the participants, we split the day into multiple iterations of assisted and unassisted working phases. Therefore, the team of current and alumni Programmierung 2 tutors alternated in every time slot. In the following, we present the organization of coaching and tutorials, lectures are shown in section 3 and the tutor preparatory course follows in section 5.

2.1 Coaching

The coaching slot corresponds to the unassisted self-learning and -working phase for students. To prepare for the Programmierung 2 projects, each programming exercise was to be worked on by every student. However this does not mean that they were forced to work quietly on their own since discussing implementation ideas in small groups was always preceding the programming work. This helped to revisit their thoughts and potentially increase the quality of the ideas. We have noticed that the working phases led to friendships that we were still able to recognize throughout the Programmierung 2 course. Thus we are sure that our setting caused people to work together from the start in contrast to the observations made by Porter et al. [Po13]. Since technical issues were to be expected with the distribution of a virtual machine² running on multiple host systems, a group of up to 6 coaches was spread over all working areas for the students. They focused on fixing unintended software behaviour without helping to find a solution for the actual exercises. On top of that, they were always there to clarify parts of the assignment. Since “students have trouble following written instructions” this helped to avoid misunderstandings [Gi03]. During the coaching, the project team directly implemented and handled technical feedback. Therefore, the tutors in the office hours were prepared to help with common problems and answer questions.

² used to create a common starting point for the projects

2.2 Tutorial

In a project that is based on previous days, having a correct implementation before the next project release is necessary. Therefore, and for the convenience of the students, the end of the day was a tutorial in which the actual solution was presented. Furthermore, some independent exercises targeting common problems were treated. The tutor team that accompanies the Programmierung 2 course was also responsible for the APC tutorials. The new tutors were able to practice the knowledge learnt in the TPW and also get to know the students that will be present during the following semester. The content of the tutorials was prepared in team work as part of TPW which is dual to the mode of operation during Programmierung 2, where working as a team has a high priority.

3 Lectures

In this section, we present the concept of the aMAZEing Programming lectures. The structure as well as the topics provided by the lecturers is presented. We introduce the aMAZEing instruction language (AIL) as key contribution of this course.

The lectures of aMAZEing Programming are the start of every course day. The necessary background for each project day is explained to the participants. We provide a theoretical foundation as well as specific programming-language dependent information that enables to solve the projects and act as a linkage from aMAZEing Programming to the following course Programmierung 2. Since working synchronized on the servers provided by the university, knowledge of the given infrastructure is the starting point for the lectures. From day two on, the lectures are concerning Python, the aMAZEing instruction language and algorithms as well as programming concepts for the aMAZEing project. Every lecture ends with the presentation of the daily project with technical and conceptual information.

3.1 Unix & Git

Unix & Git are introduced on the first day of the course. Due to the amount of necessary commands that is inevitable for the efficient usage of a Unix system, the lecture of the first day lasts until the lunch break. The interactive lecture combines slides with information and on the fly execution of shell commands. Throughout the course, the poll system **DirectPoll**³ works as interface between the lecturer and the students. Thereby, the audience's understanding can be checked and answering the questions provides short breaks that increase the attention. For a similar approach, namely Peer Instruction, there are observations that teaching with this technique clearly outperforms standard classes [Po13]. The slides are built to deliver a documentation that is complete w.r.t. the knowledge requirement of the course Programmierung 2 for Unix and Git.

³ <https://directpoll.com>

3.2 Python

The participant's implementations that cannot be realised with the AIL in the front-end are written with the back-end language, i.e. Python. This highly used white-space sensitive language was chosen because of the readability and functionality of written code. In the lectures, syntax as well as explicit semantic definitions are presented with respect to the Python documentation [Py19]. On top of that, the main datastructures (i.e. lists, tuples, sets and dictionaries) and their advantages and disadvantages are part of the lecture, even though they are not (yet) part of the AIL. Since File I/O is a major topic in the Programmierung 2 course, it is also part of the aMAZEing project and introduced in the lectures combined with the concept of assumptions to validate input parameters. Thereby, a first glimpse for testing and correct software is presented to the students.

3.3 AIL: aMAZEing Instruction Language

The instruction language AIL is designed to suffice for implementing game-winning/maze-solving algorithms of the aMAZEing project. It is based on a stepwise execution of instructions defining the movement of a player in the labyrinth combined with conditional jumps to labels. The algorithms that are chosen to be implemented with AIL range from simple *Wall Following* to *Depth-First-Search* on the labyrinth. The AIL syntax is presented in Figure 2. Since the first project of the Programmierung 2 course has to be implemented in MIPS [He82], the structure of AIL is inspired by a low level machine language. To keep the focus on elaborating an algorithmic understanding AIL contains no sophisticated instructions. With this motivation, we prepare the students for the unintuitive low level language with a game based instruction language that abstracts from confusing arithmetic conventions.

```

Statement ::= move | turn TurnDirection | Label:
           | branch Label Label | store direction
           | test direction | test wall PlayerDirection
           | jump Label | call Label | return
Label      ::= [a-zA-Z]+
TurnDirection ::= right | left
PlayerDirection ::= TurnDirection | front | back

```

Fig. 2: Grammar of AIL

4 Projects

In this section we present the concept of the projects used in the APC. The tasks as well as the learning objectives for the students are presented.

We published the source code of the project on [gitlab.com](https://gitlab.com/amazeing/project)⁴. Figure 3 depicts the WebUI used for implementing tasks in AIL.

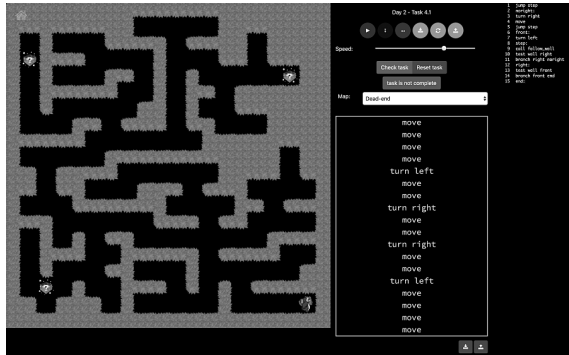


Fig. 3: Project WebUI of an AIL programming task.

The projects are the main part of every course day. The students experiment with the theoretical foundations provided by the lecture in a practical way to deepen the concepts mentioned in the lecture. As mentioned in section 3 the students work synchronized on the servers of the university such that the projects on the first day focus on the provided infrastructure. From day two on, the students work with one project that is described in more detail in subsection 4.2.

4.1 Git & Unix

Git & Unix are two short projects where the students practice the basics on how to work with projects under version control. The students explore the file system of Unix systems and how to create, copy and move files. They learn the most important things to work with Git.

4.2 aMAZEing Prog2

Motivation “The only way to learn a new programming language is by writing programs in it” quote by Dennis Ritchie. Most of the programming beginners do not know how to start their program. They fail to write the first statement because they do not try to solve the problem. Instead they think about how to solve the whole problem without doing it step by step. We try to encourage them by this project to solve problems step by step and do not fear to write and rewrite programs. “Playing games provides a visual representation to the abstract nature of the problem” [Ra05, p. 99]. This inspired us to provide a game like environment so students find it easier to come up with solutions to the given problems.

⁴ <https://gitlab.com/amazeing/project>

Tasks To introduce the project to the students, they had to implement solutions for specific mazes in AIL as well as in Python by using the provided instructions and functions. In the next part, the students automate solving mazes by implementing algorithms in both languages, AIL (List. 1) and Python (List. 2). Figure 4 presents 3 exemplary algorithms whereby the last algorithm was not to be implemented in AIL. As one can see, the corresponding AIL implementation is really short and not hard to come up with. Similar exercises form an easy start and create positive first experiences. The successful start leads to a better performance throughout the remaining tasks [Wi05]. Furthermore the students had to implement a parser for mazes and had to check constraints to reject wrong input, i.e. wrong mazes.

- *Move until obstacle*: Move in a straight line as long as there is no obstacle (wall) in front of the player.
- *Wall following (right hand rule)*: Repeatedly follow the wall to the right of the player and turn right if wall ended and left otherwise until the goal is reached.
- *Depth-First-Search*: Nodes are the crossings in the maze and edges are the routes between the crossings.

Fig. 4: Selection of algorithms the students were ask to implement

```

from server.model import algorithms

def move_until():
    if algorithms.test_goal():
        return

step:
    test wall front
    branch end next
next:
    move
    jump step
end:
    return

x, y = algorithms.get_player_position()
x_off, y_off, name = algorithms.get_player_direction()

while not algorithms.test_wall(x + x_off, y + y_off):
    algorithms.move()
    x, y = algorithms.get_player_position()
    if algorithms.test_goal():
        return

```

List. 1: Move until obstacle
in AIL.

List. 2: Move until obstacle in Python.

We used web technologies as they are mostly platform independent and easily accessible. WebSockets gave us a reliable and simple two-way communication channel between front- and back-end. Instead of using raw WebSockets we used SocketIO which gave us an event based abstraction layer. The front-end code is standard HTML and ECMAScript 6 code. For the back-end we implemented the necessary routes (front-end delivery) using Flask and an SocketIO endpoint.

Used Resources We implemented the project using the following technologies and tools:

- WebSockets (using SocketIO⁵) [FM11]
- JavaScript (using yarn⁶, Webpack Encore⁷ and SocketIO³)
- Python 3 (building on top of Flask⁸ (-SocketIO⁹) and gevent¹⁰) [Py19]

5 Tutor Preparatory Workshop

One of the most important aspects of a lecture is its tutorials. The tutorials can make the difference between a really good and a really bad lecture. Thus, it is not only important to find good and motivated tutors, but also to prepare them as good as possible, especially if its their first time giving tutorials.

Simultaneously to the first SPC four years ago, the first TPW took place. The intention of creating such a workshop was for one thing to lower the time needed for familiarization of the tutors, such that they can start giving their very best with the first week of the semester. Furthermore, we observed through the years and several basic lectures, that the assignment to different tutors can have a major impact on the chance of passing the course. Thus, it also was intended to bring all the tutors to a similar state of knowledge regarding teaching, decreasing that impact as good as possible. On top of that, team building within the team of tutors was another important, if not even the most important, consideration.

The workshop consists of a mix of educational science, rhetorical exercises and best practice of tutoring. The latter being accumulated by many tutors through the years of giving tutorials, the rhetorical exercises being collected from and with professional coaches and the educational science being provided by our team members who studied not pure computer science but how to teach it.

5.1 Timetable and Didactical Concept

In the last years, the workshop mostly was given within four days of full time work before having the students preparatory course on Friday. With the APC, the new concept of the SPC, there was the need to find a new structure for the tutor preparatory workshop.

The workshop traditionally starts with some basics that are needed for further concepts. Thus, it is really important to have some days with the tutors having the workshop only, such that they can concentrate on their training without needing to work for the APC. On

⁵ <https://socket.io>

⁶ <https://yarnpkg.com>

⁷ <https://symfony.com/doc/current/frontend.html>

⁸ <http://flask.pocoo.org>

⁹ <https://flask-socketio.readthedocs.io/en/latest/>

¹⁰ <http://www.gevent.org>

the other hand, the APC's new structure offers an unique chance: as the tutorials take place in the afternoon only (and the lectures, office hours and coachings are supervised by others), we have the opportunity to teach the tutors in the morning and let them practice in the afternoon.

To combine both, we decided to first have two days (Thursday and Friday of the week before the APC) of full time tutor preparation and additionally use the mornings from Monday to Friday during the APC.

The timetable is displayed in Figure 1b. Hereby the colors give us the categories of the modules: ■ green for educational science, ■ blue for best practice and ■ purple for rhetorical exercises. Notice that the first days mostly contain modules of educational science, steadily decreasing with the beginning of the third day. The number of modules of rhetorical exercises and best practice in opposite increase over the workshops days. Furthermore, you can find team building games between the modules, not only meant for building a team but also to clear the mind of the tutors.

All the modules were designed with respect to the principle of independent learning: the instructions and talks were kept as short as possible. Instead, the focus was on group discussions and practical exercises. Simultaneously to pupils it holds: what is discovered or elaborated by the tutors themselves is learned deeper and better as if the same fact was just presented to them.

5.2 Take away — Take with

As you can see in Figure 1b, every day is ended with a short module called Take-away. Every student writes down some concrete actions they would like to remember during their work as a tutor on a card in a specific color. The cards then are collected. On the last day of the TPW, the tutors sit together in groups of size 3 or 4 and group the actions that were handed in several times. Then a moderator chosen from the group leads the discussion to rank the maps from most important to less important. These 10 concrete actions that can help being a better tutor can be revisited during the semester. Furthermore, the ranking was placed on the wall of the tutors lab.

5.3 Summary

The tutor preparation workshop does not only form a working team within a really short period of time, but also prepares the tutors in both, in a theoretical and a practical kind, for the upcoming job. According to their feedback, they feel more comfortable and more secure giving their first tutorials. During the last four years of preparing our tutors, the evaluation grades of our tutorials have increased significantly.

6 Evaluation

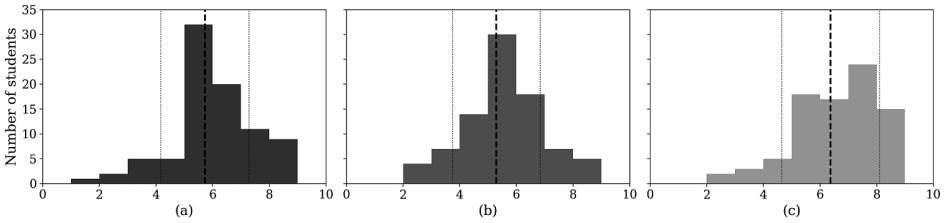


Fig. 5: (a) lecture speed; (b) lecture content; (c) project difficulty; dashed: mean; dotted: mean \pm standard deviation

As the summer semester has just started, there are no statistics available that could be presented to draw a conclusion whether the APC was successful with its intention to prepare our fellow students for their original programming course. Instead, we focus on internal and external feedback given by team members and students, respectively, and a preview of the APC’s future.

We collected feedback by our students through a short form which was handed out and collected back in at the last day of the APC. We asked them to rate the pace of the lecture from too slow (1) to too fast (10), the content of the lecture from too less (1) to too much (10) and the difficulty of the project from too easy (1) to too hard (10). As the feedback of our students (see Figure 5) was mostly positive — the pace with average 5.72 and the content of the lecture with 5.29 were rated nearly perfectly — we plan to keep repeating the APC in a very similar way in upcoming iterations. The projects had an average value of 6.37, i.e. our students think, that the projects were a little bit too difficult. Thus, we plan some changes on the project.

6.1 Project

Our students rated the projects to be too difficult. We trace this back to the circumstance, that we used two different languages for the projects. The students first learned how to use AIL and later on had to use Python. While our intention initially was to show a real programming language, we actually do not see any didactical sense in conflicting them with more than one new language within one week. For future iterations, the AIL will be revised and extended, such that we not only can use this language, but also the language is exactly adapted to the content we would like to teach in an easily accessible and comprehensible way.

More on, we would like to revisit the AP itself. One major idea is to additionally introduce a multiplayer mode which can be used from students to play against each other. To do this without overcharging our team, we want to hand in the development of the project to the lecture “Software Engineering”. This lecture gives the possibility to companies and chairs

to submit projects. These projects will then be handled by a student team. We want to benefit from this chance to even improve the AP.

6.2 Lectures

In this years first instance of the APC “Testing” was not a part of the lecture. Still, it is an important topic of the following programming course. After the decision to not handle Python in future iterations and solely relying on AIL, we can reinforce our focus on testing in future iterations. The AIL language will be expanded in such manner that there will be a good way to teach testing and furthermore integrate testing into the AP.

6.3 Tutor Preparation Workshop

The concept of the TPW has had more iterations than the APC. Nevertheless, there is always the need to revisit the modules and keep them up to date.

While the concept itself is well known and well rehearsed, this year was the first time to enlarge the seminar, such that the tutors had the chance to practice their theoretical findings in the afternoon. As the tutors feedback in this regard was consistently positive, this format shall be kept for future iterations.

7 Conclusion

In this paper we presented the aMAZEing Programming course, an attempt to provide SKILLS to fellow students, preparing them for the programming courses at Saarland University and especially for Programmierung 2. We developed the aMAZEing instruction language, an easy domain specific language reduced to the fragments of programming we would like to teach and applied it to the aMAZEing project. The required knowledge was taught with especially developed lectures. To provide even further support, there were coachings and tutorials, the latter given by our tutors that were specially trained for this by the tutor preparation workshop. Even though it is hard to compare different iterations of Programmierung 2¹¹, one of the changes was so significant that it cannot be ignored. Namely the amount of feedback from students about our tutors’ commitment. It seems like in this iteration students increasingly realized how many people are there to help them up if they fall down. By [Wi05] we conclude that this led to a boost of their self-efficacy and so performance in the course overall.

¹¹ The required coursework in terms of projects changes from iteration to iteration.

Acknowledgement

We are grateful for the confidence that the chairs Compiler Design Lab of Prof. Sebastian Hack and Foundations of Artificial Intelligence of Prof. Jörg Hoffmann have shown us. Thank you for giving us the chance to independently organize and execute the APC.

The most important shall be emphasized at the very end: without our team and their voluntary commitment, none of this would have been possible. We want to thank S. Ariesanu, J. Baumeister, S. Ehses, C. Eimer, G. Geier, J. Görgen, P. Hennen, E. Hoffmann, C. Kapp, D. Kempter, L. Kirschner, J. Klesen, D. Luche, N. Medinger, H. Meerkamp, M. Meiers, S. Oswald, J. Petermann, J. Schmidt, C. Schwan, L. Timm, S. Wack, F. Walgenbach, J. Warken and D. Weber. You are truly aMAZEing.

References

- [FM11] Fette, I.; Melnikov, A.: The WebSocket Protocol, RFC 6455, RFC Editor, Dec. 2011, URL: <http://www.rfc-editor.org/rfc/rfc6455.txt>.
- [Gi03] Giguette, R.: Pre-games: games designed to introduce CS1 and CS2 programming assignments. In (Grissom, S.; Knox, D.; Joyce, D. T.; Dann, W., eds.): Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education, 2003, Reno, Nevada, USA, February 19-23, 2003. ACM, pp. 288–292, 2003, ISBN: 1-58113-648-X, URL: <https://doi.org/10.1145/611892.611990>.
- [He82] Hennessy, J. L.; Jouppi, N. P.; Przybylski, S. A.; Rowen, C.; Gross, T. R.; Baskett, F.; Gill, J.: MIPS: A microprocessor architecture. In (Fisher, J. A.; Tracz, W. J.; Hopkins, W. C., eds.): Proceedings of the 15th annual workshop on Microprogramming, MICRO 1982, Palo Alto, California, USA, October 5-7, 1982. ACM/IEEE, pp. 17–22, 1982, URL: <http://dl.acm.org/citation.cfm?id=800930>.
- [Po13] Porter, L.; Guzdial, M.; McDowell, C.; Simon, B.: Success in Introductory Programming: What Works? Commun. ACM 56/, pp. 34–36, Aug. 2013.
- [Py19] Python Software Foundation: Python 3.7.3 documentation, 2019, URL: <https://docs.python.org/3/>.
- [Ra05] Rajaravivarma, R.: A games-based approach for teaching the introductory programming course. SIGCSE Bulletin 37/4, pp. 98–102, 2005, URL: <https://doi.org/10.1145/1113847.1113886>.
- [Sm08] Smolka, G.: Programmierung - eine Einführung in die Informatik mit Standard ML. Oldenbourg, 2008, ISBN: 978-3-486-58601-5.
- [Wi05] Wiedenbeck, S.: Factors affecting the success of non-majors in learning to program. In (Anderson, R. J.; Fincher, S.; Guzdial, M., eds.): International Computing Education Research Workshop 2005, ICER '05, Seattle, WA, USA, October 1-2, 2005. ACM, pp. 13–24, 2005, ISBN: 1-59593-043-4, URL: <https://doi.org/10.1145/1089786.1089788>.

Vorhersage der Ballbewegung in der RoboCup Standard Platform League

Carolin Gümpel¹

Abstract: Die Ballerkennung ist ein essentieller Bestandteil im Roboterfußball. Anhand der Position des Balles entscheiden die Roboter über ihre Bewegungen und es wird die beste Spielstrategie festgelegt. Doch schon durch einen sanften Schuss kann sich die Ballposition schnell um eine weite Distanz verändern. Für ein geschicktes Verhalten der Roboter in diesem Moment ist es von Vorteil, abschätzen zu können, wohin genau der Ball rollen wird. Vor allem der Torwart muss jederzeit genau wissen, ob und wann der Ball das Tor erreicht. In dieser Arbeit wird ein neuartiges Modell zur Ballvorhersage mit speziellem Fokus auf die Anwendung in der RoboCup Standard Platform League vorgestellt. Dabei wird besonders Wert auf die Vorhersage der Stoppposition des Balles gelegt. In ersten Experimenten erzielte das Modell eine Erfolgsquote von 84 %.

Keywords: Roboterfußball; Ballvorhersage; Standard Platform League

1 Einleitung

Im Bereich Roboterfußball haben sich inzwischen viele verschiedene Ligen entwickelt. Dazu zählen unter anderen die Humanoid League, Standard Platform League, Middle Size League und Small Size League [Ro12]. In jeder Liga gibt es unterschiedliche Spielbedingungen und somit spezielle Herausforderungen. In der RoboCup [Ki98] Standard Platform League spielen alle Teams mit identischen Robotern. Diese agieren völlig autonom und jeder Roboter trifft seine eigenen Entscheidungen. Trotzdem müssen die Roboter durch gute Kommunikation als Team zusammenspielen. Eine der wichtigsten Komponenten beim Zusammenspiel im Fußball ist das Passen [Ko07]. Dazu muss nicht nur der Roboter, der den Pass ausführt, den Ball in die korrekte Richtung spielen. Der Roboter, der den Ball annimmt, muss außerdem möglichst genau abschätzen können, wohin der Ball rollen wird. Im Vergleich zu menschlichen Fußballern haben die Roboter beim Passen noch eine deutlich geringere Erfolgsquote [Ab11].

Die Vorhersage, wo der Ball stoppen wird, ist auch für den Torwart extrem wichtig. Er muss schnell entscheiden können, ob und wo der Ball ins Tor rollen wird, um rechtzeitig eine Abwehrbewegung einleiten zu können. Eine mögliche Reaktion des Torwarts auf einen auf das Tor zukommenden Ball ist, sich nach rechts oder links zu werfen, um den Ball aufzuhalten. Dies führt jedoch zu schnellem Verschleiß des Roboters. Außerdem braucht der dann liegende Roboter ein paar Sekunden, um aufzustehen und um das Tor wieder schützen zu können. Aus diesen Gründen gilt es unbedingt zu vermeiden, dass der Roboter

¹ HTWK Leipzig, IMN, Karl-Liebknecht-Str.132, 04277 Leipzig carolin.guempel@stud.htwk-leipzig.de

sich aufgrund einer falschen Einschätzung der Ballstopposition unnötig auf den Boden wirft.

Eine weitere Situation, bei der die Ballvorhersage eine wichtige Rolle spielt, ist das Treffen der Entscheidung, wohin der Roboter laufen soll, um zum Ball zu gelangen [Ri08]. Die einfache Variante besteht darin, den Roboter immer direkt auf den Ball zulaufen zu lassen. Bei einem Ball in Bewegung entsteht so allerdings eine Kurve im Laufweg des Roboters. Dadurch geht wertvolle Zeit verloren und der Gegner ist eventuell schneller am Ball.

Wie man sieht, ist die Vorhersage der Ballbewegung im Roboterfußball ein wichtiger Aspekt. In der Standard Platform League besteht die Schwierigkeit bei der Ballvorhersage unter anderem darin, dass auf Kunstrasen und mit einem realistisch aussehenden Fußball gespielt wird. Dies macht die Vorhersage deutlich schwieriger als sie es auf glattem Boden mit einem regelmäßigen Ball wäre.

In dieser Arbeit wird der Bewegungspfad des Balles der RoboCup Standard Platform League untersucht. Auf Basis der Erkenntnisse zum Verlauf des Ballpfades und zu Kräften, die die Bewegung beeinflussen, wie zum Beispiel Luftwiderstand und Rollreibung, wurde ein Modell zur Ballvorhersage entwickelt. Der Fokus liegt dabei auf der Vorhersage der Stopposition des Balles. Der Bereich, in dem ein rollender Ball stoppen wird, wurde von dem entwickelten Modell bei 100 Versuchen 84 Mal korrekt vorhergesagt.

2 Stand der Wissenschaft

[Su14] beschreibt eine Methode zur Ballvorhersage im Roboterfußball auf Basis von Trigonometrie. Die Ballposition muss dazu in Koordinatenform vorliegen. Es werden für jeden Frame der Kameraaufnahmen die Differenzen zwischen den x- und y-Werten der aktuellen Ballposition und den entsprechenden Werten aus dem vorhergehenden Frame gebildet. Es wird dann ein rechtwinkliges Dreieck aufgespannt, bei welchem die berechneten Differenzen die Länge von Ankathete und Gegenkathete sind. Die Hypothense, welche dem zurückgelegten Weg des Balles entspricht, wird über die beiden Katheten berechnet. Aus dem so bestimmten Weg s , der initialen Ballgeschwindigkeit u , welche beim ersten Durchlauf auf Null festgelegt wird, und der Zeit t , welche einem Frame entspricht, kann die Beschleunigung a durch Formel 1 berechnet werden.

$$a = \frac{2(s - ut)}{t^2} \quad (1)$$

Die aktuelle Ballgeschwindigkeit v wird dann durch Formel 2 bestimmt.

$$v = u + at \quad (2)$$

Um die neue Position des Balles vorherzusagen, wird bei dieser Methode von einer konstanten Beschleunigung des Balles ausgegangen. So kann die Distanz, die der Ball im nächsten Frame zurücklegen wird, über das Weg-Zeit-Gesetz berechnet werden. Es wird außerdem angenommen, dass der Ball auf einem glatten Untergrund ohne Hindernisse rollt

und somit seine Richtung nicht ändert. Wird die Hypothenuse des aufgespannten Dreiecks um die berechnete Distanz verlängert, kann die vorhergesagte Ballposition am Endpunkt der Strecke abgelesen werden.

Diese Methode ermöglicht eine gute Ballvorhersage, allerdings nur für den jeweils nächsten Frame. Richtungsänderungen des Balles, welche durch die Rotation entstehen, werden nicht einbezogen. Auch der Luftwiderstand wird vernachlässigt.

In [Su14] wird außerdem Merlins Methode zur Ballvorhersage vorgestellt, welche von der Merlin Corp. entwickelt wurde. Diese Methode berechnet die Distanz zwischen den x- und y-Werten der Ballposition aus dem aktuellen Frame und dem vorhergehenden Frame und addiert diese Distanz für die Vorhersage auf die jeweiligen Werte der aktuellen Position. Diese Methode ist schnell in der Berechnung, aber laut [Su14] für langsame Bälle ungeeignet.

Eine weitere Methode zur Ballvorhersage wird in [Li02] in Form eines Echtzeitkollisionsmodells vorgestellt. Ziel dieses Modells ist die Vorhersage von Kollisionen zwischen einem Roboter und anderen Objekten. Diese können ein anderer Roboter, eine Wand oder auch der Ball sein. Für die Vorhersage wird ein Kollisionsvorhersagevektor verwendet. Dieser besteht aus den vorausgesagten Zeiten bis zur Kollision mit den verschiedenen Objekten. So kann jeder Roboter seine Bewegungen anhand des Vektors planen. Ausprobiert wurde dieses Verfahren in einem MiroSot² Roboterfußballsystem. Ein MiroSot-Spiel ist sehr schnell. Die Roboter werden von einem Spielcomputer gesteuert. Dieser bietet die für das Echtzeitkollisionsmodell benötigte Rechenleistung, welche die Nao-Roboter der Standard Platform League nicht aufbringen können.

Auch [Bu06] hat sich mit der Ballvorhersage im MiroSot-Roboterfußball beschäftigt. Die Vorhersage soll zur Verbesserung der Schussfertigkeiten beitragen. Dazu wird eine stabile Ball-Tracking-Methode, eine Methode zur Ballvorhersage und ein Ballkollisionsmodell vorgestellt. Um verlässliche Daten für eine akkurate Vorhersage zu bekommen, wird eine Historie der letzten Ballpositionen geführt. So kann der Geschwindigkeitsvektor des Balles recht genau bestimmt werden. Die Reibung wird bei der Vorhersage vernachlässigt, da sie abhängig vom Spielfelduntergrund ist und die Vorhersage bei dieser Methode nicht signifikant verbessert. Durch die Dimples im in dieser Liga verwendeten Golfball ist die Vorhersage laut [Bu06] bei langsamen Bällen sehr unsicher. Kollisionen mit Wänden oder anderen Robotern werden einkalkuliert. Durch einen Geschwindigkeitsdämpfungsfaktor wird der Geschwindigkeitsvektor reduziert, wenn es zu einer Kollision kommt. Beim Aufprall auf eine Wand wird außerdem die Bewegungsrichtung des Balles durch Spiegelung der Position vor der Kollision angepasst. Ansonsten wird die Richtung des Balles aus dem Durchschnitt der letzten vier Werte in der Historie bestimmt. Eine mögliche Kurve, die der Ball rollt, wird dadurch in der Vorhersage geglättet. Auch diese Methode ist also für den Einsatz in der Standard Platform League mit ihren eher langsamen Bällen ungeeignet.

² Micro Robot Soccer Tournament [ENW05]

[HV97] stellt ein physikalisches Modell zum Tracking und zur Vorhersage verschiedener Objekte vor. Entwickelt wurde dieses Modell für den Einsatz in der Small Size League. Das Vision-System ist extern, also nicht in den Robotern verbaut, und überblickt das gesamte Spielfeld. Berechnungen erfolgen ebenfalls auf externen Computern. Sowohl für das Objekt-Tracking als auch für die Vorhersage wird der erweiterte Kalman-Filter genutzt. Die Vorhersage wird durch die mehrfache Anwendung der in der Arbeit für das jeweilige Objekt entwickelten Gleichungen auf die aktuelle Zustandseinschätzung realisiert. Um den Ball im Kalman-Filter-Framework zu modellieren, werden Gleichungen auf Basis der Newtonschen Gesetze angewandt. Dabei wird auch ein Reibungsfaktor einbezogen. Auf Kollisionen mit einer Wand wird mit der Änderung des Vorzeichens beim x- oder y-Wert des Geschwindigkeitsvektors reagiert. Bei dieser Methode der Ballvorhersage kann der komplette Pfad des Balles über beliebig viele Zeitschritte hinweg vorhergesagt werden. Die gemachten Vorhersagen sind recht genau, werden jedoch ungenauer, je weiter in die Zukunft geblickt wird. Da auch dieses Modell für die Nutzung auf glattem Untergrund mit einem Golfball entwickelt wurde, ist außerdem fraglich, ob die Genauigkeit auch auf Kunstrasen mit einem Weichschaumball gegeben ist.

In [Ho97] wird eine auf einem Musterabgleich basierende Methode zum Objekt-Tracking vorgestellt. Um die Geschwindigkeit des Musterabgleichs zwischen zwei Frames zu erhöhen, wird eine Positionsvorhersage der Objekte im Bild eingesetzt. Der Ball wird dabei nicht extra betrachtet, sondern wie jedes andere Objekt, beispielsweise andere Roboter, behandelt. Faktoren wie die Rollreibung werden also nicht beachtet. Es wird bei der Vorhersage von einer Bewegung mit konstanter Geschwindigkeit ausgegangen. Die vorhergesagte Position kann unter dieser Annahme durch das Extrapolieren der Positionswerte aus dem aktuellen und dem vorhergehenden Frame bestimmt werden. Die Ballvorhersage wird in diesem Modell nur für den jeweils nächsten Frame gemacht. Die Stoppposition kann so also nicht ermittelt werden.

3 Herausforderungen bei einer Ballvorhersage in der Standard Platform League

Laut RoboCup Standard Platform League Regeln von 2018 muss das Spielfeld aus 8mm Kunstrasen auf einer ebenen Holzfläche bestehen [Ro]. Kunstrasen gibt schnell nach und hat viele kleine Unebenheiten. Eine Ballvorhersage wird dadurch schwieriger als auf einem harten, glatten Boden.

Der offizielle Ball ist aus Weichschaum, hat einen Durchmesser von 100 mm und wiegt 44 g. [Ro] Auf der Balloberfläche befinden sich leichte Kerben für das typische Fußballmuster sowie eine produktionsbedingte Nahtstelle, was in Abbildung 1 deutlich zu erkennen ist. Dadurch wird das Rollverhalten des Balles zusätzlich beeinflusst und teilweise unberechenbar. Außerdem hat der Ball keinen durch Dimples verringerten Luftwiderstand, wie es teilweise bei den in anderen Ligen verwendeten Golfbällen der Fall ist. Der Luftwiderstand spielt bei der Vorhersage für den Ball der Standard Platform League eine wichtige Rolle. Dadurch



Abb. 1: Ball der RoboCup Standard Platform League

kann nicht von einer gleichmäßigen Beschleunigung des Balles ausgegangen werden, sondern die quadratische Abhängigkeit des Luftwiderstandes von der Geschwindigkeit muss einkalkuliert werden.

Die Roboter und meist auch der Ball bewegen sich in der Standard Platform League im Vergleich zu den Geschwindigkeiten in anderen Ligen recht langsam. Je langsamer ein Ball ist, desto mehr weicht er von einer geraden Linie ab, was wiederum eine Schwierigkeit bei der Ballvorhersage darstellt. Die Roboter sind außerdem nicht so schnell wieder am Ball wie in anderen Ligen. Daher muss die Ballposition für einen größeren Zeitraum als nur für die direkt aufeinanderfolgenden Frames vorhergesagt werden.

4 Ballvorhersage

Da die Beschaffenheit des Spielfelds und des Balles eine punktgenaue Vorhersage der Ballposition auf Distanzen, die über ein paar Zentimeter hinaus gehen, nahezu unmöglich macht, wurde ein Modell entwickelt, um zumindest den Bereich, in dem ein rollender Ball liegen bleiben wird, zu bestimmen. Als Ausgangswerte für die Vorhersage sind die aktuelle Ballposition als (x,y) Koordinaten, die Geschwindigkeit und die Rollrichtung des Balles notwendig. Diese Werte müssen also zuvor durch geeignete Verfahren vom Roboter bestimmt werden.

Für die Entwicklung des Vorhersagemodells wurde zunächst in mehreren Versuchen der Ball über das Spielfeld gerollt und jeweils die Geschwindigkeit und die Distanz zwischen Start- und Stopposition des Balles gemessen. Um eine Vorhersage der Stopposition des

Balles machen zu können, wurde anhand dieser Werte die Beschleunigung des Balles berechnet. Die Beschleunigung ist nicht gleichmäßig. Neben der Rollreibung wirkt sich auch der Luftwiderstand merklich auf die Beschleunigung des Balles aus. Der Luftwiderstand F_L steht in quadratischer Abhängigkeit zur Geschwindigkeit v .

$$F_L = c_W * A * \frac{1}{2} * \rho * v^2 \quad (3)$$

Mit Widerstandsbeiwert $c_W = 0,45$ für Kugeln, der Querschnittsfläche $A = 0,00785 \text{ m}^2$ des Balles und der Dichte ρ der Luft.

Aufgrund der quadratischen Abhängigkeit wurde ein Regressionsverfahren angewandt, um den bei jeder Geschwindigkeit konstanten Anteil der Rollreibung an der Beschleunigung zu ermitteln. Dafür wurde eine Funktion entwickelt, welche für eine gegebene Startgeschwindigkeit und einen Rollwiderstand eine Vorhersage der Distanz, die der Ball zurücklegen wird, trifft. Das Ergebnis wird jeweils mit den in den Versuchen gemessenen Werten verglichen, um so den Fehler zu bestimmen. Der Rollwiderstand wird dann über alle Versuchsdaten optimiert, sodass der quadratische Fehler minimiert wird.

Für die Vorhersage der Stoppposition des Balles wurde ein Algorithmus entwickelt, welcher den Pfad des Balles schrittweise simuliert. Für ein gegebenes Zeitintervall wird jeweils die aktuelle Geschwindigkeit v , die aktuelle Beschleunigung a und die Strecke s , welche der Ball auf Basis dieser Werte in dem gegebenen Zeitintervall t zurücklegen wird, nach Formel 4 berechnet. Dies wird durchgeführt bis s kleiner als $0,001 \text{ m}$ ist. Die aktuelle Beschleunigung setzt sich jeweils aus dem immer konstanten Rollwiderstand und dem in jedem Simulationsschritt neu berechneten Luftwiderstand zusammen.

$$s = t * \left(v + \frac{t * a}{2} \right) \quad (4)$$

Durch die Addition der so berechneten Distanzen und das Wissen, in welche Richtung der Ball rollt, kann bereits eine grobe Vorhersage getroffen werden, wo der Ball stoppen wird. Aufnahmen der Ballpfade aus den durchgeführten Versuchen zeigen allerdings, dass der Ball nicht einfach geradeaus rollt, sondern immer eine leichte Kurve macht, wie in Abbildung 2 zu sehen ist. Dabei fällt auf, dass die Kurve sich stärker ausprägt, je langsamer der Ball wird. Um dieses Verhalten nachzubilden, wurde im Algorithmus eine Änderung der Ballrichtung um einen festgelegten Winkel in jedem Simulationsschritt eingefügt. Dadurch, dass die Distanz in jedem Schritt kleiner wird, wirkt sich der Winkel automatisch mit abnehmender Ballgeschwindigkeit immer stärker aus. Ein auf diese Weise vorhergesagter Ballpfad ist in Abbildung 3 zu sehen.

Da die bekannten Werte zu Beginn der Simulation nur Geschwindigkeit und Richtung des Balles sind, ist es nicht möglich, den Winkel, um den die Richtung pro Simulationsschritt angepasst werden muss, zu bestimmen. Auch in der durch die Versuche und das Regressionsverfahren berechneten Beschleunigung durch die Rollreibung treten in der Realität Abweichungen auf. Diese werden verursacht durch die nicht immer exakt gleichen Umweltbedingungen, welche auf den Ball einwirken. Zum Beispiel können Unebenheiten im Kunstrasen eine Rolle spielen. Um den Fehler der Simulation darzustellen, nutzt das

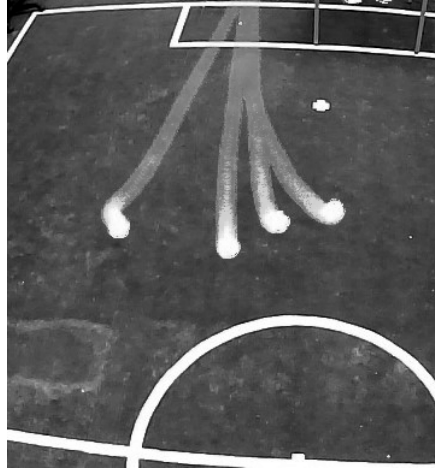


Abb. 2: Ballbewegungspfade

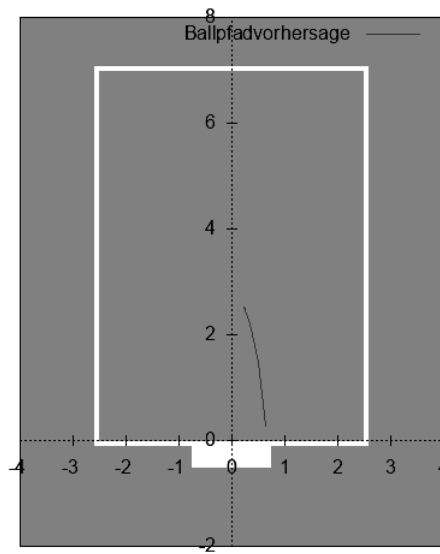


Abb. 3: Simulierter Ballbewegungspfad

Modell zufällig erzeugte, normalverteilte Werte auf der Basis der im Versuch bestimmten Werte. So wird durch mehrmaliges Durchführen des beschriebenen Algorithmus mit den verschiedenen Winkel- und Beschleunigungswerten eine Punktwolke erzeugt, in welcher mit hoher Wahrscheinlichkeit die Stopposition des Balles liegt.

Implementationsschritte für die Ballvorhersage

1. Bestimme einen zufälligen Wert für die Beschleunigung auf der Basis von Mittelwert und Standardabweichung der gemessenen Werte durch Anwendung einer Normalverteilung.
 2. Bestimme einen zufälligen Wert für den Richtungswinkel auf der Basis von festgelegtem Mittelwert und Standardabweichung durch Anwendung einer Normalverteilung.
 3. Definiere ein Zeitintervall für die Länge der Simulationsschritte.
 4. Tue Folgendes, solange die berechnete Distanz größer als $0,001m$ ist:
 - a) Berechne den Luftwiderstand bei der aktuellen Geschwindigkeit und kalkuliere ihn bei der Beschleunigung ein.
 - b) Berechne die Distanz, die der Ball bei aktueller Geschwindigkeit mit aktueller Beschleunigung im gegebenen Zeitintervall zurücklegen wird.
 - c) Bringe den aktuellen Geschwindigkeitsvektor des Balles auf die berechnete Länge und rotiere ihn um den gegebenen Winkel.
 - d) Addiere den Ergebnisvektor zur aktuellen Position des Balles.
 - e) Setze die neue Beschleunigung, Geschwindigkeit und Ballposition für den nächsten Schleifendurchlauf.
-

5 Auswertung

Um die Erfolgsquote des beschriebenen Ballvorhersagemodells zu prüfen, wurde die in den Versuchen bestimmten Ballstopppositionen mit dem durch den Algorithmus vorhergesagten Bereich verglichen. Die Resultate von 20 Versuchen sind in den Abbildungen 4 bis 7 dargestellt. Die roten Punkte sind dabei die gemessenen Werte, die kleinen Kreuze bilden die vorhergesagte Punktwolke. Das + vor dem Tor markiert den Startpunkt des Balles. In den dargestellten Versuchen wurden jeweils fünf Bälle in vier unterschiedlichen Geschwindigkeiten über das Spielfeld gerollt. Nach diesen insgesamt 20 Versuchen befindet sich nur ein Ball deutlich außerhalb der vorhergesagten Koordinaten (siehe Abbildung 6). Es wurden für jede der vier Geschwindigkeiten noch weitere 20 Versuche gemacht, sodass die Ballstoppposition insgesamt 100 Mal gemessen wurde. Dabei lagen 16 Bälle außerhalb des vorhergesagten Bereiches. Die Erfolgsquote der vorgestellten Methode liegt also bei 84 %, wenn der Ball frei rollen kann. Hindernisse werden nicht beachtet. Dabei ist der vorhergesagte Bereich nicht größer als nötig. Bei den schnelleren Bällen, welche eine relativ weite Strecke rollen, gibt es größere Abweichungen bei der Ballstoppposition als bei den langsameren Bällen. Dies spiegelt sich im Vorhersagebereich wieder. Dieser ist bei einer Ball-Start-Stop-Distanz von $3,30m$ etwa $1m^2$ groß, bei einer Distanz von etwa $1,20m$ nur rund $50cm^2$. Dies ist beides völlig ausreichend, um dem Roboter eine Richtung vorzugeben,

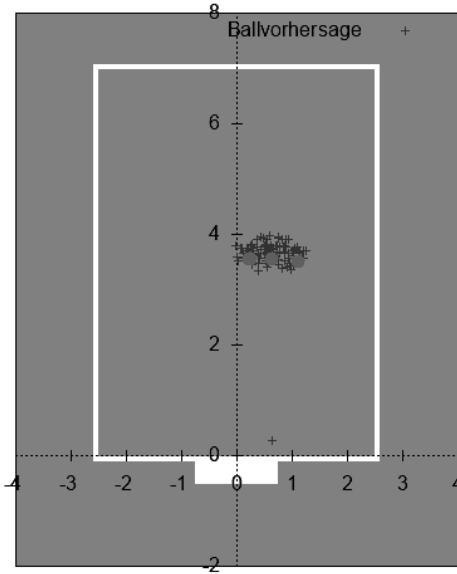


Abb. 4: Vorhergesagte und reale Stopppositionen vom Ball mit einer Startgeschwindigkeit von $1,5 \text{ m s}^{-1}$

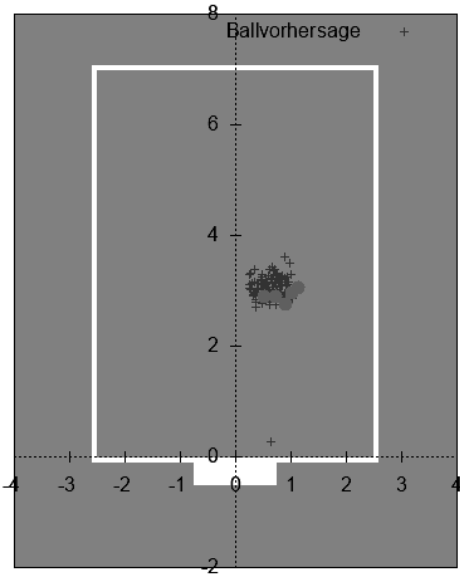


Abb. 5: Vorhergesagte und reale Stopppositionen vom Ball mit einer Startgeschwindigkeit von $1,37 \text{ m s}^{-1}$

in die er sich bewegen muss. Auch eine gute Abschätzung, ob der Ball ins Tor gehen wird, ist damit möglich.

Die hier vorgestellte Methode der Ballvorhersage ist auf unregelmäßigem Boden mit einem ebenfalls leicht unregelmäßigem Ball besser geeignet als die bereits bekannten Methoden aus der RoboCup Small Size League. Dort besteht das Spielfeld aus einer Filzmatte mit flachem, hartem Untergrund. Als Ball wird in dieser Liga ein Golfball verwendet. [Ru] Die Dimples auf der Oberfläche eines Golfballes verringern den Luftwiderstand des Balles deutlich [Sh14]. Das Rollverhalten des Balles in der Small Size League ist folglich gleichmäßiger und vorhersehbarer als in der Standard Platform League.

Da die Roboter der Standard Platform League viel langsamer sind als die der Small Size League oder die beim MiroSot, besteht das primäre Ziel nicht darin, den Ball auf seinem Weg abzufangen, wie in den meisten bekannten Methoden. Dies wäre aufgrund der fehlenden Geschwindigkeit sowieso nur möglich, wenn der Roboter sich zufällig sehr nah am Ballpfad befindet. Somit ist es nicht notwendig, den gesamten Ballverlauf perfekt vorherzusagen, sondern nur die Stoppposition des Balles. So kann der Roboter direkt dorthin laufen und geht nicht immer in die Richtung, in der er den Ball sieht. Dadurch wird das Laufen einer Kurve und so der Verlust wertvoller Zeit verhindert. Auch für die Abschätzung, ob der Ball ins Tor gehen wird, ist nicht der gesamte Pfad, sondern nur die voraussichtliche Stoppposition des Balles entscheidend.

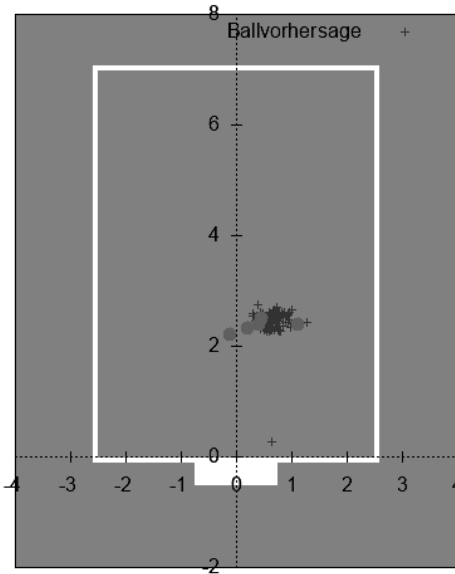


Abb. 6: Vorhergesagte und reale Stopppositionen vom Ball mit einer Startgeschwindigkeit von $1,2 \text{ m s}^{-1}$

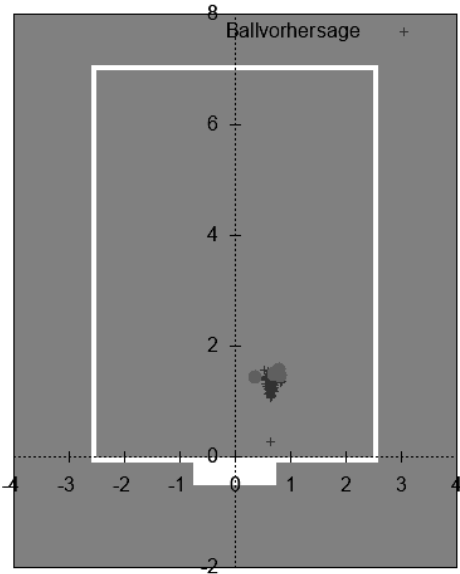


Abb. 7: Vorhergesagte und reale Stopppositionen vom Ball mit einer Startgeschwindigkeit von $0,82 \text{ m s}^{-1}$

6 Fazit und Ausblick

Das entwickelte Modell zur Ballvorhersage ist sehr gut für den Einsatz in langsamen Roboterfußballspielen mit einem schwierig einzuschätzendem Ballrollverhalten geeignet. Da das Modell einen Wert für die Rollreibung benötigt, ist es sehr speziell auf die Nutzung auf einem bestimmten Boden beschränkt. Sobald es auf einem anderen Boden angewandt werden soll, muss die Rollreibung auf diesem Boden neu bestimmt werden, was mit etwas Aufwand verbunden ist.

Eine wichtige Voraussetzung für den Einsatz der vorgestellten Vorhersagemethode ist eine sehr gute Ballerkennung. Sowohl die Ballposition als auch der Geschwindigkeitsvektor müssen möglichst exakt in die Berechnung hinein gegeben werden, um eine korrekte Vorhersage zu erhalten.

Dadurch, dass der Luftwiderstand mit einkalkuliert wird, ist die Vorhersage sowohl bei langsamen als auch bei schnellen Bällen korrekt. Der Zeitraum oder die Distanz der Vorhersage haben auf die Korrektheit keinen negativen Einfluss. Lediglich der vorausgesagte Bereich wird mit zunehmender Distanz etwas größer.

Um das Modell flexibler zu gestalten, sollte in Zukunft eine Methode gefunden werden, um die Rollreibung auf verschiedenen Böden mit relativ geringem Aufwand zu bestimmen. Durch die exaktere Bestimmung von Rollreibung und Luftwiderstand kann eventuell auch der Vorhersagebereich noch etwas eingeschränkt und somit genauer werden.

Literaturverzeichnis

- [Ab11] Abreu, P.; Costa, I.; Casteldo, D.; Reis, L. P.; Garganta, J.: Human vs. Robotic Soccer: How Far Are They? A Statistical Comparison. In: RoboCup 2010 Robot Soccer World Cup XIV. Springer, Berlin, Heidelberg, S. 242–253, 2011.
- [Bu06] Buth, M. D.: Ball-handling motion control for soccer playing mini-robots. Masterarbeit, University of Twente, EEMCS: Electrical Engineering, Mathematics and Computer Science, 2006.
- [ENW05] Egly, U.; Novak, G.; Weber, D.: Decision making for mirosot soccer playing robots. 1st CLAWAR/EURON/IARPWorkshop on Robots in Entertainment, Leisure and Hobby, 2005.
- [Ho97] Hong, C. S.; Chun, S. M.; Lee, J. S.; Hong, K. S.: A vision-guided object tracking and prediction algorithm for soccer robots. In: Proceedings of International Conference on Robotics and Automation. S. 346–351 vol.1, April 1997.
- [HV97] Han, K.; Veloso, M.: Physical Model Based Multi-objects Tracking and Prediction in RoboSoccer. In: Working Notes of the AAAI 1997 Fall Symposium. AAAI. MIT Press, 1997.
- [Ki98] Kitano, H.; Tambe, M.; Stone, P.; Veloso, M.; Coradeschi, S.; Osawa, E.; Matsubara, H.; Noda, I.; Asada, M.: The RoboCup synthetic agent challenge 97. In (Kitano, H., Hrsg.): RoboCup-97: Robot Soccer World Cup I. Springer Berlin Heidelberg, Berlin, Heidelberg, S. 62–73, 1998.
- [Ko07] Kobayashi, H.; Osaki, T.; Williams, E.; Ishino, A.; Shinohara, A.: Autonomous Learning of Ball Trapping in the Four-Legged Robot League. In: RoboCup 2006 Robot soccer world cup X. Springer, Berlin, Heidelberg, S. 86–97, 2007.
- [Li02] Liu, H.; Zha, H.; Chen, K.; Wang, P.: A New Real-time Collision Prediction Model for Soccer Robots. National Lab. on Machine Perception, Peking University, 100871, China, 2002.
- [Ri08] Ribeiro, F.; Moutinho, I.; Pereira, N.; Oliveira, F.; Fernandes, J.; Peixoto, N.; Salgado, A.: High Accuracy Navigation in Unknown Environment Using Adaptive Control. In: RoboCup 2007 Robot Soccer World Cup XI. Springer, Berlin, Heidelberg, S. 312–319, 2008.
- [Ro] RoboCup Technical Committee: RoboCup Standard Platform League (NAO) Rule Book. <https://spl.robocup.org/wp-content/uploads/downloads/Rule2018.pdf>. Stand: 22.03.2019.
- [Ro12] RoboCup 2011 Robot Soccer World Cup XV. Springer, Berlin, Heidelberg, 2012.
- [Ru] Rules of the RoboCup Small Size League 2019. <https://robocup-ssl.github.io/ssl-rules/sslrules.pdf>. Stand: 23.03.2019.
- [Sh14] Shonibare, O. Y.: Effect of dimples on golf balls and cars: A Review. 12 2014.
- [Su14] Sudin, M. N.; Abdullah, S. N. H. S.; Nasrudin, M. F.; Sahran, S.: Trigonometry Technique for Ball Prediction in Robot Soccer. In (Kim, J.; Matson, E. T. ; Myung, H.; Xu, P.; Karray, F., Hrsg.): Robot Intelligence Technology and Applications 2: Results from the 2nd International Conference on Robot Intelligence Technology and Applications. Springer International Publishing, Cham, S. 753–762, 2014.

Algorithmik

Datenbasierter Vergleich von statistischen Tests mithilfe von Simulationen

Auswahl eines Verfahrens für das A/B-Testing

Felix J. M. Welter¹

Abstract: Die Wahl des richtigen statistischen Verfahrens ist für einen A/B-Test essentiell. Bei Nutzung eines mächtigen Tests kann ein Experiment schneller zu Erkenntnissen führen und es können früher datengetriebene Entscheidungen getroffen werden. In dieser Untersuchung wurden der Permutationstest und der Wilcoxon-Rangsummentest bezüglich Mächtigkeit und Alphanobustheit verglichen. Die Vorgehensweise kann zudem abstrahiert und auf andere Tests oder Anwendungen übertragen werden.

Keywords: Statistische Tests, A/B-Test, Permutationstest, Wilcoxon-Rangsummentest, Simulation, Python

1 Einleitung

Das Wachstum der Branche E-Commerce seit der Jahrtausendwende ist enorm. [HD18] Auf die steigenden Anforderungen reagieren Onlinehändler mit unterschiedlichsten Produktentwicklungen und neuen Features für ihre Webseiten. Ein klassisches Beispiel dafür ist das Empfehlungssystem von Amazon. [LSY03] Auch die aggregierten Kundenbewertungen, wie sie auf otto.de eingesetzt werden, gehören dazu. Jedoch sind die Auswirkungen einer neuen Entwicklung nur selten präzise vorherzusehen. Doch diese genaue Quantifizierung ist notwendig, um Ressourcen sinnvoll und mit maximalem Nutzen einzusetzen. Um dies zu ermöglichen, wird im E-Commerce eine Methode namens A/B-Testing verwendet. Die zu prüfende Änderung wird dabei nur einem Teil der Webseitennutzer angezeigt. Wenn bei allen Nutzern relevante Kennzahlen erfasst werden, häufig handelt es sich dabei um Bestellungen oder Umsatz, dann können diese im Nachhinein verglichen werden. Der Vergleich wird dabei zwischen allen Nutzern, welche das neue Feature gesehen haben, und den restlichen Nutzern gezogen. [KHS07]

Ob eine gemessene Differenz der Kennzahlen einem tatsächlichen Unterschied zwischen den Varianten entstammt, oder ob es sich um eine natürliche Schwankung handelt, kann durch einen statistischen Test geprüft werden. Es ist ersichtlich, dass ein

¹ Nordakademie, Köllner Chaussee 11, 25337 Elmshorn, felix.welter.i15c@nordakademie.org

A/B-Test wünschenswerterweise bereits nach kurzer Zeit zu einem Ergebnis führt, da positive Änderungen für alle Nutzer eingeführt und negative Änderungen früher wieder deaktiviert werden können. Daraus ergibt sich, dass der eingesetzte statistische Test mit möglichst wenig Webseitenbesuchern zu einer Aussage kommen muss. Diese Untersuchung vergleicht den Permutationstest mit dem Wilcoxon-Rangsummentest basierend auf dieser Fragestellung. Das Vorgehen kann sehr gut abstrahiert und in anderen Kontexten genutzt werden.

2 Grundlagen

Die in dieser Untersuchung genutzten Daten sind durch Bestellungen von Nutzern auf otto.de zustande gekommen. Betrachtet wird dabei der monetäre Wert der bestellten Artikel. Durch eine interne Vorprozessierung der Daten liegen sowohl für die Testgruppe, welche die Änderung ausgespielt bekommt, wie auch für die Kontrollgruppe, welche die unveränderte Webseite sieht, hundert Messwerte vor. Diese können nun mit einem statistischen Test verglichen werden.

Diese generierten Daten folgen nicht der Normalverteilung, weshalb ein nicht-parametrischer Test genutzt werden sollte. In so einem Fall kommt häufig der Wilcoxon-Rangsummentest zur Anwendung. Dies entspricht auch dem Vorgehen bei der Firma Otto (GmbH & Co KG). Die Funktionsweise ist wie folgt. Die Messwerte beider Stichproben werden übergreifend sortiert und durch ihren Rang ersetzt. Es wird die Rangsumme R der ersten Stichprobe errechnet. Für diese Rangsumme existieren Erwartungswert μ_R und Standardabweichung σ_R .

$$\mu_R = \frac{n_1(n_1 + n_2 + 1)}{2}$$
$$\sigma_R = \sqrt{\frac{n_1 n_2 (n_1 + n_2 + 1)}{12}}$$

Die errechnete Rangsumme R kann dann in eine z -Statistik überführt und entsprechend der Normalverteilung interpretiert werden. Um den Vergleich der beiden Tests zu vereinfachen, wird daraus noch der p -Wert errechnet. Es handelt sich hierbei um ein approximatives Verfahren, welches erst ab zehn Messwerten je Gruppe zuverlässig funktioniert. Dies ist im vorliegenden Anwendungsfall jedoch gegeben. [Wi45], [Ka86]

Gegenüber dem Wilcoxon-Rangsummentest soll untersucht werden, ob der Permutationstest eine Alternative darstellt. Dabei handelt es sich um ein Resamplingverfahren. Abhängig von Datenmenge und Testszenario existieren unterschiedliche Versionen. Die für diese Untersuchung genutzte wird wie folgt durchgeführt. Im ersten Schritt wird die Differenz der Mittelwerte beider Stichproben errechnet. In Schritt zwei wird die Gruppenzugehörigkeit der Messwerte neu zugewiesen, solange bis beide Gruppen ihre ursprüngliche Größe erreicht haben. Dann wird wieder die Differenz der Mittelwerte kalkuliert. Schritt zwei wird mehrmals

wiederholt, woraufhin die kalkulierten Differenzen die Permutationsverteilung bilden. Der p-Wert des Tests ist dann das Verhältnis zwischen der Anzahl von Permutationsdifferenzen, die größer sind als die real gemessene Differenz und die Anzahl der errechneten Permutationen. [Co13] Wie häufig die Gruppenzugehörigkeit für diesen Test permutiert wird, kann je nach Anwendungsfall variieren. Für Simulationen ist jedoch eine Zahl von 1000 als sinnvoll zu erachten. [Ma07]

Für die Untersuchung ist weiterhin ein statistisches Konzept namens Mächtigkeit relevant, häufig auch als Power, Teststärke oder Trennschärfe bezeichnet. Die Mächtigkeit gibt die Wahrscheinlichkeit dafür an, einen Effekt zu erkennen, wenn ein Effekt vorliegt. Im Kontext eines A/B-Tests ist es die Fähigkeit einen Unterschied zwischen den Varianten einer Webseite auszumachen, wenn diese tatsächlich zu verschiedenen Ergebnissen der Kennzahlen führen. Mächtigkeit ist damit das Komplement zum Fehler zweiter Art. [Ka86] Die Mächtigkeit steigt, wenn für den Test mehr Messwerte zur Verfügung stehen. Dies ist nachvollziehbar, da mit mehr Messwerten die Menge an Informationen steigt und die Aussage des Tests sicherer wird. [Co92] Wie zuvor erläutert, soll ein Test jedoch trotz geringen Stichprobenumfangs möglichst mächtig sein, sodass eine kurze Laufzeit eines A/B-Tests möglich wird.

3 Zielsetzung

Da der Wilcoxon-Rangsummentest derzeit in dem Unternehmen Otto (GmbH Co & KG) eingesetzt wird, ist der Permutationstest als Herausforderer zu betrachten. Es gilt also zu zeigen, dass der Permutationstest besser zur Auswertung von A/B-Tests geeignet ist, sprich, dass er kürzere Laufzeiten ermöglicht. Diese Vermutung liegt darin begründet, dass der Wilcoxon-Rangsummentest bei der Transformation der Messwerte in Ränge Informationen verliert [GS98], während der Permutationstest gemeinhin als sehr mächtiges Verfahren angesehen wird. [Go07], [Co13] Die Frage soll anhand zweier Kriterien untersucht werden.

Das erste Kriterium ist die Mächtigkeit der Tests. Der Permutationstest sollte mächtiger sein als der Wilcoxon-Rangsummentest. Häufig werden Tests über die asymptotische relative Effizienz verglichen. Dieser Wert gibt an, wie viele Messwerte die beiden statistischen Test relativ zueinander benötigen, um eine gewünschte Mächtigkeit bei gesetztem Alphaniveau und bekanntem Effekt zu erreichen. [Ni11] Für diese Untersuchung wird jedoch eine reale Datenbasis genutzt, weshalb die Stichprobengröße gesetzt ist. Der Vergleich wird deshalb über die Mächtigkeit durchgeführt. Dies ist ebenso gut möglich, da jeder Einflussfaktor (Mächtigkeit, Alphaniveau, Effekt und Stichprobengröße) als Funktion der anderen drei dargestellt werden kann. [Co92] Im Zuge der Simulation wird deshalb der Anteil der signifikanten Ergebnisse an allen Simulationsläufen errechnet und die Anteile der beiden Tests dann mit einem Test auf Gleichheit von Proportionen verglichen.

Ein mächtiger Test ist nicht nutzbar, wenn er deutlich häufiger zu Alphafehlern führt, als durch das Alphaniveau vorgegeben ist. Aus diesem Grund wird die Robustheit gegenüber dem Alphafehler als zweites Kriterium angeführt. Diese Robustheit wird für das Signifikanzniveau α und eine tatsächliche Alphafehlerwahrscheinlichkeit α^* wie folgt beschrieben

$$r = \frac{|\alpha - \alpha^*|}{\alpha}$$

und gibt somit die relative Abweichung vom gesetzten Alphaniveau an. [Bü91] Als strenge Grenze für r wird 0,1 angesetzt. [Br78] Das zweite Entscheidungskriterium wird somit erfüllt, wenn die Robustheit des Permutationstests diese Grenze nicht überschreitet.

4 Simulation

Die Datenbasis für die Simulation bilden fünf verschiedene Zeiträume, welche möglichst unterschiedliche Charakteristika für die Kennzahl Bestellwert aufzeigen. Dadurch können Erkenntnisse der Untersuchung verallgemeinert werden. Im weiteren Verlauf wird auf die Betrachtung der einzelnen Zeiträume jedoch verzichtet, da sehr ähnliche Simulationsverläufe beobachtet wurden und die abgeleiteten Aussagen dieselben sind.

Für jeden Zeitraum wurden verschiedene Effekte untersucht. Dieser Effekt wird im E-Commerce meist als Prozentsatz angegeben und als Uplift bezeichnet. Für diese Untersuchung wurden zur Feststellung der Mächtigkeit die Uplifts 1,0%, 1,2%, 1,4% und 1,6% simuliert. Zusätzlich wurde der Uplift 0% berücksichtigt, um ein Urteil über die Alpharobustheit fällen zu können. Dabei wurde der Effekt additiv berechnet. Ob diese additive Berechnung die Realität abbildet, ist zwar infrage zu stellen, jedoch handelt es sich dabei um das gemeinhin anerkannte Vorgehen zur Analyse der Mächtigkeit. [Fa07], [Co92], [DM00]

Die beiden statistischen Tests wurden in Python mithilfe der Pakete *numpy* [Ol15], *pandas* [We10], *scipy* [JO01] und *numba* [LPS15] implementiert. Auch die Simulation wurde unter Zuhilfenahme dieser Pakete realisiert. Durchgeführt wurden pro Zeitraum und Uplift 6.000 Einzelsimulationen, in welchen die Messwerte zufällig der Test- oder Kontrollgruppe zugewiesen wurden, der Effekt zu der Kontrollgruppe addiert wurde und zuletzt der Permutationstest und der Wilcoxon-Rangsummentest auf die Daten angewendet wurden. Es wurden jeweils die einseitigen p-Werte aufgezeichnet und in einer Struktur, wie in Tab. 1 dargestellt, zur weiteren Analyse persistiert.

Uplift	Permutationstest	Wilcoxon-Rangsummentest
0%	0,15	0,18
0%	0,59	0,61
...
1,6%	0,01	0,02
1,6%	0,02	0,02

Tab. 1: Struktur der Simulationsergebnisse

Da für jeden einzelnen Uplift 6.000 Einzelsimulationen durchgeführt wurden, besteht eine solche Ergebnistabelle aus 30.000 Zeilen. Pro Test und Uplift wurde nun die Anzahl der Ergebnisse errechnet, die bei einem Alphaniveau von 5% signifikant sind. Als Liniendiagramm dargestellt ergibt sich Abb. 1.

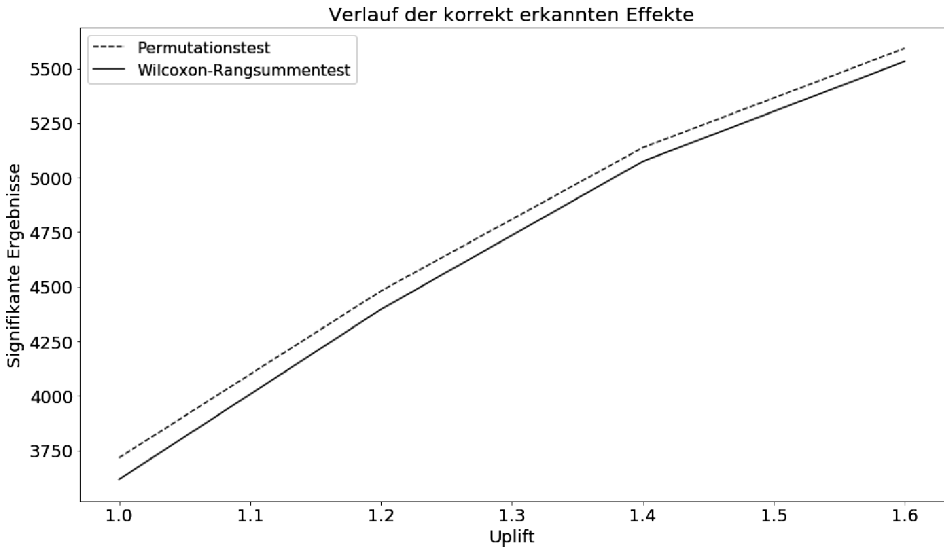


Abb. 1: Verlauf der korrekt erkannten Effekte

Da die Linie des Permutationstests oberhalb der des Wilcoxon-Rangsummentests verläuft, ist erkennbar, dass der erstgenannte scheinbar mächtiger ist. Mit einem Test auf Gleichheit von Proportionen lässt sich diese Beobachtung prüfen. Aus diesem Grund wurde pro Zeitraum und Uplift ein solcher Test durchgeführt. Der Anteil der signifikanten Ergebnisse sowie die p-Werte des Proportionentests sind in Tab. 2 dargestellt.

Datenbasis	Uplift (%)	Permutations- test	Wilcoxon- Rangsummen- test	Proportionen- test
Zeitraum 1	1,0	0,619	0,603	0,032
	1,2	0,747	0,733	0,04
	1,4	0,856	0,846	0,053
	1,6	0,932	0,922	0,018
Zeitraum 2	1,0	0,745	0,722	0,003
	1,2	0,871	0,856	0,009
	1,4	0,941	0,931	0,014
	1,6	0,976	0,972	0,126
Zeitraum 3	1,0	0,781	0,764	0,016
	1,2	0,897	0,889	0,074
	1,4	0,961	0,955	0,047
	1,6	0,986	0,984	0,129
Zeitraum 4	1,0	0,674	0,662	0,07
	1,2	0,810	0,793	0,008
	1,4	0,896	0,880	0,002
	1,6	0,955	0,946	0,01
Zeitraum 5	1,0	0,776	0,757	0,007
	1,2	0,896	0,884	0,018
	1,4	0,958	0,952	0,086
	1,6	0,986	0,982	0,029

Tab. 2: Anteil signifikanter Ergebnisse und p-Werte des Proportionentests

Aus der Tabelle geht hervor, dass der Permutationstest in vielen Fällen statistisch signifikant mächtiger ist als der Wilcoxon-Rangsummentest. Mit dieser Erkenntnis und der visuellen Bestätigung durch Abb. 1 ist das erste Entscheidungskriterium erfüllt.

Ob der Permutationstest auch alfarobust ist, wie vom zweiten Entscheidungskriterium gefordert, lässt sich mittels Tab. 3 erkennen. Dort ist die tatsächlich realisierte Häufigkeit des Fehlers erster Art dargestellt, sowie die daraus abgeleitete Alfarobustheit. Alle Werte liegen unterhalb 0,1 und genügen somit der strengen Grenze für die Alfarobustheit. Das zweite Kriterium ist somit ebenfalls erfüllt.

Datenbasis	Anteil Fehler erster Art		Alpharobustheit	
	Permutations-test	Wilcoxon-Rangsummen-test	Permutations-test	Wilcoxon-Rangsummen-test
Zeitraum 1	0,049	0,049	0,02	0,02
Zeitraum 2	0,051	0,051	0,02	0,02
Zeitraum 3	0,050	0,050	0	0
Zeitraum 4	0,052	0,053	0,04	0,06
Zeitraum 5	0,054	0,053	0,08	0,06

Tab. 3: Anteil Fehler erster Art und Alpharobustheit

5 Fazit

Die Annahme, dass der Permutationstest mächtiger ist als der Wilcoxon-Rangsummentest, kann basierend auf den gewonnenen Erkenntnissen bestätigt werden. Somit ergibt sich auch die Empfehlung, ihn künftig an Stelle des Wilcoxon-Rangsummentests einzusetzen. Aus A/B-Tests können somit schneller datenbasierte Erkenntnisse abgeleitet und positive Features früher der gesamten Nutzerbasis zur Verfügung gestellt werden. Ebenso ist es möglich, negative Änderungen frühzeitig aus dem Betrieb des Onlineshops zu entfernen. Direkt daraus abgeleitet ergibt sich ein monetärer Nutzen.

Im Zuge dieser Fragestellung wurde ebenfalls der t-Test mit dem beschriebenen Vorgehen untersucht. Bei dem Test handelt es sich um einen etablierten Standard in der Statistik. Er basiert jedoch auf der Annahme, dass die Daten normalverteilt sind. [St08] Obwohl diese Annahme bei den untersuchten Daten von otto.de stark verletzt ist, konnte der t-Test Effekte mindestens so gut wie der Permutationstest erkennen. In einigen Fällen zeigte sich sogar eine kleine Abweichung zugunsten des t-Tests. Ob es sich dabei um ein Zeugnis der hohen Robustheit des t-Tests handelt, welche im häufig nachgesagt wird [Bü91], [Ka86], [HD87], oder ob dies möglicherweise durch die additive Effektrechnung zustande kam, ist eine Frage für künftige Untersuchungen.

Weiterhin bieten sich für künftige Untersuchungen die folgenden Fragestellungen an. Zum einen sollten noch weitere Zeiträume untersucht werden. Während bereits fünf verschiedene zeitliche Abschnitte berücksichtigt wurden, kann die allgemeine Gültigkeit der Erkenntnisse damit gefestigt werden. Zweitens kann die Untersuchung für andere Kennzahlen durchgeführt werden. Während in dieser Untersuchung der monetäre Bestellwert der abgeschickten Warenkörbe berücksichtigt wurde, können auch Bestellmengen oder Retouren als Datenbasis genutzt werden. Ob dies sinnvoll ist, hängt

jedoch von den geschäftlichen Zielen ab. Zuletzt soll an dieser Stelle zudem darauf hingewiesen werden, dass die Untersuchung weiterer statistischer Verfahren ebenfalls eine Option für das weitere Vorgehen darstellt.

6 Weiterentwicklung und Operationalisierung

Um auf den Erkenntnissen dieser Untersuchung aufbauen zu können, werden in diesem Abschnitt einige Ansätze aus der Literatur vorgestellt, welche zu einer besseren Anwendbarkeit oder Verlässlichkeit des Permutationstests beitragen sollen. Während der Durchführung des Permutationstests wird die Gruppenzugehörigkeit wiederholt neu zugewiesen. Der Rechenaufwand dafür ist auf modernen Rechnern zu bewältigen, jedoch kann ein Test besonders bei größeren Stichproben durchaus eine Rechenzeit erreichen, welche einen akzeptablen Rahmen sprengt. Zur Reduzierung der benötigten Zeit schlagen Boos und Zhang vor, nach einer kleineren Anzahl von Permutationen mittels Extrapolation auf die restlichen zu schließen. [BZ00] Eine andere Methode kalkuliert nach einer gewissen Anzahl von Permutationen die Wahrscheinlichkeit für das Eintreten eines signifikanten Ergebnisses, sollte noch keines aufgetreten sein und beendet den Test, wenn diese Wahrscheinlichkeit gering ist. Diese Prüfung wird auch von der anderen Seite vollzogen. Der Test wird also bei einem bereits hochsignifikanten Ergebnis beendet, ohne weitere Permutationen zu berechnen. Auch dieses Verfahren trägt zur Reduzierung der Rechenzeit bei. [Kn09]

Häufig soll bei einem Experiment nicht nur ein Effekt nachgewiesen, sondern auch eine Aussage über die Stärke des Effekts gemacht werden. Dabei kommen Konfidenzintervalle zum Einsatz, welche den Zahlenwert der Teststatistik mit einer bestimmten Wahrscheinlichkeit innerhalb gewisser Grenzen einordnen. Im Gegensatz zu einigen anderen Verfahren kann aus dem Permutationstest nicht direkt ein Konfidenzintervall abgeleitet werden. Heiler und Wechselberger haben ein Verfahren vorgestellt, welches die Grenzen des Konfidenzintervalls im Vorfeld schätzt und diese Schätzung dann iterativ verbessert. [HW69]

Weitere Ansätze versuchen die Mächtigkeit des Permutationstest durch Umgehen von doppelten Gruppenkonstellationen zu erhöhen, [Op03] berechnen Konfidenzintervalle für den p-Wert des Tests, welcher aufgrund der zufälligen Ziehungen schwankt [Er04] oder erweitern die Einsatzmöglichkeiten auf andere Verfahren wie die Regression. [Pi37], [Ma06]

Beim operativen Einsatz des Permutationstests muss bedacht werden, dass dieser, wie bereits erwähnt, nicht deterministisch ist. Um sicherzustellen, dass unabhängige Auswertungen dennoch zu dem gleichen Ergebnis kommen, kann vor der Auswertung ein sogenannter Seed, ein Startwert für einen Zufallsgenerator, festgelegt werden. Die Pseudozufallszahlen des Auswertungsprogramms werden somit reproduzierbar, ebenso wie das Ergebnis des Permutationstests. [PM88]

Bei der Einführung sollte bedacht werden, dass ein Parallelbetrieb beider Testverfahren zu einer praktischen Vergleichbarkeit sinnvoll sein kann, wenn dies nicht zu viele Ressourcen bindet. Unterstützt wird der Permutationstest jedoch bereits von vielen Statistikanwendungen, wie SPSS [Ha98], STATA [Ka07] und R [FS10]. Die Implementierung in beliebigen Programmiersprachen ist ebenfalls umstandslos möglich.

7 Danksagung

Mein Dank gilt dem Unternehmen Otto (GmbH & Co KG), welches mir ein großartiges duales Studium ermöglicht hat, ebenso wie den Testanalysten Antje Krumnack und Rene Gilster, welche mich bei dieser Untersuchung unterstützt haben.

Literaturverzeichnis

- [Br78] Bradley, J. V.: Robustness? In *British Journal of Mathematical and Statistical Psychology*, 1978, 31; S. 144–152.
- [Bü91] Büning, H.: *Robuste und adaptive Tests*, 1991.
- [BZ00] Boos, D. D.; Zhang, J.: Monte Carlo Evaluation of Resampling-Based Hypothesis Tests. In *Journal of the American Statistical Association*, 2000, 95; S. 486.
- [Co13] Collingridge, D. S.: A Primer on Quantitized Data Analysis and Permutation Testing. In *Journal of Mixed Methods Research*, 2013, 7; S. 81–97.
- [Co92] Cohen, J.: A power primer. In *Psychological bulletin*, 1992, 112; S. 155–159.
- [DM00] Davidson, R.; MacKinnon, J.: Bootstrap tests: how many bootstraps? In *Econometric Reviews*, 2000, 19; S. 55–68.
- [Er04] Ernst, M. D.: Permutation Methods: A Basis for Exact Inference. In *Statistical Science*, 2004, 19; S. 676–685.
- [Fa07] Faul, F. et al.: G*Power 3: A flexible statistical power analysis program for the social, behavioral, and biomedical sciences. In *Behavior Research Methods*, 2007, 39; S. 175–191.
- [FS10] Fay, M. P.; Shaw, P. A.: Exact and Asymptotic Weighted Logrank Tests for Interval Censored Data: The intervalR Package. In *Journal of Statistical Software*, 2010, 36.
- [Go07] Govindarajulu, Z.: *Nonparametric inference*, 2007.
- [GS98] Gebhard, J.; Schmitz, N.: Permutation tests — A revival?! In *Statistical Papers*, 1998, 39; S. 75–85.
- [Ha98] Hayes, A. F.: SPSS procedures for approximate randomization tests. In *Behavior Research Methods, Instruments, & Computers*, 1998, 30; S. 536–543.

- [HD18] HDE: Umsatz durch E-Commerce (B2C) in Deutschland in den Jahren 1999 bis 2017 sowie eine Prognose für 2018 (in Milliarden Euro). <https://de.statista.com/statistik/daten/studie/3979/umfrage/e-commerce-umsatz-in-deutschland-seit-1999/>, 14.12.2018.
- [HD87] Heeren, T.; D'Agostino, R.: Robustness of the two independent samples t-test when applied to ordinal scaled data. In *Statistics in Medicine*, 1987, 6; S. 79–90.
- [HW69] Heiler, S.; Weichselberger, K.: Über den Permutationstest und ein daraus ableitbares Konfidenzintervall. In *Metrika*, 1969, 14; S. 232–248.
- [JO01] Jones, E.; Oliphant, T.; Peterson, P.; others: SciPy: Open source scientific tools for Python. <http://www.scipy.org/>, 15.01.2019.
- [Ka07] Kaiser, J.: An exact and a Monte Carlo proposal to the Fisher–Pitman permutation tests for paired replicates and for independent samples. In *Stata Journal*, 2007, 7; S. 402–412.
- [Ka86] Kachigan, S. K.: *Statistical analysis. An interdisciplinary introduction to univariate & multivariate methods*. Radius Press, New York, 1986.
- [KHS07] Kohavi, R.; Henne, R. M.; Sommerfield, D.: Practical guide to controlled experiments on the web. In (Berkhin, P.; Caruana, R.; Wu, X. Hrsg.): *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '07*. ACM Press, New York, New York, USA, 2007; S. 959.
- [Kn09] Knijnenburg, T. A. et al.: Fewer permutations, more accurate P-values. In *Bioinformatics (Oxford, England)*, 2009, 25; i161–8.
- [LPS15] Lam, S. K.; Pitrou, A.; Seibert, S.: Numba. In (Finkel, H. Hrsg.): *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC - LLVM '15*. ACM Press, New York, New York, USA, 2015; S. 1–6.
- [LSY03] Linden, G.; Smith, B.; York, J.: Amazon.com recommendations: item-to-item collaborative filtering. In *IEEE Internet Computing*, 2003, 7; S. 76–80.
- [Ma06] Manly, B. F. J.: *Randomization, Bootstrap and Monte Carlo Methods in Biology*. CRC Press, Boca Raton, FL, 2006.
- [Ma07] Marozzi, M.: Some remarks about the number of permutations one should consider to perform a permutation test, 2007.
- [Ni11] Nikitin, Y.: Asymptotic Relative Efficiency in Testing. In (Lovric, M. Hrsg.): *International Encyclopedia of Statistical Science*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011; S. 72–75.
- [Ol15] Oliphant, T. E.: *Guide to NumPy*. Continuum Press, Austin, Tex., op. 2015.
- [Op03] Opdyke, J. D.: Fast Permutation Tests that Maximize Power Under Conventional Monte Carlo Sampling for Pairwise and Multiple Comparisons. In *Journal of Modern Applied Statistical Methods*, 2003, 2; S. 27–49.

- [Pi37] Pitman, E. J. G.: Significance Tests Which May be Applied to Samples from any Populations. II. The Correlation Coefficient Test. In Supplement to the Journal of the Royal Statistical Society, 1937, 4; S. 225.
- [PM88] Park, S. K.; Miller, K. W.: Random number generators: good ones are hard to find. In Communications of the ACM, 1988, 31; S. 1192–1201.
- [St08] Student: The Probable Error of a Mean. In Biometrika, 1908, 6; S. 1.
- [We10] Wes, M.: Data structures for statistical computing in Python, 2010.
- [Wi45] Wilcoxon, F.: Individual Comparisons by Ranking Methods. In Biometrics Bulletin, 1945, 1; S. 80.

Heuristische Optimierung durch menschliche Intuition – Das Beste aus zwei Welten

Iona Kuhn¹

Abstract: Algorithmen können exakte Lösungen finden, indem sie den Suchraum durchlaufen. Ist das Problem aber zu groß, haben Algorithmen oft Schwierigkeiten eine gute Lösung in akzeptabler Zeit zu finden. Menschen hingegen scheinen komplizierte Probleme oft schnell „intuitiv“ zu lösen. Ziel dieser Arbeit ist es daher, menschliche Intuition zur Verbesserung von Heuristiken am Beispiel von Job-Shop-Problemen zu nutzen. Bei Job-Shop-Problemen müssen mehrere Aufträge auf unterschiedlichen Maschinen möglichst schnell erledigt werden, also die richtige Bearbeitungsreihenfolge gefunden werden. Um die Intuition von Menschen verwenden zu können, wurde ein Gamification-Ansatz eingesetzt, also ein Spiel implementiert in dem Menschen das Problem in einer übertragenen, vereinfachten Form lösen. Um das gewonnene menschliche Wissen in Heuristiken einzuarbeiten wurden zwei verschiedene Ansätze entwickelt und mit diesen Vergleiche zur ursprünglichen Heuristik durchgeführt. Dabei konnte gezeigt werden, dass in der Tat die Effizienz von Heuristiken mit menschlicher Intuition gesteigert werden kann.

Keywords: Heuristik, menschliche Intuition, Gamification, genetischer Algorithmus, Job-Shop-Problem, case-based reasoning

1 Einleitung

Für viele Herausforderungen in der heutigen Welt verwenden wir Algorithmen. Ist ein Problem zu komplex (wie z.B. das Traveling Salesman Problem), kann es aber zu lange dauern, es mit einem exakten Verfahren zu berechnen. Dann setzt man eine Heuristik ein, die sich z.B. durch zufälliges Ausprobieren im Suchraum der Lösung annähert, statt den Suchraum vollständig zu durchsuchen. Dabei ist aber nicht gesichert, dass immer die beste Lösung gefunden wird. Außerdem können auch diese Verfahren sehr lange dauern.

Wenn Menschen solche komplexen Herausforderungen angehen, hilft ihnen oft ihre Intuition, eine möglichst gute Lösung zu finden. Allerdings sind viele dieser Probleme zu komplex, um alleine von einem Menschen gelöst zu werden. Darüber hinaus können Computer wesentlich schneller rechnen.

Daher ist das Ziel dieser Arbeit, Heuristiken zu entwickeln und mit Hilfe von Strategien, die man aus dem Vorgehen von Menschen gewinnt, zu verbessern. Hierzu wurde die Problemklasse der Job-Shop-Probleme als Anwendungsdomäne gewählt. Ein Job-Shop-Problem ist eine Unterklasse von Scheduling-Problemen. Dabei geht es darum eine Reihe von Aufträgen, die auf unterschiedlichen Maschinen bearbeitet werden müssen,

¹ Otto-Schott-Gymnasium, An Schneiders Mühle 1, 55124 Mainz, iona.kuhn@web.de

möglichst effizient abzuarbeiten, d.h. in der Reihenfolge mit der insgesamt kürzesten Laufzeit und mit den geringsten Leerzeiten für die Maschinen.

Als Heuristik wurde eine vereinfachte Form eines genetischen Algorithmus gewählt, da solche Algorithmen oft zur Lösung von dieser und vergleichbaren Problemklassen verwendet werden [MHZ10]. Zudem lässt sich bei der vereinfachten Form eines genetischen Algorithmus die heuristische Suchkomponente (bezeichnet als Mutationsoperator) leicht austauschen.

Um den genetischen Algorithmus für die gewählte Problemklasse nun zu verbessern, muss man zunächst an die Strategien herankommen, mit denen Menschen solche Herausforderungen intuitiv angehen. Dafür wurde ein Gamification-Ansatz gewählt. Das bedeutet, dass man ein Computerspiel entwickelt, das einem Job-Shop-Problem entspricht. Mit dem Wissen aus den Daten, welche die Testspieler erzeugen, kann dann der genetische Algorithmus angepasst und mit der ursprünglichen Version verglichen werden.

2 Verwandte Arbeiten

Bereits früher wurden Versuche unternommen, menschliche Erfahrungen algorithmisch zu nutzen. In [Wu02] wurde zum Beispiel ein Algorithmus auf alten Weisheiten aufgebaut um eine Verbesserung zu erzielen. Das algorithmische Nutzen von menschlichen Vorgehensweisen ist aber immer noch Bestandteil aktueller Forschungen. So wurden beispielsweise in [KUG18] bekannte (hypothetische) menschliche Verhaltensweisen beim Lösen des Traveling Salesmen Problems eingesetzt. Bei diesen beiden Arbeiten wurden jeweils vorhandene bzw. angenommene Strategien von Menschen verwendet, der Mensch aber nicht selbst eingesetzt, um Strategien zu erzeugen.

In [JBS17] wird gezeigt, dass man Heuristiken mit menschlichen Strategien verbessern kann. Die Autoren haben aber den Menschen immer konkret nach ihrer Strategie befragt und nicht zunächst unabhängig menschliche Strategien gesammelt.

Der Ansatz der Gamification wurde schon häufiger benutzt (siehe etwa [HKS14] für einen Vergleich verschiedener Ansätze), beispielsweise um die Motivation und das Engagement von Mitarbeiter eines Unternehmens zu erhöhen. In der hier beschriebenen Arbeit geht es jedoch darum Gamification zu nutzen um ein Verfahren zur Verbesserung von algorithmischen Heuristiken durch menschliche Intuition zu entwickeln.

3 Job-Shop-Problem

Bei dem hier eingesetzten Job-Shop-Problem sollen n Jobs auf m Maschinen bearbeitet werden. Jeder Job j besteht dabei aus t_j Teilarbeiten. Die Reihenfolge der Teilarbeiten eines jeden Jobs ist klar definiert. Innerhalb eines Jobs muss jede Teilarbeit auf einer

anderen Maschine ausgeführt werden. Welche Maschine das ist, steht von Anfang an fest. Die Bedingung dafür, wann mit der Arbeit an einer Teilarbeit begonnen werden kann, ist, dass die vorhergehende Teilarbeit fertig und dass die benötigte Maschine frei ist. Es gibt keinen Wartebereich, d.h. wenn eine Teilarbeit auf einer Maschine fertig ist, kann diese erst wieder gestartet werden, wenn das Produkt zur nächsten Maschine gebracht worden ist. Es werden keine Transportzeiten zwischen den Maschinen und keine Rüstzeiten betrachtet. Alle Maschinen sind vom Zeitpunkt 0 an immer verfügbar. Die Jobs können zum Zeitpunkt 0 beginnen. Es existieren keine parallelen Maschinen oder Maschinengruppen. Keine Maschine kann mehrere Teilarbeiten gleichzeitig bearbeiten. Alle Jobs haben die gleiche Priorität. Es gibt keine Freigabe- oder Fälligkeitstermine für die Jobs. Leerzeiten auf den Maschinen und Wartezeiten in den Jobs sind zulässig. Eine angefangene Teilarbeit kann nicht unterbrochen werden. Eine Maschine hat eine feste Bearbeitungsdauer. Jede Teilarbeit auf einer Maschine dauert also gleichlang (vgl. auch [He03, Re17]).

Die Zielsetzung ist, die gesamte Bearbeitungsdauer aller Jobs zu minimieren. Außerdem muss nun festgelegt werden, mit wie vielen Maschinen gearbeitet werden soll. Hier wurde $m = 6$ gewählt, damit es im Rahmen der Gamification für einen menschlichen Spieler nicht zu umfangreich wird.

4 Das Problem als Spiel – Gamification

Um an die Strategien heranzukommen, mit denen der Mensch ein Job-Shop-Problem mittels Intuition löst, wurde wie oben erläutert ein Gamification-Ansatz gewählt. Dazu wurde ein Spiel entworfen, das sechs Maschinen umfasst, mit denen unterschiedliche Produkte hergestellt werden können. Hier wurde dabei ein Kochspiel gewählt (Abb. 1), welches ein Spiel aus dem Genre des Zeitmanagements ist.



Abb. 1: Bild des Startbildschirmes des Spiels
(beispielhaft ist die Anleitung für die Gnocchi angezeigt)

Die sechs Maschinen des Job-Shop-Problems entsprechen dann sechs Küchengeräten, mit denen zehn verschiedene Produkte aus den Zutaten hergestellt werden können. Die Zutaten können unmittelbar zur Verfügung stehen, müssen aber auch zum Teil zuerst mit anderen Maschinen (und Zutaten) hergestellt werden. Das Ziel des Spieles ist es, die vorgegebene Bestellung in Form von Aufträgen (= Herstellung eines Produktes, vorher Job genannt) möglichst schnell abzuarbeiten. Wichtig dabei ist die Reihenfolge, in der die Aufträge durchgeführt werden.

Bei dem Spiel gibt es vier Level mit 3, 5, 7 bzw. 9 herzustellenden Aufträgen. Bei jedem Level wird beim Starten zufällig eine Bestellung mit der entsprechenden Anzahl an Aufträgen generiert. Jedes Level muss später von einem Spieler fünfmal gespielt werden, wobei ein Spieler in einem Level immer die gleiche Bestellung erhält. Diese Entscheidung wurde getroffen, damit jeder Spieler sich verbessern kann und so beim letzten Durchlauf eines Levels möglichst die optimale Zeit erreicht.

Im Folgenden sind alle 10 Aufträge, welche im Spiel auftauchen können in abstrakter Form (als Liste von Listen) angegeben. Dabei ist in den Unterlisten immer die Maschine (mit m1 = „Schäler“, m2 = „Schneider“, m3 = „Gnocchimaschine“, m4 = „Ofen“, m5 = „Verzierer“, m6 = „Mixer“) und die entsprechend für den Arbeitsschritt benötigt Zeit dazu aufgelistet. In der Reihenfolge, wie sie in der übergeordneten Liste sind, müssen sie auch abgearbeitet werden.

a1 („Pommes“):	[[m1, 10], [m2, 10], [m4, 20]]
a2 („Gnocchi“):	[[m1, 10], [m3, 20]]
a3 („Schokokuchen“):	[[m4, 20], [m5, 5]]
a4 („Milchshake“):	[[m6, 15]]
a5 („Obstsalat“):	[[m1, 10], [m2, 10]]
a6 („Schokoeis“):	[[m5, 5]]
a7 („Smoothie“):	[[m1, 10], [m6, 15]]
a8 („Schokomilchshake“):	[[m5, 5], [m6, 15]]
a9 („Pommes mit Ketchup“):	[[m1, 10], [m2, 10], [m4, 20], [m5, 5]]
a10 („Kakao“):	[[m5, 5]]

Die Spieldaten wurden für die Auswertung in einzelnen Textdateien gespeichert.

5 Genetischer Algorithmus

Ein genetischer Algorithmus besteht in der Regel aus einer Liste mit Lösungsindividuen, aus denen durch Selektion, Rekombination und Mutation neue Individuen erzeugt und mit Hilfe einer Fitnessfunktion bewertet werden. Dieses wird so oft wiederholt, bis eine hinreichend gute Lösung gefunden wird bzw. für eine vorher festgelegte Anzahl an Durchläufen (im folgenden Iterationen genannt). Bei dem hier verwendeten Algorithmus wurde auf die Rekombination verzichtet, wodurch das Rahmenwerk vereinfacht wird und es dadurch nur eine austauschbare Komponente gibt. Somit entspricht der gewählte genetische Algorithmus einer Art Hill-Climbing-Verfahren (welches unter Umständen

sogar vorteilhaft sein kann, vgl. [FM93]). Des Weiteren wurde auch der Selektionsoperator verändert, in dem nicht immer aus den besten Individuen neue erzeugt werden, sondern aus zufälligen. Dieser Ansatz wurde gewählt, da man teilweise über eine zunächst schlechtere Lösung eine deutlich bessere Lösung finden kann.

Auf das oben beschriebene Job-Shop-Problem übertragen ergibt sich entsprechend, dass ein Lösungsindividuum die Reihenfolge ist, in der bei einem Problem die einzelnen Aufträge durchgeführt werden. In der Liste der Individuen sind somit verschiedene Permutationen, von denen dann mittels der Fitnessfunktion die Zeit berechnet wird, die man bräuchte, wenn man die Aufträge in dieser Reihenfolge abarbeiten würde. Das Individuum mit der kürzesten Zeit ist dann dementsprechend auch das beste. Als Mutation werden zur Erzeugung neuer Individuen in zufällig ausgewählten Individuen zwei Aufträge vertauscht, sodass sich eine neue Reihenfolge der Abarbeitung ergibt. Nun werden die neuen Individuen der Liste hinzugefügt und man löscht die entsprechende Anzahl an Individuen mit der längsten Bearbeitungszeit für die Aufträge. Dieser Zyklus von Auswählen, Mutation, Auswertung und Löschen wurde dann für ein bestimmtes Problem im Regelfall 1000mal iteriert. Dies ergibt dann die beste vom genetischen Algorithmus gefundene Lösung mit Zeit und Reihenfolge der Aufträge.

6 Auswertung der Spieldaten

Das in Kapitel 4 beschriebene Spiel wurde von insgesamt 70 Spielern gespielt. Für das Ziel dieser Arbeit war es zuerst wichtig zu untersuchen, ob die Spieler im Laufe des Spiels etwas gelernt haben. Dieses ist in der Tat so, wie man in Abb. 2 beispielhaft für Level 3 und 4 erkennt (grüne Kurve). Im Durchschnitt wird die Zeit für ein Level mit jedem Durchlauf kürzer und somit beim letzten Durchlauf die schnellste Lösung der Spieler erreicht.

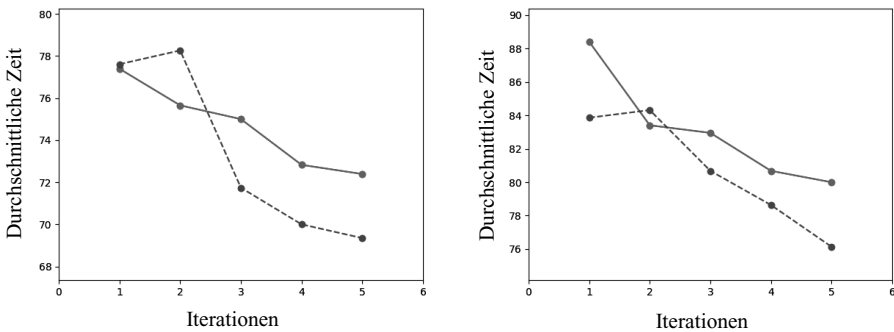


Abb. 2: Der Graph für das dritte Level ist links und der Graph für das vierte Level ist rechts. Dabei zeigt der grüne Graph die durchschnittliche Zeit (der erspielten Reihenfolge) des Menschen und der rote Graph die Zeit des genetischen Algorithmus.

Weiter stellt sich die Frage, ob der Mensch überhaupt besser ist als die Heuristik, also der genetische Algorithmus. Hierzu wurde das Spiel auch von einem genetischen

Algorithmus „gespielt“, dem die gleichen Bedingungen wie dem Menschen gegeben wurden (d.h. eine Individuenliste der Länge 1 und pro Iteration ein neues Individuum). Bei diesem Vergleich kann man deutlich erkennen, dass der genetische Algorithmus schlechter ist (rote Kurve in Abb. 2). Somit sollte es in der Tat möglich sein, den genetischen Algorithmus zu verbessern, indem man die menschlichen Strategien mit einbaut.

7 Nutzung der menschlichen Strategien

Nun soll die menschliche Intuition bzw. die hierbei verwendeten Strategien genutzt werden um neue Probleme zu lösen. Dafür wurden zwei Ansätze getestet: (i) das Fall-basierte Schließen (case-based reasoning, kurz: CBR) und (ii) die Anpassung des genetischen Algorithmus.

7.1 CBR-Algorithmus

Im Folgenden wird ein CBR-basierten-Ansatz (auf Basis von [BK08]) als alternative Heuristik eingesetzt um menschliche Intuition nutzbar zu machen. Damit nutzt man im Gegensatz zum genetischen Algorithmus, in dem die menschliche Intuition in aggregierter Form genutzt wird, das Einzelfallwissen der menschlichen Intuition. Ein CBR-Ansatz funktioniert allgemein so: Wenn ein neues Problem vorliegt, wird in einer Datenbank mit Problem/Lösungs-Paaren das ähnlichste Problem gesucht (retrieve). Dessen Lösung wird genommen (reuse) und für das neue Problem (revise) konvertiert. Die neue, angepasste Lösung wird dann üblicherweise als Lösung für das neue Problem zurückgegeben und in die Datenbank eingefügt (retain).

Dieser Ansatz wurde mit Hilfe von selbst entwickelten Verfahren auf das Job-Shop-Problem übertragen. Im Gegensatz zum herkömmlichen CBR-Verfahren wird das neue Problem-Lösungs-Paar nicht zur Datenbank hinzugefügt, da diese ausschließlich Problem-Lösungs-Paare enthalten soll, welche direkt vom Menschen erzeugt worden sind.

Der so entwickelte CBR-Ansatz wurde nun mit dem genetischen Algorithmus verglichen. Dafür wurden Probleme unterschiedlicher Länge gewählt: mit 55 Aufträgen (als ein sehr komplexes Problem) oder mit 9 Aufträgen (wie maximal beim Gamification-Ansatz). In beiden Fällen wurden für den genetischen Algorithmus die folgenden vier Varianten verwendet:

geneAlgo1: 1 Lösungsindividuum pro Generation, 1 neues Individuum pro Iteration
geneAlgo2: 2 Lösungsindividuen pro Generation, 1 neues Individuum pro Iteration
geneAlgo3: 20 Lösungsindividuen pro Generation, 1 neues Individuum pro Iteration
geneAlgo4: 20 Lösungsindividuen pro Generation, 2 neues Individuum pro Iteration

Die erste Variante dient als Beispiel für den einfachsten genetischen Algorithmus bzw. den Hill-Climbing-Ansatz. Bei den weiteren Varianten wurde die Anzahl an Lösungsindividuen erhöht, so dass der Suchraum einfacher (und schneller) erschlossen werden kann und sich der Algorithmus nicht so leicht in einem lokalen Optimum verfängt.

Das Problem mit 55 Aufträgen wurde vom CBR-Ansatz und den vier genetischen Algorithmen jeweils 20mal gelöst. Die Ergebnisse sind in Tab. 1 gezeigt. Die Ergebnisse des Vergleichs der vier genetischen Algorithmen und des CBR-Ansatzes mit den Problemen mit 9 Aufträgen sind in Tab. 2 gezeigt.

	beste gefundene Lösung	Durch- schnittlich gefundene Lösung	Anzahl gebrauchter Iterationen für Lösung	Laufzeit dafür [s]	Iterationen um beste Lösung vom CBR- Algorithmus zu finden	Laufzeit dafür [s]
geneAlgo1	320	332,00	502,55	119,01	44,10	11,96
geneAlgo2	320	333,75	471,75	112,72	41,55	11,83
geneAlgo3	320	333,50	756,25	197,22	108,70	35,49
geneAlgo4	320	328,25	617,60	297,50	46,10	31,51
Durch- schnitt	320	331,88	587,04	181,61	60,11	22,70
CBR-Algo	380	426,25		0,51		
Verhältnis	1,19	1,28		356,10		44,50
	> 19 % schlechter	> 28 % schlechter		> 356,10mal so schnell		> 44,50mal so schnell

Tab. 1: Ergebnisse des Vergleiches von CBR-Algorithmus und genetischen Algorithmus bei einem Problem der Länge 55.

	durchschnittlich gefundene Lösung	Anzahl gebrauchter Iterationen für Lösung	Laufzeit dafür [s]
geneAlgo1	78,96	20,49	5,92
geneAlgo2	78,74	19,71	6,08
geneAlgo3	77,86	27,85	13,79
geneAlgo4	77,90	15,11	14,36
Durchschnitt	78,37	20,79	10,04
CBR-Algo	89,33		0,07
Verhältnis	1,14		143,39
	> 14 % schlechter		> 143,39mal so schnell

Tab. 2: Ergebnisse des Vergleiches vom CBR-Algorithmus und genetischen Algorithmus bei Problemen der Länge 9 (Durchschnitt für 70 unterschiedliche Probleme).

Dass der CBR-Algorithmus eher schlechtere Lösungen findet, lässt sich bei der Länge 55 vermutlich dadurch erklären, dass keine so großen Probleme in der Datenbank gespeichert sind. Ein weiterer Grund könnte darin bestehen, dass der Mensch wahrscheinlich nicht immer die optimale Lösung beim Spielen des Spiels gefunden hat, und daher in der Datenbank auch Problem-Lösungs-Paare vorhanden sind, welche nicht optimal sind. Hervorzuheben ist aber, dass der CBR-Ansatz deutlich schneller zu einer leicht schlechteren Lösung gekommen ist, wohingegen die genetischen Algorithmen in diesem Fall deutlich langsamer sind. Dies liegt daran, dass der genetische Algorithmus viele Lösungen ausprobiert und durchrechnet, während der CBR-Algorithmus nur eine Lösung konvertieren muss. Zusammenfassend kann man sagen, dass mit menschlicher Intuition, eingesetzt in Form eines CBR-Ansatzes, nicht unbedingt die optimale Lösung gefunden wird, aber man aufgrund des Algorithmus deutlich schneller auf eine sehr gute Lösung kommt. Dies ist für viele Anwendungen von einem Job-Shop-Problem bereits ausreichend – vor allem im Hinblick auf die wesentlich geringere Berechnungszeit.

7.2 Anpassung des genetischen Algorithmus mit menschlichem Wissen

Um den genetischen Algorithmus mit den Strategien aus der menschlichen Intuition zu verbessern, mussten aus der Datenbank entsprechende Regeln abgeleitet werden. Um möglichst einfache, generische und kompakte Entscheidungsregeln zu erhalten, werden immer zwei Aufträge miteinander verglichen. Dafür werden die Merkmale der Aufträge hierfür in abstrakter Form beschrieben (bezeichnet als Bedingung). Die Reihenfolge, in der diese Aufträge gemäß Datenbank, also des Menschen, am besten abgearbeitet werden, ist dann die Schlussfolgerung basierend auf der Bedingung.

Mit den in Kapitel 4 beschriebenen 10 Aufträgen ergeben sich $45 (= 10 * 9 / 2)$ Paare an Aufträgen. Mit jedem dieser Paare wird dann in den gesammelten Daten geschaut, ob es ein Problem gibt, das die beiden Aufträge enthält. Wenn dieses der Fall ist, wird mit diesen beiden Aufträgen eine Datenzeile (siehe Abb. 3) bestehend aus Bedingung und Schlussfolgerung erzeugt.

`j1_KKE ^ j2_ES ^ gemein_3.1 ^ j2_kuerzer => j1_j2`

Abb. 3: Eine beispielhafte Datenzeile mit Bedingung und Schlussfolgerung. Die Bedingung ist vor dem Folgepfeil und besteht aus einigen Parametern, welche die beiden Aufträge abstrakt beschreiben und verbinden. Hinter dem Folgepfeil befindet sich die Schlussfolgerung, also welchen der beiden Aufträge der Mensch als erstes gestartet hat.

Dieses wurde mit allen 45 Kombination von zwei Aufträgen für jedes Problem in der Datenbank durchgeführt, in der diese beiden Aufträge vorkommen, woraus sich insgesamt 2583 Datenzeilen ergeben haben. Die Datei mit diesen Datenzeilen wurde dann mittels eines Wissensextraktions-Algorithmus [AK17] weiterverarbeitet, der eine mehrstufige Wissensbasis zurückgibt, wobei auf der ersten Stufe eine allgemeine Regel steht und auf den tieferen Stufen die Ausnahmen der Regel bzw. Ausnahmen von Ausnahmen, etc. D.h. dieser Algorithmus komprimiert die Datenzeilen zu Regeln. Dabei werden die Bedingungen zu sogenannten Prämissen und die entsprechende

Schlussfolgerung zur Konklusion. So wurde eine Datei mit den komprimierten Regeln erhalten, die in Abb. 4 gezeigt ist.

Die so erhaltenen Regeln müssen nun in den genetischen Algorithmus eingearbeitet werden. Dafür wurde der Mutationsoperator so verändert, dass vor einem Vertauschen der beiden Aufträge diese mit den erhaltenen Regeln überprüft werden. Dazu wird mit Hilfe eines Inferenzalgorithmus [AK16] kontrolliert, ob die beiden zur Mutation ausgewählten Aufträge schon in der richtigen Reihenfolge vorliegen. Ist dieses nicht der Fall, so werden diese getauscht, d.h. die Mutation erfolgt, und die neue Reihenfolge wird der Individuenliste hinzugefügt. Ansonsten erfolgt keine Mutation und es wird zufällig ein anderes Auftragspaar ausgewählt. Somit ist die Mutation abhängig von den Regeln, welche aus den gesammelten Daten über die menschliche Intuition erzeugt wurden.

```
j2_j1 [0.75]

j1_ES => j1_j2 [1.0]
gemein_1.3_2.4 => j1_j2 [1.0]
j1=j2 => j1_j2 [0.78]
j1_KKES => j1_j2 [0.98]
j2_KK => j1_j2 [0.98]
j2_SL => j1_j2 [0.86]
j1_KKE => j1_j2 [0.83]
j2_KL => j1_j2 [0.98]

gemein_2.2 ^ j2_SL => j2_j1 [1.0]
gemein ^ j1_S => j1_j2 [0.99]
gemein_1.1 ^ j1_S => j1_j2 [1.0]
gemein ^ j1_SL => j1_j2 [0.89]
gemein ^ j1_L => j2_j1 [0.87]
j2_L ^ j2_kuerzer => j2_j1 [0.92]

j1=j2 ^ j2_SL => j2_j1 [1.0]
gemein_1.1_2.2_3.3 ^ j1_KKE => j2_j1 [1.0]
gemein_2.2 ^ j2_kuerzer => j2_j1 [1.0]
gemein_2.2 ^ j1_SL => j1_j2 [1.0]
gemein_2.2 ^ j1_kuerzer => j1_j2 [1.0]
j1_KKE ^ j2_KKES => j2_j1 [1.0]
j1_KKE ^ j2_KE => j2_j1 [1.0]
j1_KKE ^ j1_kuerzer => j2_j1 [1.0]
gemein ^ j2_kuerzer => j2_j1 [0.87]
gemein ^ j2_KE => j1_j2 [0.86]
j1_kuerzer ^ j2_KKE => j2_j1 [0.84]

gemein_1.1 ^ j2_S ^ j2_kuerzer => j2_j1 [1.0]
gemein ^ j2_SL ^ j2_kuerzer => j1_j2 [1.0]
gemein ^ j1_SL ^ j1_kuerzer => j2_j1 [1.0]
```

Abb. 4: Die Regeln aus dem Wissensextraktions-Algorithmus.

Um zu überprüfen, ob der so angepasste genetische Algorithmus besser ist als der ursprüngliche, wurde wieder ein festgelegtes Problem mit immer den gleichen 55 Aufträgen 20 Mal von den beiden Algorithmen über 1000 Iterationen gelöst. Dabei wurden für die Individuen pro Liste zwei Längen getestet (1 und 20), wobei in beiden Fällen pro Zyklus ein neues Lösungsindividuum erzeugt wurde. Zum Vergleich der beiden Algorithmen wird überprüft, wie sich die durchschnittliche Zeit zur Herstellung der Aufträge mit jeder Iteration verändert. Dieses ist in Abb. 5 gezeigt.

Es ist zu erkennen, dass in beiden Fällen der Graph bei dem angepassten genetischen Algorithmus (grün) zu Beginn schneller absinkt. Mit steigender Anzahl der Iterationen wird dann aber die Lösung des ursprünglichen genetischen Algorithmus besser, d.h. der rote Graph schneidet den Grünen (jeweils in den Ausschnitten gezeigt) und findet letztendlich eine bessere Lösung. Die Erwartung war, dass der angepasste genetische Algorithmus schneller ist und die bessere Lösung findet.

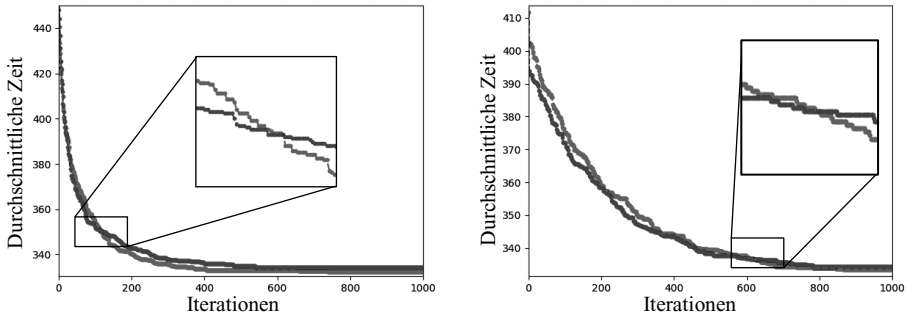


Abb. 5: Graphen des genetischen Algorithmus mit (grün) und ohne (rot) Intuition. Rechts: Individuenanzahl 1; links: Individuenanzahl 20.

Der angepasste genetische Algorithmus kann nur so gut sein wie die durch Intuition abgeleiteten Regeln, die von den Spieldaten ausgehen. Um diese zu überprüfen, wurden noch einmal die erzeugten Datensätze der Testspieler betrachtet. Für jedes Problem, das von einem Menschen gelöst worden ist, wurde überprüft, ob dies die beste Lösung ist. Dafür wurde jedes dieser Probleme in den ursprünglichen genetischen Algorithmus gegeben. Dann wurde die Lösung mit der Lösung verglichen, welche vom Menschen erspielt wurde. Dabei hat sich in der Tat gezeigt, dass der Mensch in knapp 30% der Fälle eine nicht so gute Lösung wie der genetische Algorithmus gefunden hat, d.h. das eingesetzte menschliche Wissen ist nicht perfekt. Der damit angepasste genetische Algorithmus findet aber trotzdem – gerade zu Beginn – schneller eine gute Lösung, aber letztendlich nicht die optimale. Dies beschreibt sehr genau, was ein Mensch mittels Intuition leistet: Diese weist den Menschen schnell in die Richtung für eine sehr gute Lösung, die aber nicht unbedingt die optimale sein muss.

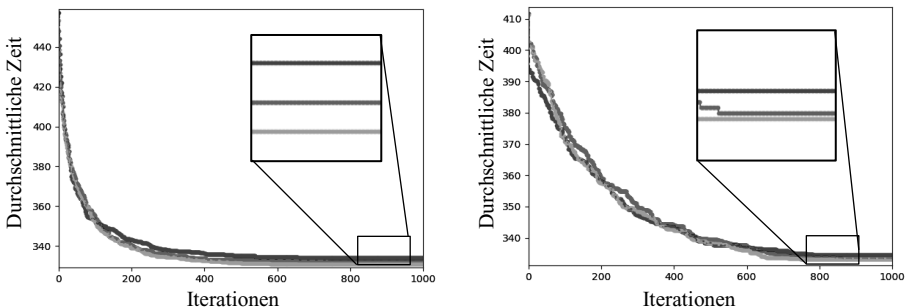


Abb. 6: Graphen des genetischen Algorithmus mit (grün) und ohne (rot) Wissen sowie für die kombinierte Form (gelb). Rechts: Individuenanzahl 1; links: Individuenanzahl 20.

Um nun die Vorteile der menschlichen Intuition mit denen des genetischen Algorithmus zu kombinieren, wurde die Idee verfolgt, ab einer bestimmten Anzahl von Iterationen auf den genetischen Algorithmus ohne Wissen umzuschalten. Gemäß den Graphen in Abb. 5 ist der Punkt, wo der ursprüngliche Algorithmus den angepassten überholt, bei ca. 110

Iterationen (1 Individuum in der Liste) bzw. 605 Iterationen (20 Individuen in der Liste). Diese wurden als „Umschaltunkte“ festgelegt und der so angepasste genetische Algorithmus wie zuvor auf das Problem mit 55 Aufträgen angewendet (gelbe Kurve in Abb. 6). Hier wurde das erwartete Ergebnis erhalten: die gelbe Kurve sinkt zu Beginn vergleichbar schnell ab wie die Grüne, wird dann aber nicht von der Roten überholt, sondern kommt auf den durchschnittlich besten Wert.

8 Zusammenfassung und Ausblick

Ziel dieser Arbeit war es, Heuristiken unter Einbeziehung von intuitiven Strategien zu verbessern, die man aus dem Vorgehen von Menschen bei der Problemlösung ableitet. Hierzu wurde als Basis ein genetischer Algorithmus eingesetzt, der Job-Shop-Probleme löst. Um an die menschlichen Strategien zu kommen, wurde ein Computerspiel aus dem Zeitmanagement-Genre entwickelt, das einem Job-Shop-Problem entspricht (Gamification). Dies wurde von den Menschen gespielt, wobei die Vorgehensweisen der einzelnen spielenden Personen aufgezeichnet wurde.

Diese gesammelten Daten über die menschliche Intuition wurden auf zwei Arten algorithmisch nutzbar gemacht. Zum einen mit Hilfe des CBR-Algorithmus, der mit dem genetischen Algorithmus verglichen wurde. Dabei hat sich gezeigt, dass die menschliche Intuition sehr schnell auf eine sehr gute Lösung kommt, aber nicht unbedingt die beste Lösung findet. Zum anderen wurde der genetische Algorithmus unter Einbeziehung der menschlichen Strategien basierend auf Intuition verbessert. Dafür wurden aus den menschlichen Daten Regeln extrahiert und mit diesen Regeln wurde dann der Mutationsschritt des genetischen Algorithmus abgewandelt. Dieses hat den genetischen Algorithmus dahingehend verbessert, dass bereits nach wenigen Iterationen schneller eine gute Lösung gefunden wird. Allerdings holt der ursprüngliche genetische Algorithmus mit der Zeit auf und kommt letztendlich zu einer besseren Lösung. Dieses spiegelt wieder, dass menschliche Intuition schnell zu einer guten Lösung führen kann, aber die Rechenleistung eines Computers – bei genügender Zeit – am Ende eine bessere Lösung findet.

Um das Beste beider „Welten“ zu kombinieren, wurde dann der genetische Algorithmus so programmiert, dass die Mutationen zunächst basierend auf den menschlichen Regeln erfolgen, gegen Ende aber wieder – wie ursprünglich – rein zufällig erfolgen. Dadurch ergab sich ein Algorithmus, der zu Beginn schneller zu einer guten Lösung kommt und diese letztendlich noch weiter verbessert. Bei der Kombination von CBR-Algorithmus und genetischem Algorithmus liegen erste Ergebnisse vor, die auf ein ähnliches Ergebnis kommen.

Somit ist es grundsätzlich möglich, eine Heuristik unter Einbeziehung von Strategien basierend auf menschlicher Intuition zu verbessern. Hierbei ist die Qualität der aus menschlicher Intuition generierten Daten sehr wichtig. Daher könnte es möglich sein, den Algorithmus noch weiter zu verbessern, wenn man noch mehr menschliches Wissen

höherer Qualität ansammelt. Man könnte nun auch untersuchen, ob eine Verbesserung basierend auf Strategien menschlicher Intuition ebenfalls für andere Arten von Problemen und/oder Heuristiken funktioniert.

9 Quellen

- [AK17] Apeldoorn, D.; Kern-Isberner, G.: Towards an Understanding of What is Learned: Extracting Multi-Abstraction-Level Knowledge from Learning Agents. In: Rus, V., Markov, Z. (Hrsg.) Proceedings of the Thirtieth International Florida Artificial Intelligence Research Society Conference. AAAI Press, Palo Alto, California (2017)
- [AK16] Apeldoorn, D.; Kern-Isberner, G.: When Should Learning Agents Switch to Explicit Knowledge? In: Benz Müller, C., Sutcliffe, G., Rojas, R. (Hrsg.) GCAI 2016. 2nd Global Conference on Artificial Intelligence. EPiC Series in Computing, vol. 41, pp. 174–186. EasyChair Publications (2016)
- [BK08] Beierle, C.; Kern-Isberner, G.: Methoden wissensbasierter Systeme: Grundlagen, Algorithmen, Anwendungen. Vieweg+Teubner, Wiesbaden (2008)
- [FM93] Forrest, S.; Mitchell, M.: Relative Building-Block Fitness and the Building Block Hypothesis. Foundations of Genetic Algorithms Volume 2, 109–126 (1993)
- [HKS14] Hamari, J.; Koivisto, J.; Sarsa, H.: Does Gamification Work? – A Literature Review of Empirical Studies on Gamification. In: 2014 47th Hawaii International Conference on System Science. IEEE, New York, S. 3025-3034, 2014.
- [He03] Henning, A.: Praktische Job-Shop Scheduling-Probleme. Jena 2003.
- [JBS17] Joseph, K.; J. Banks, C.; A. Shah, J.: Collaborative Planning with Encoding of Users' High-level Strategies. Thirty-First AAAI Conference of Artificial Intelligence. Association for the Advancement of Artificial Intelligence, Palo Alto, 2017.
- [KUG18] Kyritsis, M.; Ud Din, S.; Gulliver, S.: Quotient Algorithm: A Simple Human-Inspired Heuristic for Addressing the Travelling Salesperson Problem. 2018 Fifth HCT Information Technology Trends (ITT), Dubai, United Arab Emirates, 2018, pp. 47-49
- [MHZ10] Mönch, L. unter Mitarbeit von Habla, C., Zimmermann, J.: Entscheidungsmethoden in Unternehmensweiten Softwaresystemen. FernUniversität in Hagen, Hagen, 2010.
- [Re17] Reichert, R.: Scheduling Algorithmen: Einleitung in das Job-Shop Scheduling. SwissEduc (2017). Webseite (aufgerufen am 30.11.18): https://www.swisseduc.ch/informatik/theoretische_informatik/scheduling/einleitung.html
- [Wu02] Wu, Y.; Huang, W.; Lau, S.; Wong, C.; Young, G.: An effective quasi-human based heuristic for solving the rectangle packing problem. European Journal of Operational Research, Volume 141, Issue 2, pp. 341-358, 2002.

Adversarial N-player Search using Locality for the Game of Battlesnake

Maximilian Benedikt Schier,¹ Niclas Wüstenbecker²

Abstract: This paper presents an approach to designing a planning agent for simultaneous N-player games. We propose to reduce the complexity of such games by limiting the search to players in the locality of the acting agent. For Battlesnake, the game at hand, an iterative deepening search strategy utilizing both alpha-beta and \max^n search is suggested. Useful metrics for estimating player advantage are presented, especially using a diamond flood filler for measuring board control. Furthermore, the process of our heuristic parameter tuning with a grid search and a genetic algorithm is described. We provide a qualitative analysis of our algorithm's performance at the international artificial intelligence competition Battlesnake, Victoria. Here, our agent placed second in the intermediate division.

Keywords: Game Tree Search; Artificial Intelligence; N-player Game; \max^n ; Battlesnake

1 Introduction

Snake is a very popular game, well known from implementations on many arcade machines, personal computers and especially cell phones. The game of Battlesnake is a multiplayer variant of this classic where many snakes simultaneously fight for food and survival. The eponymous Battlesnake competition in Victoria, B.C., Canada is an annual event focusing on the development of artificial intelligence. Each team designs and implements an agent to slither their snake to victory. Competitors are grouped in multiple division depending on skill, with the intermediate division primarily targeting university students interested in the field of AI.

Traditionally, many snakes are based on simple reflex agents [RN03]. More cunning teams employ a game tree search to plan ahead. An especially popular choice in this domain is alpha-beta search because of its simplicity, efficiency, and familiarity. However, it is not a viable solution for Battlesnake as it is limited to two player games and therefore can not be used effectively. We apply a lesser-known variant of minimax, namely \max^n , and describe our experience, the process of our development and the parameter tuning of the heuristic using a genetic algorithm.

¹ Leibniz University Hannover, Institute for Information Processing, Appelstr. 9a, 30167 Hannover, Germany
schier@tnt.uni-hannover.de

² Leibniz University Hannover, Institute for Information Processing, Appelstr. 9a, 30167 Hannover, Germany
wuestenb@tnt.uni-hannover.de

Our main **contributions** are:

- We propose a general iterative deepening tree search approach using alpha-beta and \max^n search which reduces tree complexity using locality for N-player games
- We define a measure of locality for the game of Battlesnake, and adapt our algorithm
- We introduce several useful metrics to estimate the advantage of leaf nodes during search for Battlesnake
- Our heuristic parameter tuning approach using genetic optimization is outlined
- We perform a qualitative analysis of our agent in the context of the international artificial intelligence competition Battlesnake, Victoria
- Our implementation of the agent as described in this paper, which is also the version used during competition, is made publicly available³

2 Related Work

To the best of our knowledge, no prior work has been published on the game of Battlesnake. However, some work was published about search complexity reduction for general N-player games with \max^n search. Lisý *et al.* [Li10] suggest splitting a game into multiple potentially overlapping subsets of agents using game locality. These subsets may then be searched independently and are ultimately combined to form a joint plan. This approach shows many similarities to ours with regard to agent selection and masking but the play out and combination of all subsets render it more complicated to implement.

Another optimization to \max^n proposed by Lisý *et al.* is a procedural knowledge heuristic to guide tree search towards each agent's goal in their goal-based game-tree search algorithm [Li09]. However, this work uses hand-crafted or inferred opponent models, which have to be implemented on a per-game basis. This is generally much harder than simply defining the locality of agents. Whereas our method might reduce the accuracy of search by masking agents, this method does so by cutting branches which are considered unlikely to be selected.

A completely different approach to solving games with a discrete action space which has gained popularity has been reinforcement learning, for example with Deep Q-networks (DQNs). These networks learn to make intuitive decisions and have the potential to exceed all traditional heuristic-based agents as they can find metrics that are too complicated to be hand-engineered or simply not obvious to humans. Monte Carlo tree search using DQN value heuristics is already outperforming alpha-beta tree pruning for a variety of two-player games like chess and GO [Si18]. However, DQNs still require a lot of resources and time to be trained and their decision making is not transparent.

³ <https://github.com/m-schier/battlesnake-2019>

3 Battlesnake

The game Battlesnake is a multi-agent variant of the popular game Snake. In the classic game, a single player controls a snake consisting of multiple tiles on a square grid. Each turn the player can move the snake to an adjacent tile, afterward, the tail disappears. If a snake consumes randomly spawned food, it grows a tile as the tail remains during that turn. The same game mechanics apply to Battlesnake, however, the game adds more complexity. Consumed food will be placed on a free tile again with an exponentially increasing probability per turn. Failing to consume any food within 100 turns causes death by starvation. Additionally, a collision of the snake head with any other body part or trying to move outside of the board boundaries (wall collision) also kills the colliding snake immediately. If two or more snake heads move into the same tile simultaneously, all but the largest snake die, unless there is no difference in length between the largest, in which case all snakes die resulting in a draw.

Two to eight snakes play on one game board simultaneously with predetermined starting positions. The game board is always square and has one of the following sizes: 7×7 , 11×11 and 19×19 . An example board configuration is presented in figure 1.

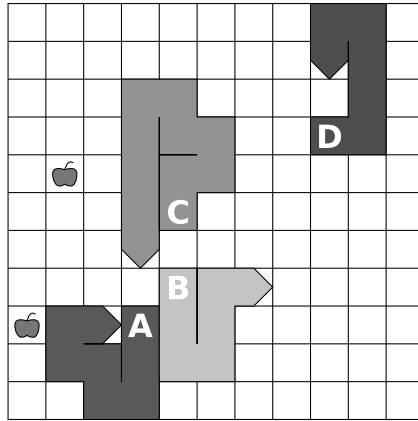


Fig. 1: Board state taken from one of our tournament games. Snake heads are marked with capital letters, tails are triangularly shaped. Food is depicted as an apple. In this state, A and C may cooperate to kill B in the next frame if A moves up and C down, leaving B with no option but running into itself or into the head of one of the longer snakes, which is exactly how this game played out.

4 Agent

An agent playing a game of Battlesnake is presented the current game state each frame and selects an appropriate movement action. In order to find a suiting algorithm for our agent, the game's characteristics were first classified:

- Discrete** There are always four actions a player may choose (North, East, South, West).
- Not deterministic** Food spawn positions are not known to agents and determined by a pseudo-random number generator.
- N-player** The game (usually) has more than 2 players.
- Simultaneous** In each state each agent must choose an action.
- Fully observable** The entire board as well as the attributes of each player are observable for every player.
- Zero-sum game** The outcome of a game of Battlesnake is only determined by the ranking of the players when the second to last player dies (the final state), where the ranking is determined by the frame in which a snake died, later being better. Consequently, each ranking position can be assigned a score such that the sum of all scores is always zero. If two or more snakes die in the same frame, this is considered a tie, in such a case the score of all ranking positions the snakes finished on will be averaged to still reach a sum of zero.

4.1 Tree Search

After classification, we evaluated that an agent using a tree search is well suited due to the game's small discrete action space, fully observable state and zero-sum nature. However, its non-deterministic, non-sequential play out is problematic for search. On the largest board size of 19×19 at the start of the game, the number of empty tiles where food may spawn is larger than 300 as few tiles are occupied. This spawn chance leads to a very high branching factor, which would limit search depth too severely to even warrant a tree search. Therefore, it was decided to not spawn any new food during game tree play-out making the game deterministic. The lack of food spawn would have to be compensated by some heuristic that measures how many tiles on the board are reachable for each snake.

Secondly, the simultaneous moves are of concern as simultaneous play requires forming a strategy to select the best action using a game equilibrium in unstable states where one action is not clearly dominant. We employ the concept of delayed move execution to sequentialize the search, where instead of performing all actions at once as usual, we apply each agent's action in sequence and only update the state with the final agent's move. During search, we follow the trivial pure strategy of always selecting the action with best minimax value. Such a strategy is not always stable and playing against a better, mixed strategy may not be optimal either, but does allow the game to be viewed in a purely sequential tree.

With these simplifications, Battlesnake is a discrete deterministic sequential multi or two-player zero-sum game. For games with more than two players, the \max^n search algorithm described by Luckhart and Irani [LI86] is used, which is a tree search algorithm where each player executes their move in turn and always selects the move giving them the best payoff for the given state. When only two players are left, a switch to alpha-beta pruning [KM75]

is performed. Alpha-beta pruning reduces the number of examined nodes in a minmax tree search by not playing out subtrees that an optimally playing opponent would not select.

4.1.1 Iterative deepening with adaptive play out selection

Iterative deepening search is often employed to find the best available action with a given time constraint and has been widely applied to alpha-beta and depth first search. While iterative deepening can be used just as easily with \max^n search, further optimization can be performed for any game with a notion of locality. Complete play out of all agents is not required if an opponent has minimal influence on our decision making. Any agent that may not be directly interacted with within the search depth, usually because of spatial locality, may be ignored. Choosing to not play out (mask) agents can greatly reduce complexity in these scenarios. In case of masking all but two agents, instead of \max^n search much more efficient alpha-beta pruning may even be used. We propose the general *IDAPOS* algorithm outlined in algorithm 1 for adaptive play out selection in N-player games. While the elapsed time t_{Elapsed} is less than the specified cutoff time t_{Cutoff} , the played out players p_{played} for the given depth and state are selected using a game specific function. The current state s is then masked to the state s_{masked} only containing played out players using a game specific masking strategy. Depending on number of played out players, the appropriate search algorithm is used and the best action a_{best} updated. Search depth is increased per loop iteration.

Algorithm 1 General iterative deepening game tree search with adaptive play out selection

```

function IDAPOS( $s, t_{\text{Cutoff}}$ )
  depth  $\leftarrow$  1
  while  $t_{\text{Elapsed}} < t_{\text{Cutoff}}$  do
     $p_{\text{played}} \leftarrow \text{SelectPlayedOut}(s, \text{depth})$ 
     $s_{\text{masked}} \leftarrow \text{Mask}(s, p_{\text{played}})$ 
    if  $|p_{\text{played}}| = 2$  then
       $a_{\text{best}} \leftarrow \text{AlphaBetaSearch}(s_{\text{masked}}, \text{depth})$ 
    else
       $a_{\text{best}} \leftarrow \text{MaxNSearch}(s_{\text{masked}}, \text{depth})$ 
    depth  $\leftarrow$  depth + 1
  return  $a_{\text{best}}$ 

```

This algorithm can be adapted to any game by implementing the selection and masking functions. The selection returns the set of players that may directly be interacted with within the cutoff depth, the masking hides all players not selected for search. We propose the following masking strategies for the general case, which strategy to use should be decided on a per-game basis:

Removal All masked agents are removed from the board.

Freezing All masked agents remain in their current state even if they have to perform an action.

Simple move The masked agents perform basic actions that require minimal computation time while keeping them alive.

4.1.2 Adaption for Battlesnake

Play out selection Due to the spatial locality of Battlesnake, situations where players are too far apart to interact directly within the search depth cutoff are often given. In these scenarios such players may be masked. Especially in the opening stages of the game, where up to eight players are present, planning an effective local strategy against neighboring players is much more important than correctly predicting developments at the far side of the board. The theoretical reduction of nodes evaluated compared to standard \max^n is shown in Figure 2. A significant improvement can be observed, especially on large board sizes. It should be noted that due to the symmetry of the board in starting positions, no alpha-beta pruning was used by *IDAPOS* in this figure, as it is impossible to be close to exactly one opponent.

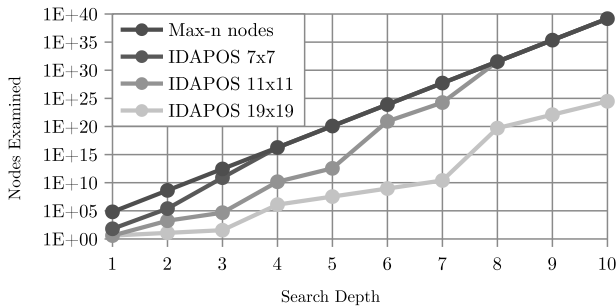


Fig. 2: Number of examined nodes with increasing search depth using *IDAPOS* presented in algorithm 1 with selection algorithm 2 for different board sizes in comparison with total tree playout. Snakes are at their starting positions.

For play out selection, all agents are chosen which the player could possibly interact with within the current search cutoff, where an interaction is either a head-on-head or head-on-body collision. Since the head of a snake moves into its 4-neighborhood each frame, the Manhattan or L_1 distance is an accurate lower bound for the number of turns required to reach a given tile. Thus any snake with its head within an L_1 distance of twice the search depth or less may be interacted with through head-on-head collision and with any body part within an L_1 distance of exactly the search depth or less may be interacted with through head-on-body collision. Algorithm 2 outlines the selection process.

Masking Removing masked agents from the board is the worst option for this game. Doing so opens up spaces on the board that should not be accessible, i.e. dead ends. In later

Algorithm 2 Play out selection for the game of Battlesnake

```

function SELECTPLAYEDOUT( $s$ , depth)
   $p_{\text{played}} \leftarrow \emptyset$ 
  for all  $p \in s.\text{players}$  do
    if Manhattan( $p.\text{head}$ ,  $s.\text{self.head}$ )  $\leq 2 \cdot \text{depth}$  then
       $p_{\text{played}} \leftarrow p_{\text{played}} \cup \{p\}$ 
    else
      for all  $\psi \in p.\text{parts}$  do
        if Manhattan( $\psi$ ,  $s.\text{self.head}$ )  $\leq \text{depth}$  then
           $p_{\text{played}} \leftarrow p_{\text{played}} \cup \{p\}$ 
          break
  return  $p_{\text{played}}$ 

```

stages of the game, snakes occupy dozens of tiles, freeing these up by removal changes the situation on the board greatly. Freezing is the safest option as there is no risk of suicidal behavior compared to a very basic simple move like running into a dead end. Making a simple move is the most realistic, but also the most complex due to time constraints. We developed a simple move approach where the agent tries to maintain direction while dodging collisions with itself and snakes of equal or larger length.

Locally connected groups An agent A may be too far from our own agent to affect our decision making but close enough to influence an agent B we are playing out. Thus our simulation of B 's behavior will be flawed as it does not consider agent A acting intelligently. To fix this issue, A would have to be included in the tree search, which is not desirable as A might invoke the same issue on a third agent, requiring the simulation of all locally connected agents. In order to keep the branching factor and tree complexity low, inaccuracies caused by this problem are tolerated.

Distortion of the heuristic by masking Another issue arises when not including the masked agents in the advantage calculation. Given the situation in figure 3 it is desirable for both agents to commit to eating, resulting in a tie. However, if the shown tiles were only part of the board with a third agent present out of view, it would not be in the best interest of both agents to commit to eating the food as the third agent would win. To resolve this issue while preserving the zero-sum nature of the game, an additional punishment for the own agent death was added. This offsets any disadvantage through weaker positioning or growth as predicted by our heuristic if the agent dies in a tie. Consequently, our agent only goes for a tie if the other options all lead to a loss. An additional implication of this modification is that all opponents are rewarded slightly for cooperating against our agent during the search, which is a strategy they are unlikely to actually follow.

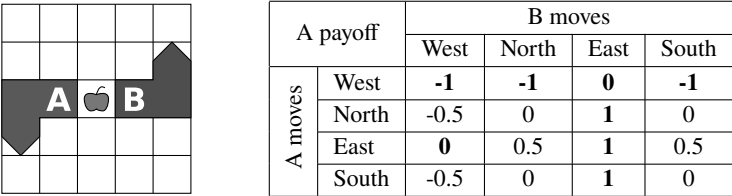


Fig. 3: A typical tie situation encountered in Battlesnake. Payoffs for A are given with terminal states marked bold. The heuristic is assumed to reward greater length by eating with 0.5 points advantage and being the sole survivor with 1 point advantage. Going for the food dominates all other actions for both A and B. However, carrying out their best available option kills them both in a head on head collision as they have equal length.

4.1.3 Scoring Leaf Nodes

Some additional biases were introduced in the scoring of leaf nodes to improve the agent’s behavior. In general, it may be beneficial to draw out a game as long as possible when facing defeat. Because we used a compiled language and had optimized our search implementation using profiling, we were confident that we could beat many competing agents in terms of search depth. Therefore, we rewarded surviving as long as possible given the chance the opponent was unaware of a branch due to his more restricted search depth. Another bias very specific to this game that was implemented for leaves was scoring of death reasons. It is preferable to die in an uncertain event rather than in a certain one. For example, given the choice between starving, and eating food but risking a head-on-head collision, it is always preferable to eat, as death is not certain. Alternatively, when the agent may only choose between colliding with a wall, its own body, or an enemy body, it should always go for the enemy. The enemy may die in the same frame but the wall or its own body won’t disappear. Therefore, the death reasons were biased in the following order from most favorable to least: head-on-head collision, head-on-body collision, starvation, own-body collision, wall collision.

4.2 Metrics

After analyzing a promising search strategy, the next step is to evaluate a game position on the board by applying a heuristic, consisting of several metrics. The computation time of a metric is very limited, as a longer computation time drastically decreases the search depth of our tree search.

Length Advantage The first metric we propose is a snake’s advantage in length. For the normal game of Snake, this is equivalent to the score, however, one has to be careful as a longer snake is also less versatile and can accidentally box itself in more easily. In general,

however, we found that length is an advantage because the snake body acts like an obstacle enemies have to avoid. To compute this metric we subtract the overall average snake length from our own length.

Board Control In addition to the length advantage, it is very important to be able to reach a tile faster than the enemy, or in other words: controlling a tile. This concept of tile control can be seen in a large number of board games, most notably the game of chess [Ha75]. In the past, many approaches used a Voronoi diagram [AK00] to estimate the number of controlled tiles. However, this method is not very accurate and can not be expanded easily. Therefore, we decided to count the number of controlled tiles using an efficient implementation of a queue-based diamond flooder [Le81] in adversarial operation. The flood fill algorithm starts with the queue initialized with each snake's head position, sorted by snake length in descending order. Since a snake's head moves into one of the directly adjacent four tiles each turn, a diamond flooder is used. The queue ensures that the snakes take turns filling out their next available tiles. The initial order guarantees that the largest snakes get to start each fill step. Thus contested tiles that multiple snakes may reach at the same time are attributed to the largest one. This fill method returns the exact number of tiles a snake can reach before any enemy. An example of a flood fill visualization can be seen in figure 4. Here, snake *B* has better control over the empty tiles and can, therefore, reach more potential food spawn locations first.

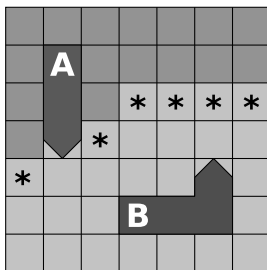


Fig. 4: Adversarial flood fill performed starting from the head of both snakes. Empty tiles are colored matching the snake they are controlled by. Tiles marked * are contested and could be reached by both snakes in the same amount of turns but are attributed to the larger snake *B*. *B* could force reaching the tile if both players go for it. The smaller snake *A* would die in a head-to-head collision.

Snake Health Another metric is the snake's health. As described in section 3 every snake can make 100 moves without eating food before starving. Therefore it is crucial to include this in the heuristic. A simple linear mapping of the health to a value in range $[0, 1]$ would not be sufficient, because this would favor eating food far too much. We factored in the distance to the nearest food computed by using the flood fill described above. The number of turns until starvation should always be greater than the distance to the nearest food item.

5 Experiments and Results

Based on the presented metrics, a heuristic can be constructed by weighting the metrics described in section 4.2. We first started by hand tuning the coefficients used to weight each metric. However, such an approach is not suited for smaller improvements as there is no systematic way of assessing the heuristic. Since an agent’s performance using a fixed search depth solely depends on the heuristic used, the search depth for all agents was reduced and a game could be simulated much faster. By playing out a large number of games we were able to calculate the average leaderboard position and thus gather statistical evidence on whether an agent’s strategy outperforms the others. Due to the game’s adversarial nature, it is not certain that self-play alone can result in the best strategy as there may be a counter-strategy that exploits it. To prevent this, multiple strategies from last years tournament were implemented: a greedy strategy that tries to consume food as fast as possible, an aggressive strategy which builds on the greedy strategy but attacks another snake when it is longer, and a tail chasing strategy that minimizes the distance between head and tail of the snake until the health is very low and then acts greedy until food is consumed. Any heuristic change could be validated by testing it against the simple agents and against the original heuristic before the change. To roughly estimate the importance of a metric a simple grid search was applied.

Genetic Optimization and Tests To further optimize the weights, a genetic algorithm which adjusted all coefficients of the heuristic using a bit string mutation on a string of 6 bits for each metric was used. The bit string represented a number that was mapped to the range determined by the grid search. We used the average leaderboard position as a fitness measure for the population. A total of 10,000 games with low search depth were carried out. The decisive advantage the heuristic found through genetic optimization has over other heuristics tested is shown in Table 1. The table on the right shows that our *IDAPOS* algorithm performs better than other tree search algorithms when using the same heuristic and same search time limit. While standard deviations are high due to the non-deterministic nature of the game, both our algorithm as well as heuristic are found to be statistically significantly better than the next best one using t-tests with $\alpha = 0.01$.

Heuristic	Average position	Algorithm	Average position
Genetic Optimization	3.33 (± 2.21)	<i>IDAPOS</i>	4.12 (± 2.13)
Greedy eater	4.15 (± 2.31)	max ⁿ	4.51 (± 2.21)
Aggressive	5.02 (± 2.13)	alpha-beta	4.64 (± 2.35)
Tail chaser	5.51 (± 1.87)	MinMax	4.72 (± 2.41)

Tab. 1: Average leaderboard positions after $n = 1,500$ simulated playouts with an eight player board and search time limit of 100ms. Left table shows different heuristics with alpha-beta search, while right table shows different algorithms with our heuristic found through genetic optimization. Standard deviation is shown in brackets.

Tournament Results Our team *Niedersächsische Kreuzotter* placed second in the intermediate division which consisted of 65 teams. It became apparent during the tournament that our agent tried to optimize the number of controlled tiles very well but sometimes failed to convert its length advantage into a real advantage. In other cases the agent focused too much on board control, neglecting to eat food placed at the edge of the board. This can be attributed to the fact that the agent would have to give up board control in order to eat. In these cases, the agent did not reach sufficient length and could easily be pressured by longer opponents. For these reasons, we, unfortunately, lost in the final stand-off. Overall, however, our agent played especially well during the opening stages of the game when most snakes were still alive. We mainly ascribe this to our \max^n search approach. The agent performed extraordinarily well in an event sponsor's unofficial exhibition tournament where each agent faced an increasingly large number of cooperating hostile agents, strongly outperforming participants of all divisions and winning a special price.

6 Conclusion and Outlook

In this paper, we introduced the game of Battlesnake as well as the international artificial intelligence competition of the same name. The game was classified and an iterative deepening tree search algorithm using both alpha-beta pruning as well as \max^n search was implemented. Our algorithm reduces the number of played out agents using the locality of the game by masking players which may not directly interact with ourselves within the depth limit. Special considerations required to perform a tree search with masked players in general, as well as specifically for this game, were outlined. We suggested several metrics which are useful for assessing non-terminal leaves during the search, notably, establishing a flood fill based metric for evaluating board control. Genetic optimization was used to fine tune coefficients and balance the influence of different metrics on the overall heuristic, outperforming both grid search as well as our initial hand-crafted heuristic. Our proposed *IDAPOS* algorithm was found to outperform both \max^n and alpha-beta search in this game using the same heuristic. Finally, a qualitative analysis of our approach compared to other tournament competitors showed that our agent generally outperformed the competition when many players are present on the board.

In the future, the efficiency of N-player search could be improved. As \max^n employs no pruning, it can generally only search a more shallow tree than alpha-beta pruning with a similar branching factor. In most of our game situations, the \max^n search depth was only four moves ahead while alpha-beta search computed up to eight moves ahead. Recent improvements in pruning N-person game trees have been made, notably, HyperMax [Fr14]. As outlined in our approach, the agent currently follows the pure strategy of selecting the action with the best minimax value. However, situations such as shown in figure 3 are unstable if more than two players are present. In these unstable situations, an accurate opponent model would increase the agent's expected payoff. Sturtevant *et al.* propose the probabilistic \max^n tree search prob- \max^n . This can outperform classic \max^n search in expected pay-off despite reducing the total number of nodes evaluated [SZB06].

7 Acknowledgements

We would like to thank the Institute for Information Processing at our university, the Leibniz University Hannover. Without the support of the Institute, especially Prof. Bodo Rosenhahn, we would not have been able to attend the competition in Victoria. Additionally, thanks to Florian Kluger and Frederik Schubert who provided guidance and accompanied us on the trip to Canada. More over, we would like to thank Christoph Reinders for his advise on this paper. Lastly, we want to thank all organizers and sponsors of the Battlesnake competition for making this event possible.

References

- [AK00] Aurenhammer, Franz; Klein, Rolf: Voronoi diagrams. *Handbook of computational geometry*, 5(10):201–290, 2000.
- [Fr14] Fridenfalk, Mikael: N-Person Minimax and Alpha-Beta Pruning. In: *NICOGRAPH International 2014*, Visby, Sweden, May 2014. pp. 43–52, 2014.
- [Ha75] Harris, Larry R: The heuristic search and the game of chess a study of quiescence, sacrifices, and plan oriented play. In: *Proceedings of the 4th international joint conference on Artificial intelligence-Volume 1*. Morgan Kaufmann Publishers Inc., pp. 334–339, 1975.
- [KM75] Knuth, Donald E; Moore, Ronald W: An analysis of alpha-beta pruning. *Artificial intelligence*, 6(4):293–326, 1975.
- [Le81] Levoy, Mark: Area flooding algorithms. *Two-Dimensional Computer Animation*, Course Notes 9 for SIGGRAPH, 82, 1981.
- [LI86] Luckhart, Carol; Irani, Keki B: An Algorithmic Solution of N-Person Games. In: *AAAI*. volume 86, pp. 158–162, 1986.
- [Li09] Lisỳ, Viliam; Bořanskỳ, Branislav; Jakob, Michal; Pěchouček, Michal: Adversarial search with procedural knowledge heuristic. In: *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*. International Foundation for Autonomous Agents and Multiagent Systems, pp. 899–906, 2009.
- [Li10] Lisỳ, Viliam; Bořanskỳ, Branislav; Vaculín, Roman; Pěchouček, Michal: Agent subset adversarial search for complex non-cooperative domains. In: *Proceedings of the 2010 IEEE Conference on Computational Intelligence and Games*. IEEE, pp. 211–218, 2010.
- [RN03] Russell, Stuart J.; Norvig, Peter: *Artificial intelligence - a modern approach*, 2nd Edition. Prentice Hall series in artificial intelligence. Prentice Hall, 2003.
- [Si18] Silver, David; Hubert, Thomas; Schrittwieser, Julian; Antonoglou, Ioannis; Lai, Matthew; Guez, Arthur; Lanctot, Marc; Sifre, Laurent; Kumaran, Dhharshan; Graepel, Thore et al.: A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, 362(6419):1140–1144, 2018.
- [SZB06] Sturtevant, Nathan; Zinkevich, Martin; Bowling, Michael: Prob-Maxⁿ: Playing N-Player Games with Opponent Models. In: *AAAI*. volume 6, pp. 1057–1063, 2006.

Baumautomaten als notationelle Variante logischer Matrix-Semantik

Lukas Grätz¹

Abstract: Automaten sind als Grundlage der theoretischen Informatik in der Regel Bestandteil einer Vorlesung über Automaten und Sprachen im Bachelor. Wahrheitsfunktionen werden für die Semantik der Aussagenlogik ebenfalls in einer Grundlagenveranstaltung behandelt. Wie sich nun herausstellt, sind beide Konzepte ab einem gewissen Abstraktionsgrad äquivalent und nur bezüglich Terminologie und Notation unterschiedlich: (aufsteigende) Baumautomaten und Matrizen für mehrwertige Logik. Neben dem deterministischen Fall ist die nicht-deterministische Semantik Thema meiner Masterarbeit.

Automata are a foundation of theoretical computer science and part of undergraduate lectures on automata and languages. Similarly, truth functions for the semantics of propositional logic are covered in foundational classes. Assuming a certain level of abstraction both concepts are equivalent and only differ in terms of notation and terminology: bottom-up tree automata for tree languages are equivalent to matrix semantics for many-valued logics. The deterministic case is enriched by non-deterministic semantics in my master's thesis.

1 Einleitung

Automaten sind als Grundlagen der theoretischen Informatik in der Regel Bestandteil einer Vorlesung über Automaten und Sprachen im Bachelor. Ein *endlicher deterministischer Automat* (DFA) ist ein Tupel $A_1 = (\{q_0, \dots, q_6\}, \Sigma, \delta, q_0, \{q_5\})$ über dem Alphabet $\Sigma = \{A, B, \dots, Z\}$ mit der durch Abb. 1 gegebenen Übergangsfunktion δ .

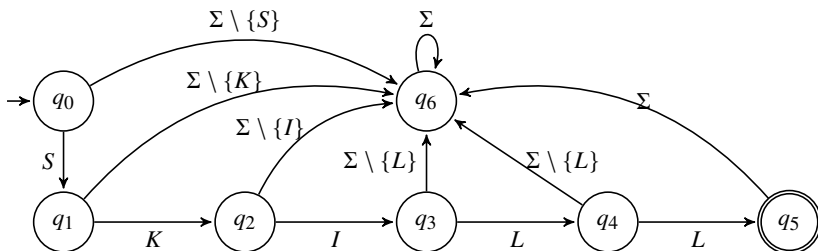


Abb. 1: Übergangsfunktion δ für den DFA A_1

Dieser Automat beginnt im Zustand q_0 zeichenweise ein Eingabewort aus den Symbolen des Alphabets Σ einzulesen. Der nächste Zustand ist durch das neu eingelesene Symbol jeweils

¹ Universität Leipzig, lukas.graetz@studserv.uni-leipzig.de

bezüglich der Übergangsfunktion δ determiniert (δ ist hier durch die Pfeile im Graph aus Abb. 1 gegeben). Falls sich der Automat am Ende der Eingabe im akzeptierenden Zustand q_5 befindet, so wird das Wort akzeptiert. Dieser Automat akzeptiert nur das Wort „SKILL“, also die Sprache $\{\text{SKILL}\}$.

Etwas nützlicher ist ein Automat, der Zeichenketten akzeptiert, die den Teilausdruck „SKILL“ enthalten. Dies kann z. B. mit einem *nichtdeterministischen endlichen Automaten* (NFA) $\mathbf{A}_2 = (\{q_0, \dots, q_5\}, \Sigma, \Delta, q_0, \{q_5\})$ erreicht werden.

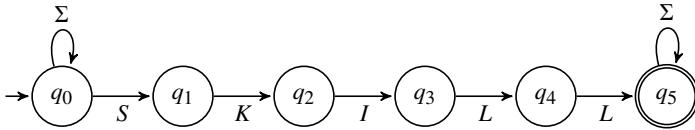


Abb. 2: Übergangsrelation Δ für den NFA \mathbf{A}_2

Beim NFA ist der nächste Zustand durch die Übergangsrelation Δ , gegeben durch Abb. 2, nicht mehr eindeutig determiniert, beispielsweise sind in q_0 beim eingelesenen Buchstaben S zwei Übergänge möglich. Der Automat akzeptiert, wenn Zustandsübergänge zum finalen akzeptierenden Zustand möglich sind. Das Wort „ABCSKSKILLDEF“ wird also akzeptiert; die akzeptierte Sprache kann auch durch den regulären Ausdruck $\Sigma^* \text{SKILL} \Sigma^*$ beschrieben werden.

Wahrheitsfunktionen werden für die Semantik der Aussagenlogik ebenfalls in einer Grundlagenveranstaltung behandelt. Aussagenlogik baut bereits auf einer Sprache von Formelausdrücken auf, die aus den Operatoren \wedge , \vee , \rightarrow und \neg sowie aus Klammern und Aussagenvariablen (für das Beispiel *motiviert*, *interessiert* und *willkommen*) besteht. Die Semantik ist durch eine Belegung der Aussagenvariablen mit den Wahrheitswerten 0, 1 und Wahrheitsfunktionen nach Abb. 3 gegeben:

f_{\wedge}	0	1
0	0	0
1	0	1

f_{\vee}	0	1
0	0	1
1	1	1

f_{\rightarrow}	0	1
0	1	1
1	0	1

	f_{\neg}
0	1
1	0

Abb. 3: Wahrheitsfunktionen für die Aussagenlogik

Sobald die Aussagenvariablen mit Wahrheitswerten belegt sind (z.B. mit $v(\text{motiviert}) = 0$, $v(\text{interessiert}) = 1$ und $v(\text{willkommen}) = 1$), ist die Bewertung v eines Formelausdrucks aus der Bewertung der Teilausdrücke und den Wahrheitsfunktionen für die Operatoren eindeutig bestimmt. Beispielsweise muss (1) gelten.

$$\begin{aligned}
 & v((\text{motiviert} \vee \text{interessiert}) \rightarrow \text{willkommen}) \\
 &= f_{\rightarrow}(f_{\vee}(v(\text{motiviert}), v(\text{interessiert})), v(\text{willkommen}))
 \end{aligned} \tag{1}$$

Die logische Folgerung ist durch die Konsequenzrelation \models definiert: Eine Aussage F folgt

aus einer Menge von Aussagen M , falls jede Bewertung, die alle Aussagen aus M mit 1 bewertet, auch die Aussage F mit 1 bewertet. So gilt z.B.

$$(\text{motiviert} \vee \text{interessiert}) \rightarrow \text{willkommen}, \text{interessiert} \models \text{willkommen}.$$

Automaten und die Semantik der Aussagenlogik sind also in ihrer Funktionsweise recht ähnlich: Der Zustand eines Automaten beim Wort ws ist durch den Zustand beim Teilwort w und dem neu eingelesenen Symbol s anhand der Übergangsfunktion determiniert. Der Wahrheitswert eines Ausdrucks $A \odot B$ ist durch die Wahrheitswerte der Ausdrücke A und B anhand der Wahrheitsfunktion von \odot determiniert.

Mitte der 1960er Jahre wurden Baumautomaten als natürliche Erweiterung der algebraischen Beschreibung endlicher Automaten entdeckt. Mehrwertige Logiken (und auch die Matrix-Semantik mehrwertiger Logiken) als Erweiterung der klassischen zweiwertigen Logik haben hingegen eine viel längere Tradition ab 1920, mit bekannten Namen wie Jan Łukasiewicz, Emil Post, Paul Bernays und Alfred Tarski (siehe [LT30]).

Nun stellt sich sogar heraus, dass Baumautomaten und logische Matrizen für mehrwertige Logiken einfach notationelle Varianten sind. Zunächst lassen sich aussagenlogische Formeln bekanntermaßen auch als Syntaxbaum darstellen, wobei in den Blättern Aussagenvariablen und in den Knoten logische Operatoren stehen. Eine Variablenbelegung entspricht dann genau einer Initialzuweisung der Blattsymbole mit Startzuständen. Generell entsprechen die Zustände eines Automaten den Wahrheitswerten, ausgezeichnete Wahrheitswerte sind die akzeptierenden Endzustände.

Doch zunächst seien hier die Definitionen (bezüglich aktueller Literatur) wiedergegeben. Ich habe die Notation nur unwesentlich geändert, um trotz üblicher (und sinnvoller) Vereinheitlichung der Notation einen Vergleich beider Strukturen zu ermöglichen, ohne dass der zugehörige Begriffsapparat den Umfang dieses Aufsatzes sprengt.

2 Baumautomaten

Baumautomaten unterscheiden sich von DFAs insofern, als dass keine Zeichenketten sondern Bäume eingelesen werden. Ein Baumautomat akzeptiert also keine Sprache aus Wörtern, sondern einen Wald, d.h. eine Menge aus Bäumen. Die folgende Definition ist angelehnt an [GS97, S. 9]. Hierfür sei Σ ein *Rangalphabet*, ein Alphabet in dem jedem Symbol $s \in \Sigma$ eine Stelligkeit $\text{Rang}(s) \in \mathbb{N}$ zugeordnet wird. Das auch als Blattalphabet bezeichnete Alphabet X hat keine Stelligkeit. Alternativ kann die Beschriftung eines Blattes auch ein nullstelliges Symbol sein ($s \in \Sigma$ mit $\text{Rang}(s) = 0$).

Definition 1. Sei Σ ein *Rangalphabet* und X ein Alphabet. Ein (deterministischer) *aufsteigender Baumautomat* ist ein Tupel $\mathbf{A} = (\mathcal{A}, \alpha, Q_+)$ mit einer endlichen Algebra $\mathcal{A} = (Q, \{s^{\mathcal{A}} \mid s \in \Sigma\})$, einer Initialzuweisung $\alpha: X \rightarrow Q$ und einer Teilmenge $Q_+ \subseteq Q$, den sogenannten akzeptierenden Endzuständen.

Genau genommen ist zwischen einem Symbol $s \in \Sigma$, dessen Stelligkeit $\text{Rang}(s) = m$ und der zugehörigen m -stelligen Operation $s^{\mathcal{A}}$ bezüglich der Algebra $\mathcal{A} = (Q, \Sigma)$ zu unterscheiden [GS97, S. 5]. Falls der Begriff „Algebra“ hier für Verunsicherung sorgen sollte: Eine Algebra $\mathcal{A} = (Q, \{s^{\mathcal{A}} \mid s \in \Sigma\})$ ist nichts weiter als eine Menge Q und Operationen $s^{\mathcal{A}}$, deren Argumente und Rückgabe Elemente von Q sind.

Die Initialzuweisung α kann eindeutig zu einer Zuweisung $\alpha^{\mathcal{A}}$ erweitert werden, die jedem Baum einen Zustand $q \in Q$ zuordnet: $\alpha^{\mathcal{A}}(s(t_1, \dots, t_m)) = s^{\mathcal{A}}(\alpha^{\mathcal{A}}(t_1), \dots, \alpha^{\mathcal{A}}(t_m))$ für Teilbäume t_1, \dots, t_m . Ein Baum t wird durch \mathbf{A} akzeptiert, falls $\alpha^{\mathcal{A}}(t) \in Q_+$.

3 Mehrwertige Matrizen

Die folgende Definition ist angelehnt an [BB92, S. 20]:

Definition 2. Sei $\mathcal{A} = (A, \{f_i \mid i \in I\})$ eine Algebra. Jedes Tupel der Form (\mathcal{A}, A_*) mit A_* einer nichtleeren Teilmenge heißt Matrix. Wenn $\mathcal{M} = (\mathcal{A}, A_*)$ eine Matrix ist, dann wird \mathcal{A} die Algebra von \mathcal{M} genannt.

Eine Belegung der Aussagenvariablen Var ist eine Funktion $v: \text{Var} \rightarrow A$ in die Wertemenge A der Algebra \mathcal{A} . Die Belegung v kann nun zu einer Bewertung $v^{\mathcal{A}}$ erweitert werden, mit $v^{\mathcal{A}}(\odot(a_1, \dots, a_m)) = f_{\odot}(v^{\mathcal{A}}(a_1), \dots, v^{\mathcal{A}}(a_m))$ für Teilausdrücke a_1, \dots, a_m .

Beispiel 3. Die Aussagenlogik lässt sich durch die Matrix $\mathcal{M} = ((\{0, 1\}, \{f_{\wedge}, f_{\vee}, f_{\rightarrow}, f_{\neg}\}), \{1\})$ definieren, wobei die Operationen durch Abb. 3 gegeben sind.

4 Ein Automat für Aussagenlogik

Wie oben schon beschrieben, ist die Übersetzung einer mehrwertigen Matrix in einen Baumatomen ausschließlich eine Frage der Notation. Dies soll nun kurz an einem schon vorgestellten Beispiel durch den *aufsteigenden Baumatomen* $\mathbf{A}_3 = (\mathcal{A}_3, \alpha_3, \{1\})$ mit $\mathcal{A}_3 = (Q_3, \Sigma_3)$, $Q_3 = \{0, 1\}$,

$$\begin{aligned} \Sigma_3 = \{ \wedge^{\mathcal{A}_3} : & \quad Q_3 \times Q_3 \rightarrow Q_3 : (0, 0) \mapsto 0, (0, 1) \mapsto 0, (1, 0) \mapsto 0, (1, 1) \mapsto 1, \\ \vee^{\mathcal{A}_3} : & \quad Q_3 \times Q_3 \rightarrow Q_3 : (0, 0) \mapsto 0, (0, 1) \mapsto 1, (1, 0) \mapsto 1, (1, 1) \mapsto 1, \\ \rightarrow^{\mathcal{A}_3} : & \quad Q_3 \times Q_3 \rightarrow Q_3 : (0, 0) \mapsto 1, (0, 1) \mapsto 1, (1, 0) \mapsto 0, (1, 1) \mapsto 1, \\ \neg^{\mathcal{A}_3} : & \quad Q_3 \rightarrow Q_3 : 0 \mapsto 1, 1 \mapsto 0 \} \end{aligned}$$

und $X_3 = \{\text{motiviert}, \text{interessiert}, \text{willkommen}\}$ dargestellt werden. Eine Belegung ist noch kein Bestandteil einer Matrix, da für die Konsequenzrelation jede Belegung der Aussagenvariablen X_3 betrachtet werden muss. Eine Belegung entspricht dann z. B. der Initialzuweisung

$$\alpha_3: X_3 \rightarrow Q_3: \text{motiviert} \mapsto 0, \text{interessiert} \mapsto 1, \text{willkommen} \mapsto 1.$$

Der Baum in Abb. 4 stellt die Formel $(motiviert \vee interessiert) \rightarrow willkommen$ dar. Da der Baumautomat diesen Baum akzeptiert, handelt es sich bei α_3 um eine erfüllende Belegung.

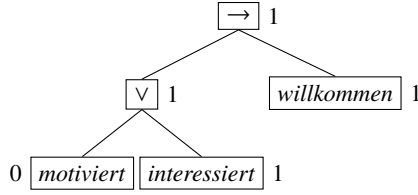


Abb. 4: Baum inklusive Zuweisung bezüglich Baumautomat A_3

5 Reguläre Sprachen als Logiken

Jeder DFA kann in eine logische Matrix umgewandelt werden: Zustände werden dann Wahrheitswerte, akzeptierende Endzustände werden zu ausgezeichneten Wahrheitswerten und aus den Symbolen werden einstellige Postfixoperatoren (in umgekehrter polnischer Notation).

Sei $A = (Q, \Sigma, \delta, q_0, Q_+)$ ein DFA und $w = s_1 s_2 \dots s_n$ ein Eingabewort aus den Symbolen $s_i \in \Sigma$ für $1 \leq i \leq n$. Bekanntermaßen wird w von A akzeptiert, falls $\delta(\dots \delta(\delta(q_0, s_1), s_2) \dots, s_n) \in Q_+$.

Der Automat lässt sich mit Hilfe des üblichen Transitionsmonoids in eine Matrix $M = (\mathcal{A}, Q_+)$ umschreiben, mit $\mathcal{A} = (Q, \mathcal{O})$ und

$$\mathcal{O} = \{f_s : q \mapsto \delta(q, s) \mid s \in \Sigma, q \in Q\} \cup \{f_\epsilon : q_0\}.$$

Jedes Symbol im Eingabewort wird zu einem 1-stelligen Postfixoperator, also jeweils angewandt auf den Wahrheitswert (vormals Zustand) des Präfixes. Das leere Wort ϵ als Präfix vor dem ersten Symbol ist eine Ausnahme. Es erhält als 0-stelliger Operator den Wahrheitswert q_0 . Umgeschrieben in gewohnte Klammerschreibweise wäre der Ausdruck also $s_n(\dots s_2(s_1(\epsilon)) \dots)$.

Der Wahrheitswert eines Wortes unter der Bewertung $v^{\mathcal{A}}$ erfüllt die Bedingungen $v^{\mathcal{A}}(s_1 s_2 \dots s_i) = f_{s_i}(v^{\mathcal{A}}(s_1 s_2 \dots s_{i-1}))$ für alle $1 \leq i \leq n$ und es gilt $v^{\mathcal{A}}(\epsilon) = f_\epsilon = q_0$. Damit gilt $v^{\mathcal{A}}(w) \in Q_+$ genau dann, wenn der Automat das Wort akzeptiert. Die Sprache des Automaten $\mathcal{L}(A)$ besteht also aus genau den bezüglich der Matrix M erfüllbaren Ausdrücken.

6 Fazit und Ausblick

Die zentrale Einsicht aus diesem Aufsatz ist nicht Abschnitt 4, dass aufsteigende Baumautomaten für die Aussagenlogik erstellt werden können (dies wird z.B. auch durch Example

2.2 in [GS84, S. 61] illustriert). Auch die Umwandlung eines endlichen Automaten in eine logische Matrix bietet hier nur eine Illustration, die eine Verbindung zwischen Grammatik und Semantik formaler Sprachen aufzeigt.

Die eigentliche Erkenntnis ist, dass Baumautomaten und die Semantik mehrwertiger Logik zusammenfallen: Einerseits die Verallgemeinerung der klassischen Zeichenkettenautomaten und andererseits die Erweiterung der zweiwertigen Semantik der klassischen Aussagenlogik. Definition 1 und 2 fallen also zusammen, ein feiner Unterschied besteht jedoch: Die Belegung wird unabhängig von der Matrix angegeben, wohingegen die Initialzuweisung Bestandteil des Baumautomaten ist. Ein Forschungsthema könnte sein, die Theoreme, Sätze und Lemmata beider Gebiete ineinander zu überführen. Eine guter Überblick über weiterführende Themen zur Automatentheorie bietet [FV09], für mehrwertige Logik sei auf [Hä01] verwiesen.

Und schließlich werden Automaten anders eingesetzt, denn ein Automat berechnet, ob ein gegebenes Wort bzw. ein gegebener Baum Bestandteil der vom Automaten akzeptierten Sprache ist. Mit einem Baumautomaten wird eine Sprache definiert und nicht eine semantische Relation (die logische Folgerung) zwischen den Ausdrücken der Sprache. Als Analogie kann man mit einer logischen Matrix und einer Belegung zeigen, dass ein Ausdruck erfüllbar ist, eine logische Folgerung ist für eine Matrix definiert.

Das Thema meiner Masterarbeit ist eine relativ neue Entwicklung der nichtdeterministischen Semantik [AZ11] und speziell der nichtdeterministischen Semantik für Modallogik auf Basis von [OS16] und [Ke81]. Spannend ist dann auch die Beziehung zwischen nichtdeterministischer Semantik und nichtdeterministischen Automaten.

7 Danksagung

Ein besonderer Dank geht an Dr. Peter Steinacker und Prof. Dr. Andreas Maletti für die Betreuung meiner Masterarbeit.

Literatur

- [AZ11] Avron, A.; Zamansky, A.: Non-deterministic Semantics for Logical Systems. In (Gabbay, D.; Guenther, F., Hrsg.): Handbook of Philosophical Logic. 2. Aufl., Bd. 16, Springer, Dordrecht, S. 227–304, 2011, ISBN: 978-94-007-0478-7.
- [BB92] Bolc, L.; Borowik, P.: Many-Valued Logics 1: Theoretical Foundations. Springer, Berlin Heidelberg New York, 1992, ISBN: 3-540-55926-4.
- [FV09] Fülöp, Z.; Vogler, H.: Weighted Tree Automata and Tree Transducers. In (Droste, M.; Kuich, W.; Vogler, H., Hrsg.): Handbook of Weighted Automata. Springer, Berlin Heidelberg, S. 313–403, 2009, ISBN: 978-3-642-01491-8.

- [GS84] Gécseg, F.; Steinby, M.: *Tree Automata*. Akadémiai Kiadó, Budapest, 1984, ISBN: 963-05-3170-4.
- [GS97] Gécseg, F.; Steinby, M.: *Tree Languages*. In (Rozenberg, G.; Salomaa, A., Hrsg.): *Handbook of Formal Languages*. Bd. 3, Springer, Berlin Heidelberg, S. 1–68, 1997, ISBN: 3-540-60649-1.
- [Hä01] Hähnle, R.: *Advanced Many-valued Logics*. In (Gabbay, D.; Guenther, F., Hrsg.): *Handbook of Philosophical Logic*. 2. Aufl., Bd. 2, Springer, Dordrecht, S. 297–395, 2001, ISBN: 978-90-481-5753-2.
- [Ke81] Kearns, J. T.: *Modal Semantics without Possible Worlds*. *The Journal of Symbolic Logic* 46/1, S. 77–86, 1981.
- [ŁT30] Łukasiewicz, J.; Tarski, A.: *Badania nad rachunkiem zdań = Untersuchungen über den Aussagenkalkül*. Sprawozdania z posiedzeń Towarzystwa Naukowego Warszawskiego, Wydział 3, Nauk Matematyczno-Fizycznych = *Comptes rendus des séances de la Société des Sciences et des Lettres de Varsovie, III, Classe des Sciences Mathématiques et Naturelles* 23/, S. 30–50, 1930.
- [OS16] Omori, H.; Skurt, D.: *More Modal Semantics without Possible Worlds*. *The IfCoLog Journal of Logics and their Applications* 3/5, 2016.

Security

Evaluation von Anti-Spoofing-Methoden mittels der Face Liveness Detection für die Verwendung in Web-Applikationen

Lisa Jasmin Moos¹

Abstract: Die Gesichtserkennung hat sich aufgrund ihrer einfachen und benutzerfreundlichen Art rasant weiterentwickelt und wird bereits heute häufig als Authentifizierungsmethode verwendet. Gesichtserkennungssysteme sind allerdings anfällig für Spoofing-Angriffe in Form von Portraitfotos, Videos und 3D-Masken. Diesen Herausforderungen stellen sich Methoden und Technologien aus dem Bereich der Lebendigkeitserkennung (Liveness Detection). Insbesondere Online-Anwendungen können von entsprechenden Technologien, bspw. im Rahmen der Authentifizierung, profitieren. In diesem Kontext existierte bisher kein Lösungsansatz, der die Besonderheiten aus dem Webbereich im Rahmen der Lebendigkeitserkennung berücksichtigt. Daher wurde eine auf Web-Applikationen optimierte Lösung konzipiert und prototypisch umgesetzt. Die Basis des Konzepts bildet die Analyse bestehender Verfahren der Lebendigkeitserkennung im Hinblick auf ihre Verwendbarkeit im Webkontext. Diese Anforderungsanalyse und der Abgleich mit dem Stand der Forschung sowie die Architektur des Prototyps und dessen Evaluation werden im vorliegenden Beitrag vorgestellt.

Keywords: Lebendigkeitserkennung; Spoofing; Echtheitsüberprüfung; Web-Applikationen

1 Motivation

Aufgrund ihrer zahlreichen Anwendungen in den Bereichen Zugangskontrolle, Internetkommunikation, Überwachung, Strafverfolgung, Sicherheit und Computerunterhaltung hat die Gesichtsidentifizierung, auch Face Recognition (FR) genannt, allgemein große Beachtung gefunden [HA15]. Dies spiegelt sich in der Prognose, dass der Markt der FR im Jahr 2021 auf 6,84 Milliarden Dollar anwachsen wird, wider. Hauptgrund für dieses Wachstum ist die Überwachung [Re17]. So wird die FR beispielsweise bereits heute erfolgreich in China eingesetzt, um Verkehrssündern ihre Bußgeldbescheide automatisch zuzusenden [Ri17].

Die vermehrte Verwendung der FR liegt darin begründet, dass diese mittlerweile eine hohe Genauigkeit erzielt [Ro18]. Ein großes Problem dieser Systeme ist allerdings die Sicherheit. Neben der bekannten Gefahr des Datendiebstahls bergen FR-Lösungen viel Potential für unautorisierte Zugriffe und letztlich Identitätsdiebstahl [GMF14]. Die Beschaffung der Daten zur Authentifizierung ist in diesem Fall nicht erforderlich. In einem Ende 2018 durchgeführten Test konnten beispielsweise 40 Prozent der Smartphones

¹ Hochschule für Wirtschaft und Gesellschaft, Fachbereich Dienstleistungen und Consulting, Ernst-Boehe-Str. 4, Ludwigshafen, 67059, LisaJasmin.Moos@studmail.hwg-lu.de

durch ein Portraitfoto entsperrt werden [Wo19]. Eine Anti-Spoofing-Methode, die Face Liveness Detection (FLD), bietet verschiedene Verfahren, um vor solchen unautorisierten Zugriffen zu schützen [Yn18].

Web-Applikationen stehen bei der Verwendung von FR vor den allgemeinen Herausforderungen der Gesichtsidentifizierung, wie der unterschiedlichen Beleuchtung, der Verdeckung von Teilen des Gesichts und dem Alterungsprozess [HA15]. Überdies führen die potenziell große Nutzeranzahl und die Möglichkeit des ungestörten Zugriffs vieler Anwender zu einem oftmals höheren Aufkommen von Spoofing-Angriffen. Insbesondere der Faktor der Ungestörtheit ist hierbei problematisch, da viele Angriffsversuche, welche technisch schwer erkennbar sind, von Menschen direkt erkannt werden. Ein Beispiel ist der Einsatz von Juggalo-Masken, die von Menschen schnell als offensichtlich gefälscht erkannt werden [Pl18]. Überdies sind Tiefenabbildungen durch Web-Kameras nicht ohne weiteres möglich [SBL15]. Web-Applikationen weisen somit über die bspw. im Kontext von Smartphones auftretenden Herausforderungen hinaus weitere Hindernisse im Einsatz von FR auf. Ein etabliertes Lösungskonzept zum Umgang mit den genannten speziellen Herausforderungen existiert bisher nicht. Aus diesem Grund stellt sich die Frage, welche FLD-Methoden es zur Verhinderung von Täuschungsmanövern gibt und welche dieser Verfahren hinsichtlich des Einsatzes in Webanwendungen vielversprechend erscheinen. Als Basis dieser Anforderungsanalyse wird im Folgenden der aktuelle Stand der Forschung im Bereich der FLD vorgestellt.

2 State of the Art der Face Liveness Detection (FLD)

Es gibt zahlreiche Ansätze zur Implementierung der FLD, allerdings zeigte ein durchgeführtes Literaturreview, dass diese noch nicht im Hinblick auf die Verwendung im Webkontext bewertet wurden. Daher werden in diesem Kapitel zum einen die Grundlagen existierender FLD-Konzepte und zum anderen deren Prüfung auf die Verwendbarkeit im Webkontext vorgestellt.

2.1 Analysen basierend auf Lebenszeichenerkennungen

Lebenszeichenerkennungen ermitteln die Lebendigkeit des Gesichts, indem sie überprüfen, ob sich ein bestimmter Gesichtsteil auf natürliche Art und Weise bewegt [KD12].

Erkennungen basierend auf Augen-, Kopf- und Lippenbewegungen (GTB)

Um die Bewegungen eines Körperteils zur Lebendigkeitserkennung nutzen zu können, muss dieses zuvor mehrmals erkannt werden. Im Anschluss werden die Differenzen der Aufnahmen berechnet, welche letztendlich mit einem Schwellenwert abgeglichen werden. Ist das Ergebnis größer als der Schwellenwert, so handelt es sich voraussichtlich um ein echtes Gesicht [KD12]. Die Erkennungen der Gesichtsteilbewegungen (GTB) gewähren einen Schutz gegen Spoofing-Angriffe, welche keine natürlichen Bewegungen

enthalten [KD12] und können in Webanwendungen eingesetzt werden.

Blinzelerkennung

Die Blinzelerkennung (Blink Detection) kann sowohl analog zu dem eben beschriebenen Verfahren als auch mittels des Eye Aspect Ratios (EAR) berechnet werden. Der EAR basiert auf dem Verhältnis der Entfernungen zwischen den Gesichtsmerkmalen der Augen (siehe Abb. 1) [SČ16].

$$\text{EAR} = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$



Abb. 1: Eye Aspect Ratio

Im Zähler dieser Gleichung wird der Abstand zwischen den vertikalen Gesichtsmerkmalen des Auges berechnet. Im Nenner wird der Abstand zwischen den horizontalen Gesichtspunkten ermittelt und gewichtet. Die Gewichtung ist notwendig, da ein Satz von horizontalen Punkten zwei Sätzen von vertikalen Punkten gegenübersteht. Bei einem geöffneten Auge ist der EAR relativ konstant. Er fällt jedoch schnell gegen null, wenn die Person blinzelt [SČ16]. Dieses Verfahren kann ebenfalls Angriffe erkennen, die keine natürlichen Bewegungen beinhalten und eignet sich für den Einsatz in Web-Applikationen.

Schätzung der Blickrichtung (GDE)

Merkmalsbasierte Methoden bilden das Blickmerkmal (z. B. die Pupille) auf die Blickposition ab [Ca14]. Diese Art der Blickrichtungsschätzung erfordert jedoch eine qualitativ hochwertige Kamera sowie mehrere Lichtquellen. Aussehens-gestützte Methoden hingegen lokalisieren zu Beginn die Augenregion und kartieren diese im Gesamten zur Blickposition. Im Gegensatz zur zuvor genannten Variante der Gaze Direction Estimation (GDE) bestehen in diesem Fall weder besondere Anforderungen an die Qualität der Kamera noch an die Beleuchtung. Dieses Verfahren ist jedoch noch nicht ausgereift und schützt daher nur gegen Angriffe, bei denen keine Augenbewegungen vorkommen [AŽŠ15]. Eine Verwendung der Aussehens-gestützten Methoden ist in Webanwendungen möglich.

2.2 Methoden basierend auf Bewegungsanalysen

Bewegungsanalysen schätzen die Tiefeninformationen eines Gesichts ab, da ein aus 2D-Material gewonnenes Gesicht einen anderen Bewegungsfluss als ein echtes Gesicht aufweist [KD12].

Optischer Fluss (OF)

Die Grundidee des optischen Flusses basiert auf der Annahme, dass ein 3D-Gesicht eine

2D-Bewegung erzeugt, die in zentralen Flächenbereichen (bspw. der Nase) höher ist, als bei den äußeren Gesichtsbereichen, wie den Ohren [SD14]. Daher bewegen sich weiter entfernte Objekte anders als Objekte, die näher an der Kamera sind. Mit der Information über die Positionen der Gesichtsteile und ihrer Geschwindigkeit ist es möglich zu determinieren, wie schnell diese im Verhältnis zueinander sind. Ein 2D-Angriff erzeugt jedoch eine gleichmäßige Bewegung. Ein Nachteil dieses Verfahrens besteht darin, dass Beleuchtungsänderungen einen negativen Effekt auf die Resultate haben. Dasselbe gilt für Sonnenbrillen und Bärte, weswegen dieses Verfahren nicht im Webkontext verwendet werden kann. Der OF schlägt auch dann fehl, wenn das unechte Gesicht nicht planar (eben) ist [SD14].

Defocus

Defocus ist eine Methode, die auf dem bestehenden Verfahren der variablen Fokussierung basiert. Für diese Lösung ist die Bildqualität von elementarer Bedeutung, da das Modell nur mit hochwertigen Bildern korrekte Ergebnisse liefern kann [SBL15]. Aus diesem Grund ist eine Verwendung in Web-Applikationen nicht angeraten. Um unechte Gesichter zu erkennen werden die Eigenschaften aus normalisierten Bildern extrahiert und daraufhin klassifiziert.

2.3 Bildqualitäts- und texturbasierte Analysen

Bildqualitätsanalysen ziehen die Bildqualität im Gesichtsbereich zur Unterscheidung heran [KD12]. Hierbei werden chromatische Bewegungsmerkmale zur Klassifizierung verwendet.

Bei texturbasierten Methoden wird davon ausgegangen, dass Oberflächenreflexionen und Verformungen des Gesichts durch verschiedene Betrugsverfahren auftreten [KD12].

Frequenz- und texturbasierte Analyse (FTA)

Die Frequenz- und texturbasierte Analyse überprüft Bilder hinsichtlich ihrer Form und Detailliertheit, um echte Gesichter von 2D-Papiermasken zu unterscheiden [SD14].

Dieses Verfahren kombiniert einen Deskriptor (Beschreiber) für die Texturanalyse mit einer Spektrum-basierten Methode für die Frequenzanalyse und kann in Webanwendungen verwendet werden, jedoch wird kein Schutz gegen 3D-Angriffe geboten.

Face Flashing (FF)

Beim Face Flashing wird das zu authentifizierende Gesicht illuminiert und anhand des zurückfallenden Lichts sowie der benötigten Zeit, wird berechnet, ob ein Spoofing-Angriff vorliegt. Dies ist möglich, da unsere Haut das Licht nicht auf die gleiche Art zurückwirft, wie beispielsweise ein Bildschirm oder eine Silikonmaske. Zur Lebendigkeitsbestimmung wird Licht in verschiedenen Farben von einem LCD-Bildschirm abgegeben (Challenge) und eine Kamera fängt das zurückkommende Licht ein (Response).

Im Anschluss werden die Faktoren Zeit und Form der Antwort analysiert. Die Zeit stellt hierbei sicher, dass der Wert nicht falsifiziert wurde [Ta18].

Dieses Verfahren besitzt den Nachteil, dass es durch Sonneneinstrahlungen und andere Helligkeitsverhältnisse beeinflusst wird. Daher ist das FF momentan nicht für Web-Applikationen geeignet. An einer Lösung des Problems wird derzeit durch Parametrisierung der Methode geforscht. Das Verfahren ist überdies lediglich als Sicherheitsmechanismus gegen 2D-Angriffe ausgelegt, wobei ein weiteres feingranuliertes Training auch einen Schutz gegen 3D-Angriffe ermöglichen sollte [Ta18].

Analyse des Blutflusses der Haut (SBFA)

Die Skin Blood Flow Analysis (SBFA) ist ein Ansatz zur Erkennung der Gesichtsliebeidigkeit, mit dem Foto- und Videoangriffe durch ein einziges Gesichtsbild identifiziert werden können [Wa17].

Durch die Extraktion der unterschiedlichen Eigenschaften im Rot- und Grünkanal des Bilds können Unterschiede im Blutfluss der Haut aufgezeigt werden. Diese werden mit der Farbverteilung in lokalen Bereichen der Gesichtsbilder und den spiegelnden und streuenden Komponenten, die durch Oberflächenreflexionen und Verformungen in ebenflächigen 3D- und 2D-Objekten auftreten, kombiniert und daraufhin klassifiziert. Für eine erfolgreiche Klassifizierung ist neben einer guten Kamera eine recht geringe Distanz zur zu authentifizierenden Person unerlässlich [Wa17]. Folglich ist diese Methode für Webanwendungen nur bedingt geeignet.

Komponentenabhängige Deskriptor-basierte Analyse (KDA)

Dieses Verfahren teilt das Gesichtsbild in mehrere Komponenten, welche lokalisiert und deren Eigenschaften daraufhin codiert werden. Im Anschluss werden die Codes mit Gewichtungen zusammengefasst. In der KDA wird hierbei primär auf die Schärfe des Bildes, Schattierungen und den Reflexionsgrad geachtet [SD14]. Diese Methode schützt ausschließlich gegen 2D-Spoofing in Form von Bild- und Videoangriffen und benötigt darüber hinaus eine vergleichsweise hohe Rechenleistung. Daher ist die Nutzung einer clientseitigen Architektur für diese Methode momentan nicht angeraten, da Endbenutzergeräte überhitzen können [Sh16]. Des Weiteren benötigt der Algorithmus länger zur Berechnung. Eine Verwendung in Web-Applikationen ist allerdings dennoch möglich.

Binäre Klassifizierung

Bei der Binären Klassifizierung erfolgt eine Einteilung in zwei Kategorien. Im Fall der FLD ist dies ein echtes oder unechtes Gesicht. Die binäre Klassifizierung macht sich zu Nutze, dass 2D- und 3D-Modelle, die aus 2D-Materialien gewonnen werden, nicht die Feinheiten eines echten Gesichts aufweisen [SD14]. Die Binäre Klassifizierung kann mittels eines tiefen neuronalen Netzes umgesetzt werden [In18]. Dieses bietet in den meisten Fällen einen Schutz gegen Foto- und Videoangriffe, wobei auch ein Training zur Erkennung von 3D-Angriffen möglich ist. Hierfür wird allerdings eine sehr große Datenmenge für eine korrekte und sichere Klassifizierung benötigt.

3 Anforderungen an FLD im Webkontext

Um die Anforderungen im Webkontext zu definieren wurde das erweiterte Unified Theory of Acceptance and Use of Technology (UTAUT)-Modell herangezogen. Dieses sagt aus, dass das Nutzungsverhalten u. A. von der Leistung der Anwendung, dem Vertrauen der Nutzer (in Bezug auf den Datenschutz) und den vom Nutzer erforderlichen Aufwand abhängig ist [CK06].

Eine Schwierigkeit besteht darin, dass im Webkontext potenziell eine hohe Nutzeranzahl bedient werden muss und die Nutzer sehr unterschiedliche Hardware verwenden (insb. im Vergleich zur FLD für Smartphones) [Lo03]. Da eine Nutzung von einer möglichst großen Anzahl an Anwendern gewünscht ist und diese unterschiedliche Hardware verwenden, sind derzeit Methoden, die auf modernste Technologien setzen, nicht praktikabel. Dies liegt darin begründet, dass nur eine gewisse Anzahl an potenziellen Nutzern beispielsweise Wärmebildkameras besitzt, was die Nutzung der Applikation mit einem hohen Aufwand verbindet [CK06].

Wie eingangs erwähnt ist auch die Häufigkeit der Spoofing-Angriffe und deren Vielfältigkeit mit hoher Wahrscheinlichkeit größer, da diese unbeobachtet geschehen können. Aus diesem Grund werden an die Genauigkeit und somit die Sicherheit des Algorithmus hohe Anforderungen gestellt. Außerdem sollte die FLD-Methode lediglich eine Web-Kamera zur Überprüfung benötigen, sodass keine weitere Hardware erforderlich ist.

Ferner soll die Anwendung überall und mit jedem Betriebssystem verwendet werden können, allerdings ohne hierbei die Sicherheit zu vernachlässigen, wobei die Sicherheit gegenüber 3D-Spoofing-Angriffen aufgrund der noch schwierigen Erstellung solcher Angriffe nachrangig behandelt werden kann. Da mit den Bildern der Web-Kamera gearbeitet wird, muss die FLD-Methode außerdem robust gegenüber Lichteinflüssen und sonstigen Störfaktoren, wie Bärten oder Brillen, sein. Auch die Wartungsfreundlichkeit spielt eine nicht zu vernachlässigende Rolle, die im Zusammenhang zur Anzahl der Abhängigkeiten eines Algorithmus steht, da diese nicht direkt mit dem restlichen System gewartet werden können [Lo03].

Da die Leistung einen relevanten Aspekt darstellt, sollte eine möglichst geringe Antwortzeit vorliegen. Die Bereitstellung der Applikation für eine große Anzahl an Nutzern bringt hohe Speicheranforderung mit sich, da große Datenmengen gespeichert werden müssen und die Nutzer eine möglichst schnelle Authentifizierung wünschen. Dabei liegt der Fokus nicht auf der Erfüllung der Echtzeitanforderung, sondern wie oben motiviert auf einer korrekten Authentifizierung. Die aktive Nutzeraufforderung in Form von Challenge-Response-Verfahren zählt nicht zu den FLD-Methoden und beeinträchtigt überdies die Zufriedenheit der Nutzer durch eine verminderte Usability [CK06]. Hieraus ergeben sich die in Abb. 2 aufgezeigten Anforderungen mit entsprechender Priorisierung.

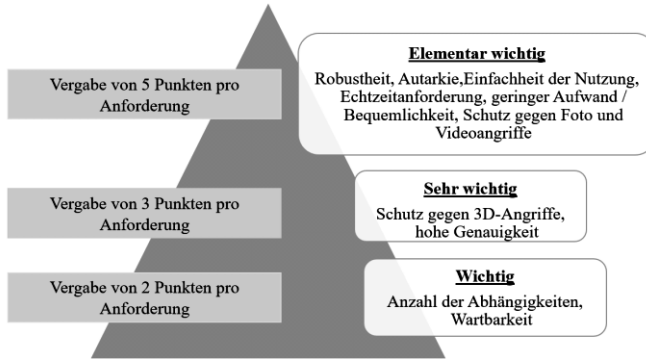


Abb. 2: Anforderungen im Webkontext

Aus den Anforderungen wurden Kriterien für die FLD im Webbereich abgeleitet (siehe Abb. 3).

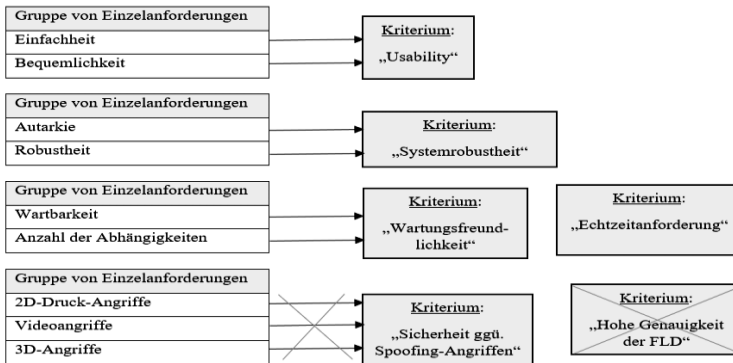


Abb. 3: Kriterien für Face Liveness Detection Methoden

Die Sicherheiten gegen **2D-Druck- / Foto-Angriffe**, **Videoangriffe** und **3D-Angriffe** werden, da sich die FLD-Verfahren gerade in diesem Bereich stark unterscheiden, separat als Kriterien aufgenommen. Daneben wird die Anforderung der **Echtzeit** als Kriterium genutzt. Da die zur Auswahl stehenden FLD-Verfahren, unter korrekter Anwendung, alle eine hohe Genauigkeit aufweisen, kann diese Anforderung allgemein als erfüllt angesehen werden.

4 Methodenauswahl

Alle Methoden, welche nicht gegen die Systemrobustheit verstoßen, können in Webanwendungen eingesetzt werden. Das FF wird weiterhin optimiert, um eine Robustheit gegenüber unterschiedlichen Lichtquellen herzustellen [Ta18], sodass dieses auch für

Web-Applikationen verwendet werden kann. Bei der Verwendung der KDA ist zu beachten, dass dieser Ansatz rechenintensiver als die vorherigen Methoden ist, da für einzelne Gesichtskomponenten Identifizierungen und Berechnungen stattfinden [SD14]. Die SBFA ist zwar in Bezug auf äußere Faktoren, wie Lichteinflüsse, robust, allerdings muss eine gute Bildqualität vorliegen, weswegen sich diese Methode nur bedingt im Webkontext anwenden lässt. Tab. 1 stellt einen Überblick der FLD-Methoden in Bezug auf deren Einhaltung der zuvor festgelegten Kriterien bereit.

Methoden	2D	3D	Video	Usability	System-robustheit	Wartungsfreundlichkeit	Echtzeit
GTB	+	+	-	+	+	+	+
Blinzel-erkennung	+	+	-	+	+	+	+
GDE	+	+	-	+	+	+	+
OF	+	-	+	+	-	+	+
Defocus	+	-	+	+	-	-	+
FTA	+	-	+	+	+	+	+
FF	+	-	+	-/+	-	+	+
SBFA	+	+	+	+	-	+	+
KDA	+	-	+	+	+	-	+
Binäre Klassifizierung	+	-/+	+	+	+	+	+

Tab. 1: allgemeiner Anforderungsabgleich

Die Kriterien wurden mit den in Kapitel 3 festgelegten Punktzahlen gewichtet für jedes evaluierte FLD-Verfahren vergeben (siehe Tab. 2). Insgesamt konnten 42 Punkte erzielt werden und somit konnte die prozentuale Erfüllung dieser berechnet werden. Sowohl die Binäre Klassifizierung als auch die FTA sind im besonderen Maße für Webanwendungen geeignet. Die FTA schützt jedoch nicht gegen 3D-Angriffe, weswegen der Verbund mit einer Blinzelerkennung, GTE oder einer Erkennung der GTB angeraten ist, um statische 3D-Angriffe abzufangen. Die Binäre Klassifizierung benötigt zwar eine große Menge an Trainingsdaten, um 3D-Angriffe korrekt zu klassifizieren, allerdings kann diese hiermit alle relevanten Spoofing-Angriffe abwehren.

GTB	Blink Detection	GDE	OF	Defocus	FTA	FF	SBFA	KDA	Binäre Klassifizierung
37	37	37	29	25	39	19	32	35	39
88%	88 %	88%	69%	60%	93%	45%	76%	83%	93%

Tab. 2: Gesamtpunktzahlen der Methoden

5 Entwicklung und Evaluation des Prototypen

Wie dem allgemeinen Anforderungsabgleich zu entnehmen ist, liefert die Binäre Klassifizierung unter Bereitstellung einer sehr großen Datenmenge einen vielversprechenden Ansatz. Auch die Verwendung der FTA in Verbund mit der Blinzelerkennung unter Verwendung des EARs ist sinnvoll, da Letztere mit der Ausnahme von Videos alle Spoofing-Angriffe abwehrt und überdies weniger Rechenleistung benötigt, als es bei der Erkennung von GTB der Fall ist [SČ16]. Die Kombination der beiden Methoden gewährt einen Schutz gegen alle relevante Spoofing-Verfahren. Da die Binäre Klassifizierung auch in der Lage ist 3D-Angriffe zu erkennen, wurde diese der FTA vorgezogen.

Die Binäre Klassifizierung wurde mittels eines vortrainierten tiefen neuronalen Netzes implementiert. Dieses konnte bereits 2D-Angriffe abwehren und wurde mittels des 3D Mask Attack Datasets des Dataset Distribution Portals weiter trainiert, um auch 3D-Masken von echten Gesichtern unterscheiden zu können [EM13]. Ferner wurde festgelegt, dass bei einer Erkennungsrate ab 96 % das Gesicht als lebendig eingestuft wird. Außerdem wurde ein bestehender Riegelalgorithmus übernommen, um bei einer erfolgreichen Authentifizierung des als lebendig erkannten Gesichts optisches Feedback zu geben.

Die Ergebnisse (siehe Abb. 4) zeigen, dass das echte Gesicht erkannt und die Anwendung entsperrt wurde, wohingegen das Video, das Foto und die Gipsmaske als Spoofing-Angriff identifiziert wurden. Daher erfüllt der Prototyp der Binären Klassifizierung alle gestellten Anforderungen.



Abb. 4: Ergebnisse der Binären Klassifizierung

Um die Funktionalität der FLD auch clientseitig im Browser zu testen, wurde eine eigens mittels AngularJS programmierte FR-Applikation um den Sicherheitsaspekt der implementierten FLD erweitert. Die hierzu erstellten Unified Modeling Language (UML)-Diagramme (siehe Abb. 5) zeigen den Aufbau der Webanwendung.

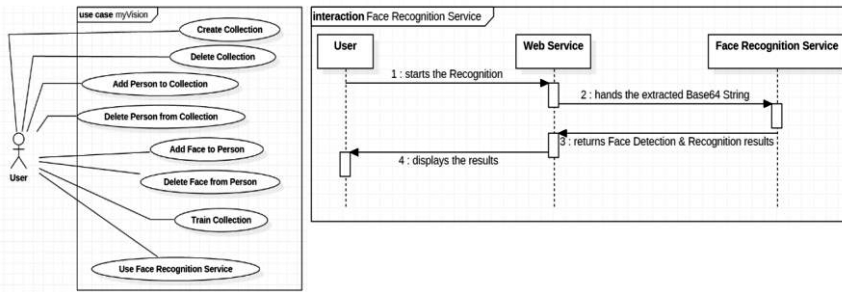


Abb. 5: Use Case und Sequenzdiagramm der Gesichtserkennungs-Applikation

Die Applikation wurde inklusive einer Alterserkennung implementiert, welche bei ersten Tests bereits ohne FLD-Verfahren einen gewissen Schutz gegen Spoofing-Angriffe bereitstellte, da die 68-Gesichtspunkte bei eigens erstellten Angriffsmaterialien von denen echter Gesichter abweichen. So wurde bspw. die Gipsmaske nach der Integration nicht mehr als Gesicht erkannt (siehe Abb. 6).



Abb. 6: Alterserkennung als bedingter Schutz gegen Spoofing-Angriffe

Im Anschluss wurde die Binäre Klassifizierung mithilfe von TensorFlow.js umgewandelt und analog zur Alterserkennung integriert. Um die Sicherheit der implementierten FLD-Methode zu überprüfen, wurde die Genauigkeit der Binären Klassifizierung berechnet. Hierzu wurden, neben eigenen Testdaten, die Bilder der Nanjing University of Aeronautics and Astronautics (NUAA) Potograph Imposter Datenbank verwendet [Ta10]. Unter Verwendung von Keras konnte eine Genauigkeit von 92,72 % ermittelt werden. Der vom neuronalen Netz berechnete Liveness Score wird im Backend der FR-Applikation zurückgegeben. Dieser betrug bei einem echten Gesicht durchschnittlich 98,96 %. Echte Gesichter fielen in einer dunklen Umgebung aufgrund fehlender Trainingsdaten jedoch auf 4 %. Sowohl 2D- als auch 3D-Spoofing-Angriffe wurden mit einer Lebendigkeit von 1 % bewertet. Da neben den genannten Trainingsdatenbanken ~60 Bilder pro Spoofing-Angriff verwendet wurden, fand eine weitere Überprüfung mittels der Falsch-Akzeptanz-Rate (FAR) sowie der Falsch-Rückweisungs-Rate (FRR) statt. Die FRR gibt den Prozentsatz der Identifikationsinstanzen, in denen befugte Personen fälschlicherweise abgelehnt werden an und die FAR ist die Wahrscheinlichkeit, dass das System einen Spoofing-Angriff fälschlicherweise authentifiziert [BPR00]. Die FAR beträgt 2 %, die FRR 34 % und die Equal Error Rate (EER) liegt somit bei 18%.

6 Zusammenfassung und Ausblick

Im Rahmen des vorliegenden Beitrags konnte gezeigt werden, dass unter Verwendung von Methoden aus dem Bereich der FLD die besonderen Anforderungen an die Gesichtserkennung im Webkontext umgesetzt werden können. Dazu wurden etablierte Methoden im Hinblick auf die Erfüllung der entsprechenden Anforderungen unter Verwendung eines erweiterten Technologieakzeptanzmodells evaluiert und für eine prototypische Entwicklung ausgewählt. Die Eignung der Liveness-Methoden konnte durch Implementierung einer lauffähigen Webanwendung nachgewiesen werden.

Im Rahmen der Evaluation des Prototyps hat sich gezeigt, dass die Binäre Klassifizierung mittels eines tiefen neuronalen Netzes als ausreichende Sicherheitsmaßnahme dienen kann. Die Genauigkeit der Binären Klassifizierung beträgt 92,72 %. Die FAR beträgt 2 %, die FRR 34 % und die EER 18 %. Dies bedeutet, dass Nutzer eher abgelehnt werden, als dass fälschlicherweise ein Spoofing-Angriff akzeptiert wird.

Die Anti-Spoofing-Verfahren werden ständig weiterentwickelt, allerdings gilt dies ebenfalls für die Spoofing-Mechanismen. Eine im Prototyp nicht umgesetzte Methode, das Face Flashing, wird aktuell optimiert und kann zukünftig voraussichtlich auch einen Schutz gegen Foto-, Video- und 3D-Spoofing-Angriffe in Web-Applikationen bereitstellen. Deren Eignung im Webkontext sollte demnach zukünftig geprüft werden, um eine weitere Optimierung des Prototyps zu erreichen.

7 Danksagung

Mein besonderer Dank gilt Prof. Dr. Peer Küppers vom Fachbereich Dienstleistungen und Consulting der Hochschule für Wirtschaft und Gesellschaft Ludwigshafen für die fachliche und persönliche Unterstützung meiner Arbeit.

Literaturverzeichnis

- [AŽŠ15] Adamiak, K.; Żurek, D; Ślot, K.: Liveness detection in remote biometrics based on gaze direction estimation. In (ed.): 2015 Federated Conference on Computer Science and Information Systems (FedCSIS). IEEE, S. 225-230, 2015.
- [BPR00] Bolle R. M.; Pankanti, S.; Ratha, N. K.: Evaluation techniques for biometrics-based authentication systems (FRR). In (ed.): International Conference on Pattern Recognition. Proceedings 15th, ICPR-2000, Barcelona, Spanien, S. 831-837, 2000.
- [Ca14] Cai, L. et al.: A Novel Face Spoofing Detection Method based on Gaze Estimation. In (ed.): Computer Vision -- ACCV 2014. Lecture Notes in Computer Science, Springer, Band 9005, 2014.
- [CK06] Cody-Allen, E.; Kishore, R.: An extension of the UTAUT model with e-quality, trust,

- and satisfaction constructs. In (ed.): SIGMIS CPR'06 - Proceedings of the 2006 ACM SIGMIS CPR Conference. ACM Press, Claremont, Kalifornien, S. 82-89, 2006.
- [EM13] Erdogmus, N.; Marcel, S.: Spoofing in 2D face recognition with 3D masks and anti-spoofing with Kinect. In (ed.): IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS). IEEE, S. 1-6, 2013.
- [GMF14] Galbally, J.; Marcel, S.; Fierrez, J.: Biometric Antispoofing Methods: A Survey in Face Recognition. In (ed.): IEEE Access, Band 2. IEEE, S. 1530-1552, 2014.
- [HA15] Hassaballah, M.; Aly, S.: Face recognition: challenges, achievements and future directions. In (ed.): IET Computer Vision, Band 9, Nr. 4. IET, S. 614-626, 2015.
- [JW09] Jeatrakul, P.; Wong, K. W.: Comparing the performance of different neural networks for binary classification problems. In (ed.): Eighth International Symposium on Natural Language Processing. IEEE, Bangkok, Thailand, S. 111-115, 2009.
- [KD12] Kähm, O.; Damer, N.: 2D Face Liveness Detection: an Overview. In (Hrsg., ed.): BIOSIG 2012. Gesellschaft für Informatik e.V., Bonn, S. 171-182, 2012.
- [Lo03] Lowe, D.: Requirements Eng. Band 8, S. 102-113, Springer, 2003.
- [Pl18] Pluta, W.: Clownsmaske besiegt Gesichtserkennung. Golem.de, 2018.
- [Ro18] Rosebrock, A.: Practical Python and OpenCV + Case Studies, An Introductory, Example Driven Guide to Image Processing and Computer Vision. S. 110-112, 2018.
- [Re17] Reportlinker: Facial Recognition Market by Component: Software Tools and Services, Technology, Use Case And Application Area - Global Forecast to 2021, 2017.
- [Ri17] Rinke, F.: Peking stellt ein ganzes Volk unter Beobachtung, RP ONLINE, 2017.
- [SBL15] Kim, S.; Ban, Y.; Lee, S.: Face Liveness Detection Using Defocus. In (ed.): Sensors 2015, Nummer 1. Sensors, S. 1537-1563, 2015.
- [SČ16] Soukupová, T.; Čech, J.: Real-Time Eye Blinking using Facial Landmarks. Prag, 2016.
- [SD14] Saptarshi, C.; Dhruvajyoti, D.: AN OVERVIEW OF FACE LIVENESS DETECTION. In (ed.): International Journal on Information Theory (IJIT), Band 3, Nummer 2. 2014.
- [Sh16] Shi, W et al.: Edge Computing: Vision and Challenges. In (ed.): IEEE Internet of Things Journal, Band 3, Nummer 5. IEEE, S. 637-646, 2016.
- [Ta10] Tan, X. et al.: Face Liveness Detection from A Single Image with Sparse Low Rank Bilinear Discriminative Model. In (ed.): Computer Vision – ECCV 2010. Lecture Notes in Computer Science, Springer, 2010.
- [Ta18] Tang, D. et al: Face Flashing: a Secure Liveness Detection Protocol based on Light Reflections, 2018.
- [Wa17] Wang, S. et al.: Face Liveness Detection Based on Skin Blood Flow Analysis. In (ed.): Symmetry, Band 9, Nummer 12. Symmetry, S. 305, 2017.
- [Wo19] Woll, J.: Foto reicht: 40% der Handys fallen bei Face-Unlock-Test komplett durch. WinFuture.de, 2019.
- [Yn18] YND: Anti Spoofing Mechanisms in Face Recognition based on DNN. Medium, 2018.

Latent representations of transaction network graphs in continuous vector spaces as features for money laundering detection

Dominik Wagner¹

Abstract: This paper explores the construction of network graphs from a large bank transaction dataset and draws from findings in language modeling and unsupervised learning to transform these graphs into multidimensional vector representations. Such latent representations encode relationships and community structures within the transaction network. Three classifiers with varying complexity are trained on these latent representations to detect suspicious behavior with respect to money laundering. The specific challenges accompanying highly imbalanced classes are discussed as well and two strategies to overcome these challenges are compared.

Keywords: feature learning; graph embeddings; latent representations; DeepWalk; anti-money laundering; language modeling; class imbalances; SMOTE; machine learning; support vector machine; naïve bayes; multilayer perceptron

1 Introduction

The process of money laundering can be divided into three stages: placement, layering and integration [Su01, p. 145]. During the placement stage, illegally obtained funds (mostly cash) are channeled into the financial system. The layering stage refers to the removal of traces and the distribution of those illegally obtained funds. Numerous transactions are carried out in this stage, often involving offshore-accounts and complex financial instruments. In the integration stage, the funds re-enter the legitimate economy, e. g. through the purchase of real estate or luxury assets. The three stages of the money laundering process show that a typical money laundering operation includes multiple transactions, which are transferred by a group of entities through a multitude of channels. Thereby, secret collusion takes place between the entities involved. The Financial Action Task Force (on Money Laundering) also names networks as one of the key characteristics of professional money laundering organizations. Networks are described as a collection of associates or contacts working together to facilitate money laundering schemes [Fi18, p. 13]. As money laundering involves groups of collaborating individuals, signs for money laundering may only be apparent when the collective behavior of these groups is examined.

¹ Technische Hochschule Nürnberg Georg Simon Ohm, Fakultät Informatik, Keßlerplatz 12, 90489 Nürnberg, Deutschland wagnerdo49883@th-nuernberg.de

2 Related Work

Considering the behavior of communities in a network to facilitate money laundering and fraud detection has produced promising results in several studies. Most of the related work, however, focuses on feature extraction with graph mining methods and metrics from Social Network Analysis (SNA). Furthermore, not all approaches are tested on real-world data.

For instance, Michalak and Korczak [MK11] propose a model for graph structure learning that can be trained on a previously annotated transaction graph and can be matched against a graph without annotated transactions. The model used in their method is parametrized using fuzzy numbers, which represent parameters of transactions. The authors perform their experiments with artificially-generated data.

Savage et al. [Sa17] generate network graphs from data provided by the Australian Transaction Reports and Analysis Centre (AUSTRAC). The authors apply a combination of network analysis and supervised learning (Support Vector Machine and Random Forest) to identify suspicious behavior indicative of money laundering activity. The study focuses on identifying small sets of interacting parties whose collective behavior is suspicious. Their model considers a range of demographic, network specific, transaction specific and time-dependent features derived from the extracted communities.

In [EH07], the authors develop three algorithms to uncover anomalies in graph structured data, particularly for fraud. The algorithms focus on different types of changes in the graph such as vertex or edge insertions. The algorithms are validated on both synthetic and real-world data. In [DF15] methods from SNA, such as centrality measures are used to investigate money laundering cases. Their algorithms are verified with random graph data.

However, to the author's knowledge, no bank transaction network graphs have been transformed with the DeepWalk algorithm [PAS14] so far. This work contributes by carrying out such transformations (Sect. 5) and by testing the effectiveness of the approach with three different supervised classification methods (Sect. 6). Most closely related is the work by Weber et al. [We18] who apply scalable Graph Convolutional Networks (GCN) [CMX18] to predict the degree of suspicion of a given target vertex in a transaction network and to identify other potential bad actors in the network via direct or indirect connections to vertices known to be suspicious. The graph structure is generated by a simulator based on transaction distributions and dynamics observed in real data.

3 Problem Definition

The data used in this work is a subset of real account transactions at a German retail bank. The majority of those entities are individual persons and a small portion are legal persons. The dataset contains 241 exemplary entities, which have been flagged as suspicious regarding money laundering. The suspicion refers to Suspicious Activity Reports (SARs)

made by the bank. SARs are filed to the Financial Intelligence Unit after compliance officers have confirmed the suspicion in a multi-step process.

The data can be represented as an undirected graph $G(V, E)$ where V is a set of vertices and E is a set of edges between the vertices with $E \subseteq \{\{u, v\} \mid u, v \in V\}$. A vertex v_i constitutes either a bank client or a country described by a two-letter country code, as defined in ISO 3166-1 alpha-2 [In19]. V is further divided into two sets of vertices. The vertices V_M for which the correct labels are known and the vertices V_N whose labels have to be determined. The task is to label the vertices $Y_i \in V_N$ with one of the two labels $\mathcal{L} = \{L_0, L_1\}$ where L_0 represents unsuspicious entities and L_1 represents suspicious entities with respect to money laundering. y_i is the label of vertex Y_i . The classification task is discussed in Sect. 6.

The information about the dependence of the examples embedded in the structure of G is utilized to classify the vertices V_N . The goal in that regard is to build multidimensional representations $X_E \in \mathbb{R}^{|V| \times k}$ where k is the number of latent dimensions. The process of creating these multidimensional representations is discussed in Sect. 5.

4 Graph Construction

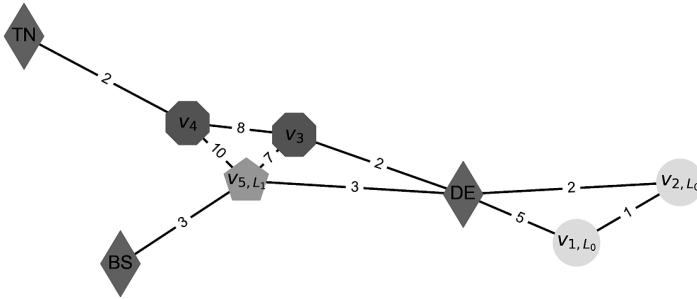


Fig. 1: Transaction graph example

Fig. 1 shows an example of a partially labeled transaction network graph $G(V, E)$ with $|V| = 8$ and $|E| = 10$. The vertices v_1 and v_2 are labeled as unsuspicious (L_0), while the vertex v_5 is labeled as suspicious (L_1) regarding money laundering. The vertices DE , BS , TN represent financial institutions in Germany (DE), The Bahamas (BS) and Tunisia (TN). The vertices v_3 and v_4 are not labeled yet. The edge attributes represent the number of transactions between the two vertices. The graph structure shows that the already suspicious vertex v_5 transacts with a country known to have strategic anti-money laundering deficiencies (BS) [Fi19]. The unlabeled entities v_3 , v_4 interact primarily with v_5 and with each other. The vertex v_4 also transacts with a high risk country (TN). From these observations one could conclude that the vertices v_3 and v_4 should also be labeled as suspicious ($y_3 = L_1$ and $y_4 = L_1$).

In order to establish a transaction network graph $G(V, E)$ as illustrated in Fig. 1, the two counterparts u and v of each transaction are examined. In case the account is an internal account, i. e. it refers to an entity that is also available in the dataset, the unique client identifier is added as a vertex to the network, provided that the identifier in question does not already exist in the network. In case the account is an external account, the target or origin country of the transaction is extracted from the SWIFT Code. SWIFT (Society for Worldwide Interbank Financial Telecommunication) Codes are a standard format of Business Identifier Codes defined in ISO 9362 [SW19]. In this way, external accounts are aggregated based on the country they are located in. Each edge $e \in E$ represents at least one transaction between two connecting vertices u and v . The final graph contains $|V| = 1,128,921$ vertices and $|E| = 1,489,209$ edges.

5 Latent Representations

The graph structure is transformed into latent multidimensional representations by means of the DeepWalk algorithm [PAS14]. DeepWalk is an unsupervised learning algorithm that learns a latent representation of each vertex in a graph with the Skip-gram language model [Mi13a; Mi13b]. The graph is explored through a sequence of random walks. These random walk sequences are equivalent to sentences in language modeling. The learned vector representations encode relationships and community structures and can be used for various purposes such as classification, clustering or similarity search.

The Skip-gram model is built from a neural network with one hidden layer. Applied within the DeepWalk framework, the model receives a vertex v_i and generates the probability for all other vertices that a given vertex appears in the neighborhood or context of the input vertex v_i . The context is defined by the window c .

Learning a latent representation means learning a mapping function from vertex co-occurrences: $\Phi : v \in V \mapsto \mathbb{R}^{|V| \times k}$. The mapping Φ is the latent representation of each vertex $v \in V$. The mapping Φ is represented by a $|V| \times k$ matrix, which is the target matrix X_E . The objective of DeepWalk is to find latent representations that are useful for predicting the surrounding vertices in a random walk sequence. The optimization problem is formalized as follows:

$$\min_{\Phi} -\log Pr(\{v_{i-c}, \dots, v_{i+c}\} \setminus v_i \mid \Phi(v_i)) \quad (1)$$

The Skip-gram model is used to update the representations according to the objective function Eq. (1). Skip-gram approximates the conditional probability in Eq. (1) by:

$$Pr(\{v_{i-c}, \dots, v_{i+c}\} \setminus v_i \mid \Phi(v_i)) = \prod_{\substack{j=i-c \\ j \neq i}}^{i+c} Pr(v_j \mid \Phi(v_i)) \quad (2)$$

5.1 Implementation

DeepWalk receives the following parameters: window size c , embedding size k , number of random walks per vertex γ and walk length t . It returns a matrix of vertex representations $\Phi \in \mathbb{R}^{|V| \times k}$. The algorithm iterates over all possible collocations of a given random walk sequence R_{v_i} that appear within the window c . R_{v_i} is a random walk sequence rooted at vertex v_i with length t . A window of length $2c + 1$ is slid over the random walk R_{v_i} , mapping the central vertex in the window v_f to its representation ($\Phi(v_f) \in \mathbb{R}^k$). Given the representation of v_f , the task is to maximize the probability of its neighbors in the random walk sequence. Such a posterior distribution is approximated using hierarchical softmax [MB05]. Hierarchical softmax is a computationally efficient approximation of the full softmax, where the graph vertices are assigned to the leaves of a balanced binary tree, turning the problem into maximizing the probability of a specific path in the tree hierarchy. The path to a neighboring vertex v_n of the central vertex v_f is identified by a sequence of tree nodes ($a_0, a_1, \dots, a_{\lceil \log |V| \rceil}$ where a_0 is the root of the tree and $a_{\lceil \log |V| \rceil} = v_n$). The conditional probability is then approximated by:

$$Pr(v_n | \Phi(v_f)) = \prod_{j=1}^{\lceil \log |V| \rceil} Pr(a_j | \Phi(v_f)) \quad (3)$$

$p(a_j | \Phi(v_f))$ is now calculated by a binary classifier assigned to the parent of the node a_j :

$$Pr(a_j | \Phi(v_f)) = 1 / (1 + e^{-\Phi(v_f) \cdot \Psi(a_j)}) \quad (4)$$

$\Psi(a_j) \in \mathbb{R}^k$ is the vertex representation at the parent node of a_j . Stochastic gradient descent (SGD) is used to optimize the model parameter set ($\theta = \{\Phi, \Psi\}$). DeepWalk learns a latent

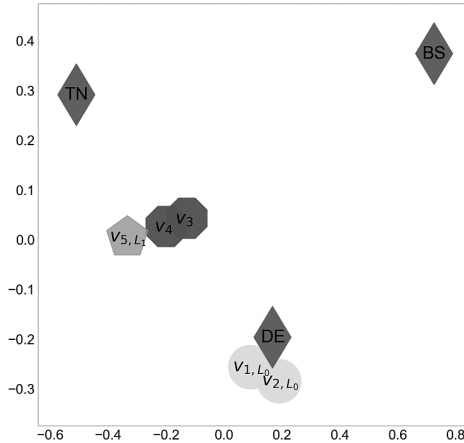


Fig. 2: Latent representation of the example graph

representation of transactions in \mathbb{R}^k . In Fig. 2 the method is used on the example graph from

Fig. 1 to generate a latent representation in \mathbb{R}^2 . The community structure in the input graph corresponds well with the embedding. The unlabeled vertices v_3 and v_4 are located close to v_5 , which has been already labeled as suspicious. The vertices labeled as unsuspicious v_1 and v_2 are grouped close to financial institutions in Germany (*DE*) and are relatively far away from v_3 , v_4 and v_5 . With standard classification methods applied on the latent representations, v_3 and v_4 would be more likely classified as suspicious (L_1).

The DeepWalk algorithm is trained on the full bank transaction dataset with $\gamma = 1000$ walks per vertex, walk length $t = 8$, window size $c = 2$, learning rate of the neural network $\alpha = 10^{-2}$ and $k \in \mathbb{R}^{20}$ hidden units (latent dimensions).

6 Classification

The learned representations are used to classify entities into the categories unsuspicious (L_0) and suspicious (L_1). Under the assumption that “suspicious” representations are sufficiently different from “unsuspicious” ones, supervised learning algorithms are used for this task. The class L_1 refers to Suspicious Activity Reports (SARs) made by the bank, while the class L_0 refers to hitherto inconspicuous parties.

The latent representations serve as inputs for three widely used classifiers with varying complexity: Naive Bayes (NB), Support Vector Machine (SVM) and Multilayer Perceptron (MLP). The standard variant of Naive Bayes, in which the likelihood of the features is assumed to be Gaussian, is applied here. SVMs are trained with both a linear and a Radial Basis Function (RBF) kernel. The optimal SVM hyperparameters are determined by evaluating the area under the ROC curve [Fa06; Sw01] in a threefold cross-validation on the training set, using the grid search technique. The kernel parameter γ for the RBF kernel is selected from the set $\gamma \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\} \subset \mathbb{R}_{>0}$. The linear SVM employs, the ℓ^2 penalty on a squared hinge loss. The penalty parameter of the error term C is selected from the set $C \in \{10^k \mid k = -2, -1, \dots, 5\} \subset \mathbb{R}_{>0}$ for the linear and the RBF kernel. Several

Hyperparameter	Value
Optimization	Stochastic Gradient Descent
Learning Rate	$\alpha = 10^{-4}$
Learning Rate Decay	$d = 10^{-6}$
Momentum	Nesterov ($m = 0.9$)
Dropout	$p = 0.2$
Activation Function	$\text{leakyReLU } f(x) = \begin{cases} x, & \text{if } x > 0 \\ 0.3x, & \text{else} \end{cases}$

Tab. 1: MLP hyperparameters

architectures and hyperparameters have been considered to train the MLP. The best results were achieved with three hidden layers and 150 hidden units in each layer combined with the hyperparameters shown in Tab. 1. The output layer consists of a single neuron and employs

a sigmoid activation function. The training samples are processed in mini-batches of 32 samples. The training procedure runs for 20 epochs, i. e. the number of complete passes through the training dataset is 20.

6.1 Class Imbalances and Synthetic Over-sampling

The dataset at hand is highly imbalanced, i. e. the classification categories are not at all equally represented. Vertex representations were obtainable for 223 of the 241 entities flagged as suspicious. For the remaining 18 entities, the data provided in the subset was insufficient to generate embeddings, i. e. no transactions were available for these entities. On the other hand, more than one million vertex representations encoding regular client behavior were generated.

To overcome this problem, two strategies are considered and compared afterwards. First, the majority class instances (i. e. unsuspicious entities) are under-sampled and the minority class instances are randomly duplicated until the desired share of suspicious samples to total samples is reached. Second, the majority class instances (i. e. unsuspicious entities) are under-sampled and the minority class instances are over-sampled with synthetic examples by means of the SMOTE algorithm [Bo11]. Both strategies use subsets from the 1,128,921 available latent representations. The processed training data consist of 12,000 “unsuspicious” records and 12,000 “suspicious” records. The test data consist of 3,000 “unsuspicious” and 45 “suspicious” records.

The second approach to overcome the class imbalance problem is inspired by the work of Xu et al. [Xu15], who use the SMOTE algorithm as an over-sampling method to produce balanced training sets from word embeddings. This work proceeds as follows: First, 12,000 samples are randomly drawn from the majority class (“unsuspicious”). Second, the SMOTE algorithm is used, to generate samples for the minority class (“suspicious”), until the training set is fully balanced. The algorithm attempts to over-sample the minority class by taking each minority class sample and generates synthetic examples from the k minority class nearest neighbors. The neighbor parameter is set to $k = 5$ here. The training pipeline has five

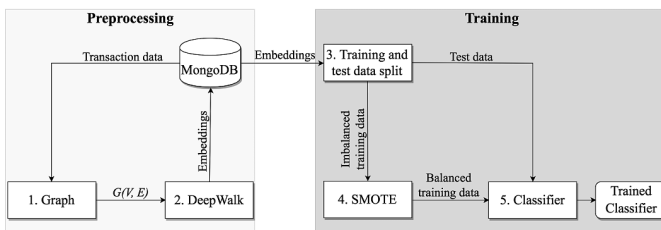


Fig. 3: Preprocessing and training pipeline with SMOTE

main steps, which are illustrated in Fig. 3. The first step constructs the transaction graph as discussed in Sect. 4. The second step performs the DeepWalk algorithm according to Sect. 5

and stores the latent representations in a database. The first two steps are preprocessing components, which have to be executed at least once before the actual training procedure can begin. The training procedure starts with the third step, which splits the vertex embeddings retrieved from the database into a training set and a test set. The fourth step applies the SMOTE algorithm on the minority class samples of the training data. The final step trains a classifier on the data.

Principal component analysis (PCA) is applied for the purpose of visualizing the learned 20-dimensional embeddings in two-dimensional space [TB99]. Fig. 4 shows the PCA projection of vertex representations from the majority class “unsuspicious” and the minority class “suspicious” without over-sampling in the left diagram. In contrast, the PCA projection of vertex representations from the majority class “unsuspicious” and the minority class “suspicious” with synthetic over-sampling of the minority class is shown in the right diagram. The newly generated samples are denoted by rhombuses. The right diagram shows better formed clusters for the “suspicious” class compared to the left diagram. Furthermore, the newly generated samples in the right diagram are usually close to the original samples, indicating that over-sampling with the SMOTE algorithm generally maintains the same distribution as the original samples.

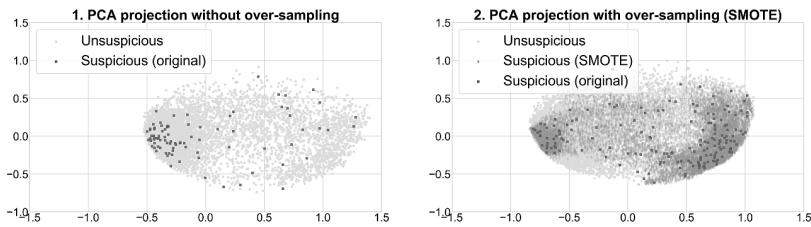


Fig. 4: PCA projection with and without over-sampling

6.2 Results

Each classifier is trained ten times on different subsets of the data. The samples from the majority class are randomly drawn prior to each new training procedure. The unprocessed minority class samples are shuffled and split into training data and test data prior to each new training procedure. Both over-sampling strategies are only applied on minority class samples of the training data. Hence, the 45 minority class samples used for performance evaluation are not synthetically generated or duplicated. Each input vector is scaled individually such that the ℓ^2 -norm equals one. The performance measures are averaged over all ten training instances.

In the presence of imbalanced datasets, it is preferable to use Receiver Operating Characteristic (ROC) curves or other similar techniques over traditional performance measures such as accuracy [PF01]. ROC curves indicate the performance of binary classifiers at various false positive rates [Fa06; Sw01]. Fig. 5 illustrates the average ROC curves for both sampling

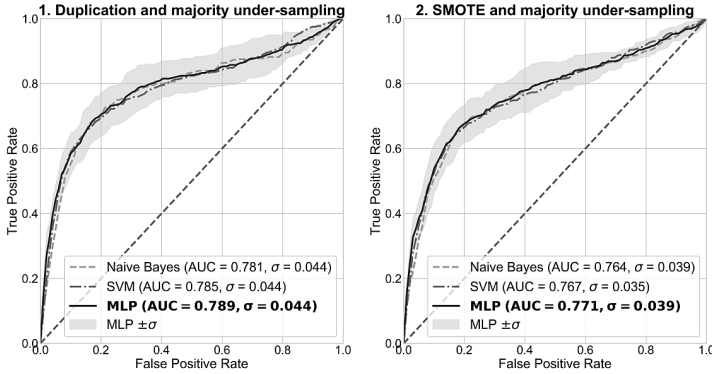


Fig. 5: Average ROC curves for different strategies and classifiers

strategies and the three classifiers. The dashed line $f(x) = x$ represents the scenario of randomly guessing the class. Despite the varying complexity of the three classifiers, the performance difference between them is relatively small for both strategies. The best overall results produces the MLP in combination with random duplication of the minority class. The average Area under the ROC Curve (AUC) equals 0.789. All three classifiers achieve better average performance, when majority under-sampling is applied in combination with random duplication (left diagram). Naive Bayes achieves an average $AUC = 0.781$ with random duplication of the minority class samples and an average $AUC = 0.764$ with synthetic over-sampling. The SVM classifier delivers an average $AUC = 0.785$ with random duplication and an average $AUC = 0.767$ with synthetic over-sampling.

However, the standard deviations of the AUC scores are lower for all three classifiers, when majority under-sampling is applied in combination with the SMOTE algorithm, indicating that this strategy produces more stable results compared to majority under-sampling combined with random duplication. The largest difference in the standard deviation of the AUC score shows the SVM classifier with $\sigma = 0.044$ for the strategy including random duplication (left diagram) and $\sigma = 0.035$ for the strategy including the SMOTE algorithm (right diagram). The shaded area in Fig. 5 illustrates the development of σ around the ROC curve of the MLP.

7 Discussion and Future Work

The approach presented in this paper has only been tested on a simple edge creation strategy, which was based on the existence of at least one transaction between two entities. Many other strategies are imaginable here. For instance, edges could be created only for certain types of transactions such as cash withdrawals, transactions to or from high risk countries or transactions made by politically exposed persons (PEP). Another strategy would be to create an edge only when certain transaction amount thresholds are exceeded.

The DeepWalk algorithm lacks generalizability. Whenever a new vertex is added, the model has to be re-trained in order to represent the new vertex as an embedding. The approach is therefore not suitable for real-time transaction monitoring systems. DeepWalk is also limited to the structure immediately around a vertex. This local focus implicitly ignores long-distance relationships and the learned representations might not uncover important global structural patterns. Furthermore, DeepWalk relies on stochastic gradient descent optimization, which can become stuck in local minima and consequently lead to poor results.

A comprehensive empirical study conducted by Khosla et al. [KAS19] considers 9 popular unsupervised network representation learning (UNRL) approaches including DeepWalk and Graph Convolutional Networks (GCN). The authors perform vertex classification and edge prediction tasks on 11 real-world datasets with varying structural properties. They find that there is no single method superior to the others and that the choice of a suitable method is determined by certain properties of the embedding methods, the task and structural properties of the underlying graph. Given these findings, other UNRL techniques should be applied to bank transaction network graphs in order to assess whether further improvements in the quality of the vertex embeddings are possible.

8 Conclusion

This paper demonstrates the transformation of transaction network graphs into real-valued latent representations with the DeepWalk algorithm. By doing so, an alternative way of representing bank transactions and the relationships between the entities involved is illustrated. Experiments on the latent representations generated from real bank transaction data show the effectiveness of the approach on a binary classification task.

Strong class imbalances pose an extra challenge to machine learning systems. The small number of “suspicious” records compared to the very large number of “unsuspicious” records requires additional feature engineering efforts. This paper explores two different strategies to cope with such imbalances. The more stable strategy for the data at hand is a combination of majority under-sampling and minority over-sampling with the SMOTE algorithm, while the combination of majority under-sampling and random duplication of the minority class achieves better results in terms of the average *AUC* score.

Acknowledgments

The results presented in this study are part of the development of a machine learning based prototype for money laundering detection during a 20 week internship, which undergraduate students at TH Nürnberg take in their fifth semester. The author would like to thank Wolfgang Füssel and Torsten Hoffmann for their support and numerous helpful remarks during this project. Thanks should also go to Prof. Dr. Jens Albrecht for the comprehensive feedback on this paper.

References

- [Bo11] Bowyer, K. W.; Chawla, N. V.; Hall, L. O.; Kegelmeyer, W. P.: SMOTE: Synthetic Minority Over-sampling Technique. CoRR abs/1106.1813/, 2011, arXiv: 1106.1813, URL: <http://arxiv.org/abs/1106.1813>.
- [CMX18] Chen, J.; Ma, T.; Xiao, C.: FastGCN: Fast Learning with Graph Convolutional Networks via Importance Sampling. CoRR abs/1801.10247/, 2018, arXiv: 1801.10247, URL: <http://arxiv.org/abs/1801.10247>.
- [DF15] Drezewski Rafał and Sepielak, J.; Filipkowski, W.: The Application of Social Network Analysis Algorithms in a System Supporting Money Laundering Detection. Inf. Sci. 295/C, pp. 18–32, Feb. 2015, ISSN: 0020-0255, URL: <https://doi.org/10.1016/j.ins.2014.10.015>.
- [EH07] Eberle, W.; Holder, L.: Anomaly Detection in Data Represented As Graphs. Intell. Data Anal. 11/6, pp. 663–689, Dec. 2007, ISSN: 1088-467X, URL: <http://dl.acm.org/citation.cfm?id=1368018.1368024>.
- [Fa06] Fawcett, T.: An introduction to ROC analysis. Pattern Recognition Letters 27/8, ROC Analysis in Pattern Recognition, pp. 861–874, 2006, ISSN: 0167-8655, URL: <http://www.sciencedirect.com/science/article/pii/S016786550500303X>.
- [Fi18] Financial Action Task Force: Professional Money Laundering, Financial Action Task Force, 2018, URL: <http://www.fatf-gafi.org/media/fatf/documents/Professional-Money-Laundering.pdf>, visited on: 03/17/2019.
- [Fi19] Financial Action Task Force: Improving Global AML/CFT Compliance, Financial Action Task Force, 2019, URL: <https://www.fatf-gafi.org/publications/high-risk-and-other-monitored-jurisdictions/documents/fatf-compliance-february-2019.html>, visited on: 05/10/2019.
- [In19] International Organization for Standardization: Country Codes - ISO 3166, 2019, URL: <https://www.iso.org/iso-3166-country-codes.html>, visited on: 05/10/2019.
- [KAS19] Khosla, M.; Anand, A.; Setty, V.: A Comprehensive Comparison of Unsupervised Network Representation Learning Methods. CoRR abs/1903.07902/, 2019, arXiv: 1903.07902, URL: <http://arxiv.org/abs/1903.07902>.
- [MB05] Morin, F.; Bengio, Y.: Hierarchical Probabilistic Neural Network Language Model. In (Cowell, R. G.; Ghahramani, Z., eds.): Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics. Society for Artificial Intelligence and Statistics, pp. 246–252, 2005, URL: <http://www.iro.umontreal.ca/~lisa/pointeurs/hierarchical-nnlm-aistats05.pdf>.
- [Mi13a] Mikolov, T.; Chen, K.; Corrado, G.; Dean, J.: Efficient Estimation of Word Representations in Vector Space. CoRR abs/1301.3781/, 2013, arXiv: 1301.3781, URL: <http://arxiv.org/abs/1301.3781>.

- [Mi13b] Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; Dean, J.: Distributed Representations of Words and Phrases and their Compositionality. In (Burgess, C. J. C.; Bottou, L.; Welling, M.; Ghahramani, Z.; Weinberger, K. Q., eds.): *Advances in Neural Information Processing Systems 26*. Curran Associates, Inc., pp. 3111–3119, 2013, URL: <http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf>.
- [MK11] Michalak, K.; Korczak, J. J.: Graph mining approach to suspicious transaction detection. *2011 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pp. 69–75, 2011.
- [PAS14] Perozzi, B.; Al-Rfou, R.; Skiena, S.: DeepWalk: Online Learning of Social Representations. In: *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '14*, ACM, New York, New York, USA, pp. 701–710, 2014, ISBN: 978-1-4503-2956-9, URL: <http://doi.acm.org/10.1145/2623330.2623732>.
- [PF01] Provost, F.; Fawcett, T.: Robust Classification for Imprecise Environments. *Mach. Learn.* 42/3, pp. 203–231, Mar. 2001, ISSN: 0885-6125, URL: <http://dx.doi.org/10.1023/A:1007601015854>.
- [Sa17] Savage, D.; Zhang, X.; Wang, Q.; Yu, X.; Chou, P.: Detection of Money Laundering Groups: Supervised Learning on Small Networks. In: *The AAAI-17 Workshop on AI and Operations Research for Social Good*. Vol. WS-17-01, Association for the Advancement of Artificial Intelligence, pp. 43–49, 2017.
- [Su01] Suendorf, U.: *Geldwäsche - Eine kriminologische Untersuchung*. Hermann Luchterhand Verlag GmbH, 2001.
- [Sw01] Swets, J. A.: Measuring the Accuracy of Diagnostic Systems. *Science* 240/4857, pp. 1285–1293, 2001, ISSN: 0036-8075, URL: <https://doi.org/10.1126/science.3287615>.
- [SW19] SWIFT: Data Standards, 2019, URL: <https://www.swift.com/standards/data-standards/bic>, visited on: 05/10/2019.
- [TB99] Tipping, M. E.; Bishop, C. M.: Mixtures of Probabilistic Principal Component Analyzers. *Neural Computation* 11/2, pp. 443–482, 1999, URL: <https://doi.org/10.1162/089976699300016728>.
- [We18] Weber, M.; Chen, J.; Suzumura, T.; Pareja, A.; Ma, T.; Kanezashi, H.; Kaler, T.; Leiserson, C. E.; Schardl, T. B.: Scalable Graph Learning for Anti-Money Laundering: A First Look. *CoRR abs/1812.00076*, 2018, arXiv: 1812.00076, URL: <http://arxiv.org/abs/1812.00076>.
- [Xu15] Xu, R.; Chen, T.; Xia, Y.; Lu, Q.; Liu, B.; Wang, X.: Word Embedding Composition for Data Imbalances in Sentiment and Emotion Classification. *Cognitive Computation* 7/2, pp. 226–240, 2015, URL: <https://doi.org/10.1007/s12559-015-9319-y>.

GI-Edition Lecture Notes in Informatics

- P-1 Gregor Engels, Andreas Oberweis, Albert Zündorf (Hrsg.): Modellierung 2001.
- P-2 Mikhail Godlevsky, Heinrich C. Mayr (Hrsg.): Information Systems Technology and its Applications, ISTA'2001.
- P-3 Ana M. Moreno, Reind P. van de Riet (Hrsg.): Applications of Natural Language to Information Systems, NLDB'2001.
- P-4 H. Wörn, J. Mühling, C. Vahl, H.-P. Meinzer (Hrsg.): Rechner- und sensor-gestützte Chirurgie; Workshop des SFB 414.
- P-5 Andy Schürr (Hg.): OMER – Object-Oriented Modeling of Embedded Real-Time Systems.
- P-6 Hans-Jürgen Appelpath, Rolf Beyer, Uwe Marquardt, Heinrich C. Mayr, Claudia Steinberger (Hrsg.): Unternehmen Hochschule, UH'2001.
- P-7 Andy Evans, Robert France, Ana Moreira, Bernhard Rumpe (Hrsg.): Practical UML-Based Rigorous Development Methods – Countering or Integrating the extremists, pUML'2001.
- P-8 Reinhard Keil-Slawik, Johannes Magenheimer (Hrsg.): Informatikunterricht und Medienbildung, INFOS'2001.
- P-9 Jan von Knop, Wilhelm Haverkamp (Hrsg.): Innovative Anwendungen in Kommunikationsnetzen, 15. DFN Arbeits-tagung.
- P-10 Mirjam Minor, Steffen Staab (Hrsg.): 1st German Workshop on Experience Management: Sharing Experiences about the Sharing Experience.
- P-11 Michael Weber, Frank Kargl (Hrsg.): Mobile Ad-Hoc Netzwerke, WMAN 2002.
- P-12 Martin Glinz, Günther Müller-Luschnat (Hrsg.): Modellierung 2002.
- P-13 Jan von Knop, Peter Schirmbacher und Viljan Mahni_ (Hrsg.): The Changing Universities – The Role of Technology.
- P-14 Robert Tolksdorf, Rainer Eckstein (Hrsg.): XML-Technologien für das Semantic Web – XSW 2002.
- P-15 Hans-Bernd Bludau, Andreas Koop (Hrsg.): Mobile Computing in Medicine.
- P-16 J. Felix Hampe, Gerhard Schwabe (Hrsg.): Mobile and Collaborative Business 2002.
- P-17 Jan von Knop, Wilhelm Haverkamp (Hrsg.): Zukunft der Netze –Die Verletz-barkeit meistern, 16. DFN Arbeitstagung.
- P-18 Elmar J. Sinz, Markus Plaha (Hrsg.): Modellierung betrieblicher Informations-systeme – MobIS 2002.
- P-19 Sigrid Schubert, Bernd Reusch, Norbert Jesse (Hrsg.): Informatik bewegt – Infor-matik 2002 – 32. Jahrestagung der Gesell-schaft für Informatik e.V. (GI) 30.Sept.-3. Okt. 2002 in Dortmund.
- P-20 Sigrid Schubert, Bernd Reusch, Norbert Jesse (Hrsg.): Informatik bewegt – Infor-matik 2002 – 32. Jahrestagung der Gesell-schaft für Informatik e.V. (GI) 30.Sept.-3. Okt. 2002 in Dortmund (Ergänzungs-band).
- P-21 Jörg Desel, Mathias Weske (Hrsg.): Promise 2002: Prozessorientierte Metho-den und Werkzeuge für die Entwicklung von Informationssystemen.
- P-22 Sigrid Schubert, Johannes Magenheimer, Peter Hubwieser, Torsten Brinda (Hrsg.): Forschungsbeiträge zur "Didaktik der Informatik" – Theorie, Praxis, Evaluation.
- P-23 Thorsten Spitta, Jens Borchers, Harry M. Sneed (Hrsg.): Software Management 2002 – Fortschritt durch Beständigkeit
- P-24 Rainer Eckstein, Robert Tolksdorf (Hrsg.): XMIDX 2003 – XML-Technologien für Middleware – Middle-ware für XML-Anwendungen
- P-25 Key Pousttchi, Klaus Turowski (Hrsg.): Mobile Commerce – Anwendungen und Perspektiven – 3. Workshop Mobile Commerce, Universität Augsburg, 04.02.2003
- P-26 Gerhard Weikum, Harald Schöning, Erhard Rahm (Hrsg.): BTW 2003: Daten-banksysteme für Business, Technologie und Web
- P-27 Michael Kroll, Hans-Gerd Lipinski, Kay Melzer (Hrsg.): Mobiles Computing in der Medizin
- P-28 Ulrich Reimer, Andreas Abecker, Steffen Staab, Gerd Stumme (Hrsg.): WM 2003: Professionelles Wissensmanagement – Er-fahrungen und Visionen
- P-29 Antje Düsterhöft, Bernhard Thalheim (Eds.): NLDB'2003: Natural Language Processing and Information Systems
- P-30 Mikhail Godlevsky, Stephen Liddle, Heinrich C. Mayr (Eds.): Information Systems Technology and its Applications
- P-31 Arslan Brömme, Christoph Busch (Eds.): BIOSIG 2003: Biometrics and Electronic Signatures

- P-32 Peter Hubwieser (Hrsg.): Informatische Fachkonzepte im Unterricht – INFOS 2003
- P-33 Andreas Geyer-Schulz, Alfred Taudes (Hrsg.): Informationswirtschaft: Ein Sektor mit Zukunft
- P-34 Klaus Dittrich, Wolfgang König, Andreas Oberweis, Kai Rannenber, Wolfgang Wahlster (Hrsg.): Informatik 2003 – Innovative Informatikanwendungen (Band 1)
- P-35 Klaus Dittrich, Wolfgang König, Andreas Oberweis, Kai Rannenber, Wolfgang Wahlster (Hrsg.): Informatik 2003 – Innovative Informatikanwendungen (Band 2)
- P-36 Rüdiger Grimm, Hubert B. Keller, Kai Rannenber (Hrsg.): Informatik 2003 – Mit Sicherheit Informatik
- P-37 Arndt Bode, Jörg Desel, Sabine Rathmayer, Martin Wessner (Hrsg.): DeLFI 2003: e-Learning Fachtagung Informatik
- P-38 E.J. Sinz, M. Plaha, P. Neckel (Hrsg.): Modellierung betrieblicher Informationssysteme – MobIS 2003
- P-39 Jens Nedon, Sandra Frings, Oliver Göbel (Hrsg.): IT-Incident Management & IT-Forensics – IMF 2003
- P-40 Michael Rebstock (Hrsg.): Modellierung betrieblicher Informationssysteme – MobIS 2004
- P-41 Uwe Brinkschulte, Jürgen Becker, Dietmar Fey, Karl-Erwin Großpietsch, Christian Hochberger, Erik Maehle, Thomas Runkler (Edts.): ARCS 2004 – Organic and Pervasive Computing
- P-42 Key Pousttchi, Klaus Turowski (Hrsg.): Mobile Economy – Transaktionen und Prozesse, Anwendungen und Dienste
- P-43 Birgitta König-Ries, Michael Klein, Philipp Obreiter (Hrsg.): Persistence, Scalability, Transactions – Database Mechanisms for Mobile Applications
- P-44 Jan von Knop, Wilhelm Haverkamp, Eike Jessen (Hrsg.): Security, E-Learning, E-Services
- P-45 Bernhard Rumpe, Wolfgang Hesse (Hrsg.): Modellierung 2004
- P-46 Ulrich Flegel, Michael Meier (Hrsg.): Detection of Intrusions of Malware & Vulnerability Assessment
- P-47 Alexander Prosser, Robert Krimmer (Hrsg.): Electronic Voting in Europe – Technology, Law, Politics and Society
- P-48 Anatoly Doroshenko, Terry Halpin, Stephen W. Liddle, Heinrich C. Mayr (Hrsg.): Information Systems Technology and its Applications
- P-49 G. Schiefer, P. Wagner, M. Morgenstern, U. Rickert (Hrsg.): Integration und Datensicherheit – Anforderungen, Konflikte und Perspektiven
- P-50 Peter Dadam, Manfred Reichert (Hrsg.): INFORMATIK 2004 – Informatik verbindet (Band 1) Beiträge der 34. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 20.-24. September 2004 in Ulm
- P-51 Peter Dadam, Manfred Reichert (Hrsg.): INFORMATIK 2004 – Informatik verbindet (Band 2) Beiträge der 34. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 20.-24. September 2004 in Ulm
- P-52 Gregor Engels, Silke Seehusen (Hrsg.): DELFI 2004 – Tagungsband der 2. e-Learning Fachtagung Informatik
- P-53 Robert Giegerich, Jens Stoye (Hrsg.): German Conference on Bioinformatics – GCB 2004
- P-54 Jens Borchers, Ralf Kneuper (Hrsg.): Softwaremanagement 2004 – Outsourcing und Integration
- P-55 Jan von Knop, Wilhelm Haverkamp, Eike Jessen (Hrsg.): E-Science und Grid Ad-hoc-Netze Medienintegration
- P-56 Fernand Feltz, Andreas Oberweis, Benoit Otjacques (Hrsg.): EMISA 2004 – Informationssysteme im E-Business und E-Government
- P-57 Klaus Turowski (Hrsg.): Architekturen, Komponenten, Anwendungen
- P-58 Sami Beydeda, Volker Gruhn, Johannes Mayer, Ralf Reussner, Franz Schweiggert (Hrsg.): Testing of Component-Based Systems and Software Quality
- P-59 J. Felix Hampe, Franz Lehner, Key Pousttchi, Kai Rannenber, Klaus Turowski (Hrsg.): Mobile Business – Processes, Platforms, Payments
- P-60 Steffen Friedrich (Hrsg.): Unterrichtskonzepte für informatische Bildung
- P-61 Paul Müller, Reinhard Gotzhein, Jens B. Schmitt (Hrsg.): Kommunikation in verteilten Systemen
- P-62 Federrath, Hannes (Hrsg.): „Sicherheit 2005“ – Sicherheit – Schutz und Zuverlässigkeit
- P-63 Roland Kaschek, Heinrich C. Mayr, Stephen Liddle (Hrsg.): Information Systems – Technology and its Applications

- P-64 Peter Liggesmeyer, Klaus Pohl, Michael Goedicke (Hrsg.): Software Engineering 2005
- P-65 Gottfried Vossen, Frank Leymann, Peter Lockemann, Wolfrid Stucky (Hrsg.): Datenbanksysteme in Business, Technologie und Web
- P-66 Jörg M. Haake, Ulrike Lucke, Djamshid Tavangarian (Hrsg.): DeLFI 2005: 3. deutsche e-Learning Fachtagung Informatik
- P-67 Armin B. Cremers, Rainer Manthey, Peter Martini, Volker Steinhage (Hrsg.): INFORMATIK 2005 – Informatik LIVE (Band 1)
- P-68 Armin B. Cremers, Rainer Manthey, Peter Martini, Volker Steinhage (Hrsg.): INFORMATIK 2005 – Informatik LIVE (Band 2)
- P-69 Robert Hirschfeld, Ryszard Kowalczyk, Andreas Polze, Matthias Weske (Hrsg.): NODe 2005, GSEM 2005
- P-70 Klaus Turowski, Johannes-Maria Zaha (Hrsg.): Component-oriented Enterprise Application (COAE 2005)
- P-71 Andrew Torda, Stefan Kurz, Matthias Rarey (Hrsg.): German Conference on Bioinformatics 2005
- P-72 Klaus P. Jantke, Klaus-Peter Fähnrich, Wolfgang S. Wittig (Hrsg.): Marktplatz Internet: Von e-Learning bis e-Payment
- P-73 Jan von Knop, Wilhelm Haverkamp, Eike Jessen (Hrsg.): "Heute schon das Morgen sehen"
- P-74 Christopher Wolf, Stefan Lucks, Po-Wah Yau (Hrsg.): WEWoRC 2005 – Western European Workshop on Research in Cryptology
- P-75 Jörg Desel, Ulrich Frank (Hrsg.): Enterprise Modelling and Information Systems Architecture
- P-76 Thomas Kirste, Birgitta König-Ries, Key Pousttchi, Klaus Turowski (Hrsg.): Mobile Informationssysteme – Potentiale, Hindernisse, Einsatz
- P-77 Jana Dittmann (Hrsg.): SICHERHEIT 2006
- P-78 K.-O. Wenkel, P. Wagner, M. Morgens-tern, K. Luzi, P. Eisermann (Hrsg.): Land- und Ernährungswirtschaft im Wandel
- P-79 Bettina Biel, Matthias Book, Volker Gruhn (Hrsg.): Softwareengineering 2006
- P-80 Mareike Schoop, Christian Huemer, Michael Rebstock, Martin Bichler (Hrsg.): Service-Oriented Electronic Commerce
- P-81 Wolfgang Karl, Jürgen Becker, Karl-Erwin Großpietsch, Christian Hochberger, Erik Maehle (Hrsg.): ARCS'06
- P-82 Heinrich C. Mayr, Ruth Breu (Hrsg.): Modellierung 2006
- P-83 Daniel Huson, Oliver Kohlbacher, Andrei Lupas, Kay Nieselt and Andreas Zell (eds.): German Conference on Bioinformatics
- P-84 Dimitris Karagiannis, Heinrich C. Mayr, (Hrsg.): Information Systems Technology and its Applications
- P-85 Witold Abramowicz, Heinrich C. Mayr, (Hrsg.): Business Information Systems
- P-86 Robert Krimmer (Ed.): Electronic Voting 2006
- P-87 Max Mühlhäuser, Guido Rößling, Ralf Steinmetz (Hrsg.): DELFI 2006: 4. e-Learning Fachtagung Informatik
- P-88 Robert Hirschfeld, Andreas Polze, Ryszard Kowalczyk (Hrsg.): NODe 2006, GSEM 2006
- P-90 Joachim Schelp, Robert Winter, Ulrich Frank, Bodo Rieger, Klaus Turowski (Hrsg.): Integration, Informationslogistik und Architektur
- P-91 Henrik Stormer, Andreas Meier, Michael Schumacher (Eds.): European Conference on eHealth 2006
- P-92 Fernand Feltz, Benoît Otjacques, Andreas Oberweis, Nicolas Poussing (Eds.): AIM 2006
- P-93 Christian Hochberger, Rüdiger Liskowsky (Eds.): INFORMATIK 2006 – Informatik für Menschen, Band 1
- P-94 Christian Hochberger, Rüdiger Liskowsky (Eds.): INFORMATIK 2006 – Informatik für Menschen, Band 2
- P-95 Matthias Weske, Markus Nüttgens (Eds.): EMISA 2005: Methoden, Konzepte und Technologien für die Entwicklung von dienstbasierten Informationssystemen
- P-96 Saartje Brockmans, Jürgen Jung, York Sure (Eds.): Meta-Modelling and Ontologies
- P-97 Oliver Göbel, Dirk Schadt, Sandra Frings, Hardo Hase, Detlef Günther, Jens Nedon (Eds.): IT-Incident Mangament & IT-Forensics – IMF 2006

- P-98 Hans Brandt-Pook, Werner Simonsmeier und Thorsten Spitta (Hrsg.): Beratung in der Softwareentwicklung – Modelle, Methoden, Best Practices
- P-99 Andreas Schwill, Carsten Schulte, Marco Thomas (Hrsg.): Didaktik der Informatik
- P-100 Peter Forbrig, Günter Siegel, Markus Schneider (Hrsg.): HDI 2006: Hochschuldidaktik der Informatik
- P-101 Stefan Böttinger, Ludwig Theuvsen, Susanne Rank, Marlies Morgenstern (Hrsg.): Agrarinformatik im Spannungsfeld zwischen Regionalisierung und globalen Wertschöpfungsketten
- P-102 Otto Spaniol (Eds.): Mobile Services and Personalized Environments
- P-103 Alfons Kemper, Harald Schöning, Thomas Rose, Matthias Jarke, Thomas Seidl, Christoph Quix, Christoph Brochhaus (Hrsg.): Datenbanksysteme in Business, Technologie und Web (BTW 2007)
- P-104 Birgitta König-Ries, Franz Lehner, Rainer Malaka, Can Türker (Hrsg.) MMS 2007: Mobilität und mobile Informationssysteme
- P-105 Wolf-Gideon Bleek, Jörg Raasch, Heinz Züllighoven (Hrsg.) Software Engineering 2007
- P-106 Wolf-Gideon Bleek, Henning Schwentner, Heinz Züllighoven (Hrsg.) Software Engineering 2007 – Beiträge zu den Workshops
- P-107 Heinrich C. Mayr, Dimitris Karagiannis (eds.) Information Systems Technology and its Applications
- P-108 Arslan Brömme, Christoph Busch, Detlef Hühnlein (eds.) BIOSIG 2007: Biometrics and Electronic Signatures
- P-109 Rainer Koschke, Otthein Herzog, Karl-Heinz Rödiger, Marc Ronthaler (Hrsg.) INFORMATIK 2007 Informatik trifft Logistik Band 1
- P-110 Rainer Koschke, Otthein Herzog, Karl-Heinz Rödiger, Marc Ronthaler (Hrsg.) INFORMATIK 2007 Informatik trifft Logistik Band 2
- P-111 Christian Eibl, Johannes Magenheimer, Sigrid Schubert, Martin Wessner (Hrsg.) DeLFI 2007: 5. e-Learning Fachtagung Informatik
- P-112 Sigrid Schubert (Hrsg.) Didaktik der Informatik in Theorie und Praxis
- P-113 Sören Auer, Christian Bizer, Claudia Müller, Anna V. Zhdanova (Eds.) The Social Semantic Web 2007 Proceedings of the 1st Conference on Social Semantic Web (CSSW)
- P-114 Sandra Frings, Oliver Göbel, Detlef Günther, Hardo G. Hase, Jens Nedon, Dirk Schadt, Arslan Brömme (Eds.) IMF2007 IT-incident management & IT-forensics Proceedings of the 3rd International Conference on IT-Incident Management & IT-Forensics
- P-115 Claudia Falter, Alexander Schliep, Joachim Selbig, Martin Vingron and Dirk Walther (Eds.) German conference on bioinformatics GCB 2007
- P-116 Witold Abramowicz, Leszek Maciszek (Eds.) Business Process and Services Computing 1st International Working Conference on Business Process and Services Computing BPSC 2007
- P-117 Ryszard Kowalczyk (Ed.) Grid service engineering and management The 4th International Conference on Grid Service Engineering and Management GSEM 2007
- P-118 Andreas Hein, Wilfried Thoben, Hans-Jürgen Appelrath, Peter Jensch (Eds.) European Conference on ehealth 2007
- P-119 Manfred Reichert, Stefan Strecker, Klaus Turowski (Eds.) Enterprise Modelling and Information Systems Architectures Concepts and Applications
- P-120 Adam Pawlak, Kurt Sandkuhl, Wojciech Cholewa, Leandro Soares Indrusiak (Eds.) Coordination of Collaborative Engineering - State of the Art and Future Challenges
- P-121 Korbinian Hermann, Bernd Bruegge (Hrsg.) Software Engineering 2008 Fachtagung des GI-Fachbereichs Softwaretechnik
- P-122 Walid Maalej, Bernd Bruegge (Hrsg.) Software Engineering 2008 - Workshopband Fachtagung des GI-Fachbereichs Softwaretechnik

- P-123 Michael H. Breitner, Martin Breunig, Elgar Fleisch, Ley Pousttchi, Klaus Turowski (Hrsg.)
Mobile und Ubiquitäre Informationssysteme – Technologien, Prozesse, Marktfähigkeit
Proceedings zur 3. Konferenz Mobile und Ubiquitäre Informationssysteme (MMS 2008)
- P-124 Wolfgang E. Nagel, Rolf Hoffmann, Andreas Koch (Eds.)
9th Workshop on Parallel Systems and Algorithms (PASA)
Workshop of the GI/ITG Special Interest Groups PARS and PARVA
- P-125 Rolf A.E. Müller, Hans-H. Sundermeier, Ludwig Theuvsen, Stephanie Schütze, Marlies Morgenstern (Hrsg.)
Unternehmens-IT:
Führungsinstrument oder Verwaltungsbürde
Referate der 28. GIL Jahrestagung
- P-126 Rainer Gimnich, Uwe Kaiser, Jochen Quante, Andreas Winter (Hrsg.)
10th Workshop Software Reengineering (WSR 2008)
- P-127 Thomas Kühne, Wolfgang Reisig, Friedrich Steimann (Hrsg.)
Modellierung 2008
- P-128 Ammar Alkassar, Jörg Siekmann (Hrsg.)
Sicherheit 2008
Sicherheit, Schutz und Zuverlässigkeit
Beiträge der 4. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.V. (GI)
2.-4. April 2008
Saarbrücken, Germany
- P-129 Wolfgang Hesse, Andreas Oberweis (Eds.)
Sigsand-Europe 2008
Proceedings of the Third AIS SIGSAND European Symposium on Analysis, Design, Use and Societal Impact of Information Systems
- P-130 Paul Müller, Bernhard Neumair, Gabi Dreö Rodosek (Hrsg.)
1. DFN-Forum Kommunikationstechnologien Beiträge der Fachtagung
- P-131 Robert Krimmer, Rüdiger Grimm (Eds.)
3rd International Conference on Electronic Voting 2008
Co-organized by Council of Europe, Gesellschaft für Informatik und E-Voting, CC
- P-132 Silke Seehusen, Ulrike Lucke, Stefan Fischer (Hrsg.)
DeLFI 2008:
Die 6. e-Learning Fachtagung Informatik
- P-133 Heinz-Gerd Hegering, Axel Lehmann, Hans Jürgen Ohlbach, Christian Scheideler (Hrsg.)
INFORMATIK 2008
Beherrschbare Systeme – dank Informatik Band 1
- P-134 Heinz-Gerd Hegering, Axel Lehmann, Hans Jürgen Ohlbach, Christian Scheideler (Hrsg.)
INFORMATIK 2008
Beherrschbare Systeme – dank Informatik Band 2
- P-135 Torsten Brinda, Michael Fothe, Peter Hubwieser, Kirsten Schlüter (Hrsg.)
Didaktik der Informatik –
Aktuelle Forschungsergebnisse
- P-136 Andreas Beyer, Michael Schroeder (Eds.)
German Conference on Bioinformatics GCB 2008
- P-137 Arslan Brömme, Christoph Busch, Detlef Hühnlein (Eds.)
BIOSIG 2008: Biometrics and Electronic Signatures
- P-138 Barbara Dinter, Robert Winter, Peter Chamoni, Norbert Gronau, Klaus Turowski (Hrsg.)
Synergien durch Integration und Informationslogistik
Proceedings zur DW2008
- P-139 Georg Herzwurm, Martin Mikusz (Hrsg.)
Industrialisierung des Software-Managements
Fachtagung des GI-Fachausschusses Management der Anwendungsentwicklung und -wartung im Fachbereich Wirtschaftsinformatik
- P-140 Oliver Göbel, Sandra Frings, Detlef Günther, Jens Nedon, Dirk Schadt (Eds.)
IMF 2008 - IT Incident Management & IT Forensics
- P-141 Peter Loos, Markus Nüttgens, Klaus Turowski, Dirk Werth (Hrsg.)
Modellierung betrieblicher Informationssysteme (MobIS 2008)
Modellierung zwischen SOA und Compliance Management
- P-142 R. Bill, P. Korduan, L. Theuvsen, M. Morgenstern (Hrsg.)
Anforderungen an die Agrarinformatik durch Globalisierung und Klimaveränderung
- P-143 Peter Liggesmeyer, Gregor Engels, Jürgen Münch, Jörg Dörr, Norman Riegel (Hrsg.)
Software Engineering 2009
Fachtagung des GI-Fachbereichs Softwaretechnik

- P-144 Johann-Christoph Freytag, Thomas Ruf, Wolfgang Lehner, Gottfried Vossen (Hrsg.)
Datenbanksysteme in Business, Technologie und Web (BTW)
- P-145 Knut Hinkelmann, Holger Wache (Eds.)
WM2009: 5th Conference on Professional Knowledge Management
- P-146 Markus Bick, Martin Breunig, Hagen Höpfner (Hrsg.)
Mobile und Ubiquitäre Informationssysteme – Entwicklung, Implementierung und Anwendung
4. Konferenz Mobile und Ubiquitäre Informationssysteme (MMS 2009)
- P-147 Witold Abramowicz, Leszek Maciaszek, Ryszard Kowalczyk, Andreas Speck (Eds.)
Business Process, Services Computing and Intelligent Service Management
BPSC 2009 · ISM 2009 · YRW-MBP 2009
- P-148 Christian Erfurth, Gerald Eichler, Volkmar Schau (Eds.)
9th International Conference on Innovative Internet Community Systems
I²CS 2009
- P-149 Paul Müller, Bernhard Neumair, Gabi Dreö Rodosek (Hrsg.)
2. DFN-Forum
Kommunikationstechnologien
Beiträge der Fachtagung
- P-150 Jürgen Münch, Peter Liggesmeyer (Hrsg.)
Software Engineering
2009 - Workshopband
- P-151 Armin Heinzl, Peter Dadam, Stefan Kirm, Peter Lockemann (Eds.)
PRIMIUM
Process Innovation for
Enterprise Software
- P-152 Jan Mendling, Stefanie Rinderle-Ma, Werner Esswein (Eds.)
Enterprise Modelling and Information Systems Architectures
Proceedings of the 3rd Int'l Workshop
EMISA 2009
- P-153 Andreas Schwill, Nicolas Apostolopoulos (Hrsg.)
Lernen im Digitalen Zeitalter
DeLFI 2009 – Die 7. E-Learning
Fachtagung Informatik
- P-154 Stefan Fischer, Erik Maehle
Rüdiger Reischuk (Hrsg.)
INFORMATIK 2009
Im Focus das Leben
- P-155 Arslan Brömme, Christoph Busch, Detlef Hühnlein (Eds.)
BIOSIG 2009:
Biometrics and Electronic Signatures
Proceedings of the Special Interest Group on Biometrics and Electronic Signatures
- P-156 Bernhard Koerber (Hrsg.)
Zukunft braucht Herkunft
25 Jahre »INFOS – Informatik und Schule«
- P-157 Ivo Grosse, Steffen Neumann, Stefan Posch, Falk Schreiber, Peter Stadler (Eds.)
German Conference on Bioinformatics
2009
- P-158 W. Claudepein, L. Theuvsen, A. Kämpf, M. Morgenstern (Hrsg.)
Precision Agriculture
Reloaded – Informationsgestützte
Landwirtschaft
- P-159 Gregor Engels, Markus Luckey, Wilhelm Schäfer (Hrsg.)
Software Engineering 2010
- P-160 Gregor Engels, Markus Luckey, Alexander Pretschner, Ralf Reussner (Hrsg.)
Software Engineering 2010 –
Workshopband
(inkl. Doktorandensymposium)
- P-161 Gregor Engels, Dimitris Karagiannis
Heinrich C. Mayr (Hrsg.)
Modellierung 2010
- P-162 Maria A. Wimmer, Uwe Brinkhoff, Siegfried Kaiser, Dagmar Lück-Schneider, Erich Schweighofer, Andreas Wiebe (Hrsg.)
Vernetzte IT für einen effektiven Staat
Gemeinsame Fachtagung
Verwaltungsinformatik (FTVI) und
Fachtagung Rechtsinformatik (FTRI) 2010
- P-163 Markus Bick, Stefan Eulgem, Elgar Fleisch, J. Felix Hampe, Birgitta König-Ries, Franz Lehner, Key Pousttchi, Kai Rannenberg (Hrsg.)
Mobile und Ubiquitäre
Informationssysteme
Technologien, Anwendungen und
Dienste zur Unterstützung von mobiler
Kollaboration
- P-164 Arslan Brömme, Christoph Busch (Eds.)
BIOSIG 2010: Biometrics and Electronic
Signatures Proceedings of the Special
Interest Group on Biometrics and
Electronic Signatures

- P-165 Gerald Eichler, Peter Kropf, Ulrike Lechner, Phayung Meesad, Herwig Unger (Eds.)
10th International Conference on Innovative Internet Community Systems (I²CS) – Jubilee Edition 2010 –
- P-166 Paul Müller, Bernhard Neumair, Gabi Dreö Rodosek (Hrsg.)
3. DFN-Forum Kommunikationstechnologien
Beiträge der Fachtagung
- P-167 Robert Krimmer, Rüdiger Grimm (Eds.)
4th International Conference on Electronic Voting 2010
co-organized by the Council of Europe, Gesellschaft für Informatik and E-Voting.CC
- P-168 Ira Diethelm, Christina Dörge, Claudia Hildebrandt, Carsten Schulte (Hrsg.)
Didaktik der Informatik
Möglichkeiten empirischer Forschungsmethoden und Perspektiven der Fachdidaktik
- P-169 Michael Kerres, Nadine Ojstersek, Ulrik Schroeder, Ulrich Hoppe (Hrsg.)
DeLFI 2010 - 8. Tagung der Fachgruppe E-Learning der Gesellschaft für Informatik e.V.
- P-170 Felix C. Freiling (Hrsg.)
Sicherheit 2010
Sicherheit, Schutz und Zuverlässigkeit
- P-171 Werner Esswein, Klaus Turowski, Martin Juhrisch (Hrsg.)
Modellierung betrieblicher Informationssysteme (MobIS 2010)
Modellgestütztes Management
- P-172 Stefan Klink, Agnes Koschmider, Marco Mevius, Andreas Oberweis (Hrsg.)
EMISA 2010
Einflussfaktoren auf die Entwicklung flexibler, integrierter Informationssysteme
Beiträge des Workshops der GI-Fachgruppe EMISA
(Entwicklungsmethoden für Informationssysteme und deren Anwendung)
- P-173 Dietmar Schomburg, Andreas Grote (Eds.)
German Conference on Bioinformatics 2010
- P-174 Arslan Brömme, Torsten Eymann, Detlef Hühnlein, Heiko Roßnagel, Paul Schmücker (Hrsg.)
perspeGktive 2010
Workshop „Innovative und sichere Informationstechnologie für das Gesundheitswesen von morgen“
- P-175 Klaus-Peter Fährnich, Bogdan Franczyk (Hrsg.)
INFORMATIK 2010
Service Science – Neue Perspektiven für die Informatik
Band 1
- P-176 Klaus-Peter Fährnich, Bogdan Franczyk (Hrsg.)
INFORMATIK 2010
Service Science – Neue Perspektiven für die Informatik
Band 2
- P-177 Witold Abramowicz, Rainer Alt, Klaus-Peter Fährnich, Bogdan Franczyk, Leszek A. Maciaszek (Eds.)
INFORMATIK 2010
Business Process and Service Science – Proceedings of ISSS and BPSC
- P-178 Wolfram Pietsch, Benedikt Krams (Hrsg.)
Vom Projekt zum Produkt
Fachtagung des GI-Fachausschusses Management der Anwendungsentwicklung und -wartung im Fachbereich Wirtschaftsinformatik (WI-MAW), Aachen, 2010
- P-179 Stefan Gruner, Bernhard Rumpe (Eds.)
FM+AM'2010
Second International Workshop on Formal Methods and Agile Methods
- P-180 Theo Härder, Wolfgang Lehner, Bernhard Mitschang, Harald Schöning, Holger Schwarz (Hrsg.)
Datenbanksysteme für Business, Technologie und Web (BTW)
14. Fachtagung des GI-Fachbereichs „Datenbanken und Informationssysteme“ (DBIS)
- P-181 Michael Clasen, Otto Schätzel, Brigitte Theuvsen (Hrsg.)
Qualität und Effizienz durch informationsgestützte Landwirtschaft, Fokus: Moderne Weinwirtschaft
- P-182 Ronald Maier (Hrsg.)
6th Conference on Professional Knowledge Management
From Knowledge to Action
- P-183 Ralf Reussner, Matthias Grund, Andreas Oberweis, Walter Tichy (Hrsg.)
Software Engineering 2011
Fachtagung des GI-Fachbereichs Softwaretechnik
- P-184 Ralf Reussner, Alexander Pretschner, Stefan Jähnichen (Hrsg.)
Software Engineering 2011
Workshopband
(inkl. Doktorandensymposium)

- P-185 Hagen Höpfner, Günther Specht, Thomas Ritz, Christian Bunse (Hrsg.)
MMS 2011: Mobile und ubiquitäre Informationssysteme Proceedings zur 6. Konferenz Mobile und Ubiquitäre Informationssysteme (MMS 2011)
- P-186 Gerald Eichler, Axel Küpper, Volkmar Schau, Hacène Fouchal, Herwig Unger (Eds.)
11th International Conference on Innovative Internet Community Systems (I²CS)
- P-187 Paul Müller, Bernhard Neumair, Gabi Dreö Rodosek (Hrsg.)
4. DFN-Forum Kommunikationstechnologien, Beiträge der Fachtagung 20. Juni bis 21. Juni 2011 Bonn
- P-188 Holger Rohland, Andrea Kienle, Steffen Friedrich (Hrsg.)
DeLFI 2011 – Die 9. e-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V. 5.–8. September 2011, Dresden
- P-189 Thomas, Marco (Hrsg.)
Informatik in Bildung und Beruf INFOS 2011
14. GI-Fachtagung Informatik und Schule
- P-190 Markus Nüttgens, Oliver Thomas, Barbara Weber (Eds.)
Enterprise Modelling and Information Systems Architectures (EMISA 2011)
- P-191 Arslan Brömme, Christoph Busch (Eds.)
BIOSIG 2011
International Conference of the Biometrics Special Interest Group
- P-192 Hans-Ulrich Heiß, Peter Pepper, Holger Schlingloff, Jörg Schneider (Hrsg.)
INFORMATIK 2011
Informatik schafft Communities
- P-193 Wolfgang Lehner, Gunther Piller (Hrsg.)
IMDM 2011
- P-194 M. Clasen, G. Fröhlich, H. Bernhardt, K. Hildebrand, B. Theuvsen (Hrsg.)
Informationstechnologie für eine nachhaltige Landwirtschaft Fokus Forstwirtschaft
- P-195 Neeraj Suri, Michael Waidner (Hrsg.)
Sicherheit 2012
Sicherheit, Schutz und Zuverlässigkeit Beiträge der 6. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.V. (GI)
- P-196 Arslan Brömme, Christoph Busch (Eds.)
BIOSIG 2012
Proceedings of the 11th International Conference of the Biometrics Special Interest Group
- P-197 Jörn von Lucke, Christian P. Geiger, Siegfried Kaiser, Erich Schweighofer, Maria A. Wimmer (Hrsg.)
Auf dem Weg zu einer offenen, smarten und vernetzten Verwaltungskultur Gemeinsame Fachtagung Verwaltungsinformatik (FTVI) und Fachtagung Rechtsinformatik (FTRI) 2012
- P-198 Stefan Jähnichen, Axel Küpper, Sahin Albayrak (Hrsg.)
Software Engineering 2012
Fachtagung des GI-Fachbereichs Softwaretechnik
- P-199 Stefan Jähnichen, Bernhard Rumpe, Holger Schlingloff (Hrsg.)
Software Engineering 2012
Workshopband
- P-200 Gero Mühl, Jan Richling, Andreas Herkersdorf (Hrsg.)
ARCS 2012 Workshops
- P-201 Elmar J. Sinz Andy Schürr (Hrsg.)
Modellierung 2012
- P-202 Andrea Back, Markus Bick, Martin Breunig, Key Poustchi, Frédéric Thiesse (Hrsg.)
MMS 2012: Mobile und Ubiquitäre Informationssysteme
- P-203 Paul Müller, Bernhard Neumair, Helmut Reiser, Gabi Dreö Rodosek (Hrsg.)
5. DFN-Forum Kommunikationstechnologien
Beiträge der Fachtagung
- P-204 Gerald Eichler, Leendert W. M. Wienhofen, Anders Kofod-Petersen, Herwig Unger (Eds.)
12th International Conference on Innovative Internet Community Systems (I²CS 2012)
- P-205 Manuel J. Kripp, Melanie Volkamer, Rüdiger Grimm (Eds.)
5th International Conference on Electronic Voting 2012 (EVOTE2012)
Co-organized by the Council of Europe, Gesellschaft für Informatik und E-Voting.CC
- P-206 Stefanie Rinderle-Ma, Mathias Weske (Hrsg.)
EMISA 2012
Der Mensch im Zentrum der Modellierung
- P-207 Jörg Desel, Jörg M. Haake, Christian Spannagel (Hrsg.)
DeLFI 2012: Die 10. e-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V.
24.–26. September 2012

- P-208 Ursula Goltz, Marcus Magnor, Hans-Jürgen Appelrath, Herbert Matthies, Wolf-Tilo Balke, Lars Wolf (Hrsg.)
INFORMATIK 2012
- P-209 Hans Brandt-Pook, André Fleer, Thorsten Spitta, Malte Wattenberg (Hrsg.)
Nachhaltiges Software Management
- P-210 Erhard Plödereder, Peter Dencker, Herbert Klenk, Hubert B. Keller, Silke Spitzer (Hrsg.)
Automotive – Safety & Security 2012
Sicherheit und Zuverlässigkeit für automobile Informationstechnik
- P-211 M. Clasen, K. C. Kersebaum, A. Meyer-Aurich, B. Theuvsen (Hrsg.)
Massendatenmanagement in der Agrar- und Ernährungswirtschaft
Erhebung - Verarbeitung - Nutzung
Referate der 33. GIL-Jahrestagung
20. – 21. Februar 2013, Potsdam
- P-212 Arslan Brömme, Christoph Busch (Eds.)
BIOSIG 2013
Proceedings of the 12th International Conference of the Biometrics Special Interest Group
04.–06. September 2013
Darmstadt, Germany
- P-213 Stefan Kowalewski, Bernhard Rumpe (Hrsg.)
Software Engineering 2013
Fachtagung des GI-Fachbereichs Softwaretechnik
- P-214 Volker Markl, Gunter Saake, Kai-Uwe Sattler, Gregor Hackenbroich, Bernhard Mitschang, Theo Härder, Veit Köppen (Hrsg.)
Datenbanksysteme für Business, Technologie und Web (BTW) 2013
13. – 15. März 2013, Magdeburg
- P-215 Stefan Wagner, Horst Lichter (Hrsg.)
Software Engineering 2013
Workshopband
(inkl. Doktorandensymposium)
26. Februar – 1. März 2013, Aachen
- P-216 Gunter Saake, Andreas Henrich, Wolfgang Lehner, Thomas Neumann, Veit Köppen (Hrsg.)
Datenbanksysteme für Business, Technologie und Web (BTW) 2013 – Workshopband
11. – 12. März 2013, Magdeburg
- P-217 Paul Müller, Bernhard Neumair, Helmut Reiser, Gabi Dreö Rodosek (Hrsg.)
6. DFN-Forum Kommunikationstechnologien
Beiträge der Fachtagung
03.–04. Juni 2013, Erlangen
- P-218 Andreas Breiter, Christoph Rensing (Hrsg.)
DeLFI 2013: Die 11 e-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V. (GI)
8. – 11. September 2013, Bremen
- P-219 Norbert Breier, Peer Stechert, Thomas Wilke (Hrsg.)
Informatik erweitert Horizonte
INFOS 2013
15. GI-Fachtagung Informatik und Schule
26. – 28. September 2013
- P-220 Matthias Horbach (Hrsg.)
INFORMATIK 2013
Informatik angepasst an Mensch, Organisation und Umwelt
16. – 20. September 2013, Koblenz
- P-221 Maria A. Wimmer, Marijn Janssen, Ann Macintosh, Hans Jochen Scholl, Efthimios Tambouris (Eds.)
Electronic Government and Electronic Participation
Joint Proceedings of Ongoing Research of IFIP EGOV and IFIP ePart 2013
16. – 19. September 2013, Koblenz
- P-222 Reinhard Jung, Manfred Reichert (Eds.)
Enterprise Modelling and Information Systems Architectures (EMISA 2013)
St. Gallen, Switzerland
September 5. – 6. 2013
- P-223 Detlef Hühnlein, Heiko Roßnagel (Hrsg.)
Open Identity Summit 2013
10. – 11. September 2013
Kloster Banz, Germany
- P-224 Eckhart Hanser, Martin Mikusz, Masud Fazal-Baqaie (Hrsg.)
Vorgehensmodelle 2013
Vorgehensmodelle – Anspruch und Wirklichkeit
20. Tagung der Fachgruppe Vorgehensmodelle im Fachgebiet Wirtschaftsinformatik (WI-VM) der Gesellschaft für Informatik e.V.
Lörrach, 2013
- P-225 Hans-Georg Fill, Dimitris Karagiannis, Ulrich Reimer (Hrsg.)
Modellierung 2014
19. – 21. März 2014, Wien
- P-226 M. Clasen, M. Hamer, S. Lehnert, B. Petersen, B. Theuvsen (Hrsg.)
IT-Standards in der Agrar- und Ernährungswirtschaft Fokus: Risiko- und Krisenmanagement
Referate der 34. GIL-Jahrestagung
24. – 25. Februar 2014, Bonn

- P-227 Wilhelm Hasselbring,
Nils Christian Ehmke (Hrsg.)
Software Engineering 2014
Fachtagung des GI-Fachbereichs
Softwaretechnik
25. – 28. Februar 2014
Kiel, Deutschland
- P-228 Stefan Katzenbeisser, Volkmar Lotz,
Edgar Weippl (Hrsg.)
Sicherheit 2014
Sicherheit, Schutz und Zuverlässigkeit
Beiträge der 7. Jahrestagung des
Fachbereichs Sicherheit der
Gesellschaft für Informatik e.V. (GI)
19. – 21. März 2014, Wien
- P-229 Dagmar Lück-Schneider, Thomas
Gordon, Siegfried Kaiser, Jörn von
Lucke, Erich Schweighofer, Maria
A. Wimmer, Martin G. Löhle (Hrsg.)
Gemeinsam Electronic Government
ziel(gruppen)gerecht gestalten und
organisieren
Gemeinsame Fachtagung
Verwaltungsinformatik (FTVI) und
Fachtagung Rechtsinformatik (FTRI)
2014, 20.-21. März 2014 in Berlin
- P-230 Arslan Brömme, Christoph Busch (Eds.)
BIOSIG 2014
Proceedings of the 13th International
Conference of the Biometrics Special
Interest Group
10. – 12. September 2014 in
Darmstadt, Germany
- P-231 Paul Müller, Bernhard Neumair,
Helmut Reiser, Gabi Dreö Rodosek
(Hrsg.)
7. DFN-Forum
Kommunikationstechnologien
16. – 17. Juni 2014
Fulda
- P-232 E. Plödereder, L. Grunske, E. Schneider,
D. Ull (Hrsg.)
INFORMATIK 2014
Big Data – Komplexität meistern
22. – 26. September 2014
Stuttgart
- P-233 Stephan Trahasch, Rolf Plötzner, Gerhard
Schneider, Claudia Gayer, Daniel Sassiat,
Nicole Wöhrle (Hrsg.)
DeLFI 2014 – Die 12. e-Learning
Fachtagung Informatik
der Gesellschaft für Informatik e.V.
15. – 17. September 2014
Freiburg
- P-234 Fernand Feltz, Bela Mutschler, Benoît
Otjacques (Eds.)
Enterprise Modelling and Information
Systems Architectures
(EMISA 2014)
Luxembourg, September 25-26, 2014
- P-235 Robert Giegerich,
Ralf Hofestädt,
Tim W. Nattkemper (Eds.)
German Conference on
Bioinformatics 2014
September 28 – October 1
Bielefeld, Germany
- P-236 Martin Engstler, Eckhart Hanser,
Martin Mikusz, Georg Herzwurm (Hrsg.)
Projektmanagement und
Vorgehensmodelle 2014
Soziale Aspekte und Standardisierung
Gemeinsame Tagung der Fachgruppen
Projektmanagement (WI-PM) und
Vorgehensmodelle (WI-VM) im
Fachgebiet Wirtschaftsinformatik der
Gesellschaft für Informatik e.V., Stuttgart
2014
- P-237 Detlef Hühnlein, Heiko Roßnagel (Hrsg.)
Open Identity Summit 2014
4.–6. November 2014
Stuttgart, Germany
- P-238 Arno Ruckelshausen, Hans-Peter
Schwarz, Brigitte Theuvsen (Hrsg.)
Informatik in der Land-, Forst- und
Ernährungswirtschaft
Referate der 35. GIL-Jahrestagung
23. – 24. Februar 2015, Geisenheim
- P-239 Uwe Aßmann, Birgit Demuth, Thorsten
Spitta, Georg Püschel, Ronny Kaiser
(Hrsg.)
Software Engineering & Management
2015
17.-20. März 2015, Dresden
- P-240 Herbert Klenk, Hubert B. Keller, Erhard
Plödereder, Peter Dencker (Hrsg.)
Automotive – Safety & Security 2015
Sicherheit und Zuverlässigkeit für
automobile Informationstechnik
21.–22. April 2015, Stuttgart
- P-241 Thomas Seidl, Norbert Ritter,
Harald Schöning, Kai-Uwe Sattler,
Theo Härder, Steffen Friedrich,
Wolfram Wingerath (Hrsg.)
Datenbanksysteme für Business,
Technologie und Web (BTW 2015)
04. – 06. März 2015, Hamburg

- P-242 Norbert Ritter, Andreas Henrich, Wolfgang Lehner, Andreas Thor, Steffen Friedrich, Wolfram Wingerath (Hrsg.)
Datenbanksysteme für Business, Technologie und Web (BTW 2015) – Workshopband
02. – 03. März 2015, Hamburg
- P-243 Paul Müller, Bernhard Neumair, Helmut Reiser, Gabi Dreö Rodosek (Hrsg.)
8. DFN-Forum
Kommunikationstechnologien
06.–09. Juni 2015, Lübeck
- P-244 Alfred Zimmermann, Alexander Rossmann (Eds.)
Digital Enterprise Computing (DEC 2015)
Böblingen, Germany June 25-26, 2015
- P-245 Arslan Brömme, Christoph Busch, Christian Rathgeb, Andreas Uhl (Eds.)
BIOSIG 2015
Proceedings of the 14th International Conference of the Biometrics Special Interest Group
09.–11. September 2015
Darmstadt, Germany
- P-246 Douglas W. Cunningham, Petra Hofstedt, Klaus Meer, Ingo Schmitt (Hrsg.)
INFORMATIK 2015
28.9.-2.10. 2015, Cottbus
- P-247 Hans Pongratz, Reinhard Keil (Hrsg.)
DeLFI 2015 – Die 13. E-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V. (GI)
1.–4. September 2015
München
- P-248 Jens Kolb, Henrik Leopold, Jan Mendling (Eds.)
Enterprise Modelling and Information Systems Architectures
Proceedings of the 6th Int. Workshop on Enterprise Modelling and Information Systems Architectures, Innsbruck, Austria
September 3-4, 2015
- P-249 Jens Gallenbacher (Hrsg.)
Informatik
allgemeinbildend begreifen
INFOS 2015 16. GI-Fachtagung Informatik und Schule
20.–23. September 2015
- P-250 Martin Engstler, Masud Fazal-Baqaie, Eckhart Hanser, Martin Mikusz, Alexander Volland (Hrsg.)
Projektmanagement und Vorgehensmodelle 2015
Hybride Projektstrukturen erfolgreich umsetzen
Gemeinsame Tagung der Fachgruppen Projektmanagement (WI-PM) und Vorgehensmodelle (WI-VM) im Fachgebiet Wirtschaftsinformatik der Gesellschaft für Informatik e.V., Elmshorn 2015
- P-251 Detlef Hühnlein, Heiko Roßnagel, Raik Kuhlisch, Jan Ziesing (Eds.)
Open Identity Summit 2015
10.–11. November 2015
Berlin, Germany
- P-252 Jens Knoop, Uwe Zdun (Hrsg.)
Software Engineering 2016
Fachtagung des GI-Fachbereichs Softwaretechnik
23.–26. Februar 2016, Wien
- P-253 A. Ruckelshausen, A. Meyer-Aurich, T. Rath, G. Recke, B. Theuvsen (Hrsg.)
Informatik in der Land-, Forst- und Ernährungswirtschaft
Fokus: Intelligente Systeme – Stand der Technik und neue Möglichkeiten
Referate der 36. GIL-Jahrestagung
22.-23. Februar 2016, Osnabrück
- P-254 Andreas Oberweis, Ralf Reussner (Hrsg.)
Modellierung 2016
2.–4. März 2016, Karlsruhe
- P-255 Stefanie Betz, Ulrich Reimer (Hrsg.)
Modellierung 2016 Workshopband
2.–4. März 2016, Karlsruhe
- P-256 Michael Meier, Delphine Reinhardt, Steffen Wendzel (Hrsg.)
Sicherheit 2016
Sicherheit, Schutz und Zuverlässigkeit
Beiträge der 8. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.V. (GI)
5.–7. April 2016, Bonn
- P-257 Paul Müller, Bernhard Neumair, Helmut Reiser, Gabi Dreö Rodosek (Hrsg.)
9. DFN-Forum
Kommunikationstechnologien
31. Mai – 01. Juni 2016, Rostock

- P-258 Dieter Hertweck, Christian Decker (Eds.)
Digital Enterprise Computing (DEC 2016)
14.–15. Juni 2016, Böblingen
- P-259 Heinrich C. Mayr, Martin Pinzger (Hrsg.)
INFORMATIK 2016
26.–30. September 2016, Klagenfurt
- P-260 Arslan Brömme, Christoph Busch,
Christian Rathgeb, Andreas Uhl (Eds.)
BIOSIG 2016
Proceedings of the 15th International
Conference of the Biometrics Special
Interest Group
21.–23. September 2016, Darmstadt
- P-261 Detlef Rätz, Michael Breidung, Dagmar
Lück-Schneider, Siegfried Kaiser, Erich
Schweighofer (Hrsg.)
Digitale Transformation: Methoden,
Kompetenzen und Technologien für die
Verwaltung
Gemeinsame Fachtagung
Verwaltungsinformatik (FTVI) und
Fachtagung Rechtsinformatik (FTRI) 2016
22.–23. September 2016, Dresden
- P-262 Ulrike Lucke, Andreas Schwill,
Raphael Zender (Hrsg.)
DeLFI 2016 – Die 14. E-Learning
Fachtagung Informatik
der Gesellschaft für Informatik e.V. (GI)
11.–14. September 2016, Potsdam
- P-263 Martin Engstler, Masud Fazal-Baqaie,
Eckhart Hanser, Oliver Linsen, Martin
Mikusz, Alexander Volland (Hrsg.)
Projektmanagement und
Vorgehensmodelle 2016
Arbeiten in hybriden Projekten: Das
Sowohl-als-auch von Stabilität und
Dynamik
Gemeinsame Tagung der Fachgruppen
Projektmanagement (WI-PM) und
Vorgehensmodelle (WI-VM) im
Fachgebiet Wirtschaftsinformatik
der Gesellschaft für Informatik e.V.,
Paderborn 2016
- P-264 Detlef Hühnlein, Heiko Roßnagel,
Christian H. Schunck, Maurizio Talamo
(Eds.)
Open Identity Summit 2016
der Gesellschaft für Informatik e.V. (GI)
13.–14. October 2016, Rome, Italy
- P-265 Bernhard Mitschang, Daniela
Nicklas, Frank Leymann, Harald
Schöning, Melanie Herschel, Jens
Teubner, Theo Härder, Oliver Kopp,
Matthias Wieland (Hrsg.)
Datenbanksysteme für Business,
Technologie und Web (BTW 2017)
6.–10. März 2017, Stuttgart
- P-266 Bernhard Mitschang, Norbert Ritter,
Holger Schwarz, Meike Klettke, Andreas
Thor, Oliver Kopp, Matthias Wieland
(Hrsg.)
Datenbanksysteme für Business,
Technologie und Web (BTW 2017)
Workshopband
6.–7. März 2017, Stuttgart
- P-267 Jan Jürjens, Kurt Schneider (Hrsg.)
Software Engineering 2017
21.–24. Februar 2017, Hannover
- P-268 A. Ruckelshausen, A. Meyer-Aurich,
W. Lentz, B. Theuvsen (Hrsg.)
Informatik in der Land-, Forst- und
Ernährungswirtschaft
Fokus: Digitale Transformation –
Wege in eine zukunftsfähige
Landwirtschaft
Referate der 37. GIL-Jahrestagung
06.–07. März 2017, Dresden
- P-269 Peter Dencker, Herbert Klenk, Hubert
Keller, Erhard Plödereder (Hrsg.)
Automotive – Safety & Security 2017
30.–31. Mai 2017, Stuttgart
- P-270 Arslan Brömme, Christoph Busch,
Antitza Dantcheva, Christian Rathgeb,
Andreas Uhl (Eds.)
BIOSIG 2017
20.–22. September 2017, Darmstadt
- P-271 Paul Müller, Bernhard Neumair, Helmut
Reiser, Gabi Dreö Rodosek (Hrsg.)
10. DFN-Forum Kommunikations-
technologien
30. – 31. Mai 2017, Berlin
- P-272 Alexander Rossmann, Alfred
Zimmermann (eds.)
Digital Enterprise Computing
(DEC 2017)
11.–12. Juli 2017, Böblingen

- P-273 Christoph Igel, Carsten Ullrich,
Martin Wessner (Hrsg.)
BILDUNGSRÄUME
DeLFI 2017
Die 15. e-Learning Fachtagung Informatik
der Gesellschaft für Informatik e.V. (GI)
5. bis 8. September 2017, Chemnitz
- P-274 Ira Diethelm (Hrsg.)
Informatische Bildung zum Verstehen
und Gestalten der digitalen Welt
13.–15. September 2017, Oldenburg
- P-275 Maximilian Eibl, Martin Gaedke (Hrsg.)
INFORMATIK 2017
25.–29. September 2017, Chemnitz
- P276 Alexander Volland, Martin Engstler,
Masud Fazal-Baqaie, Eckhart Hanser,
Oliver Linssen, Martin Mikusz (Hrsg.)
Projektmanagement und
Vorgehensmodelle 2017
Die Spannung zwischen dem Prozess
und den Menschen im Projekt
Gemeinsame Tagung der Fachgruppen
Projektmanagement und
Vorgehensmodelle im Fachgebiet
Wirtschaftsinformatik der
Gesellschaft für Informatik e.V.
in Kooperation mit der Fachgruppe
IT-Projektmanagement der GPM e.V.,
Darmstadt 2017
- P-277 Lothar Fritsch, Heiko Roßnagel,
Detlef Hühnlein (Hrsg.)
Open Identity Summit 2017
5.–6. Oktober 2017, Karlstad, Sweden
- P-278 Arno Ruckelshausen,
Andreas Meyer-Aurich, Karsten Borchard,
Constanze Hofacker, Jens-Peter Loy,
Rolf Schwerdtfeger,
Hans-Hennig Sundermeier, Helga Floto,
Brigitte Theuvsen (Hrsg.)
Informatik in der Land-, Forst- und
Ernährungswirtschaft
Referate der 38. GIL-Jahrestagung
26.–27. Februar 2018, Kiel
- P-279 Matthias Tichy, Eric Bodden,
Marco Kuhrmann, Stefan Wagner,
Jan-Philipp Steghöfer (Hrsg.)
Software Engineering und Software
Management 2018
5.–9. März 2018, Ulm
- P-280 Ina Schaefer, Dimitris Karagiannis,
Andreas Vogelsang, Daniel Méndez,
Christoph Seidl (Hrsg.)
Modellierung 2018
21.–23. Februar 2018, Braunschweig
- P-281 Hanno Langweg, Michael Meier, Bernhard
C. Witt, Delphine Reinhardt (Hrsg.)
Sicherheit 2018
Sicherheit, Schutz und Zuverlässigkeit
25.–27. April 2018, Konstanz
- P-282 Arslan Brömme, Christoph Busch,
Antitza Dantcheva, Christian Rathgeb,
Andreas Uhl (Eds.)
BIOSIG 2018
Proceedings of the 17th International
Conference of the Biometrics Special
Interest Group
26.–28. September 2018
Darmstadt, Germany
- P-283 Paul Müller, Bernhard Neumair, Helmut
Reiser, Gabi Dreö Rodosek (Hrsg.)
11. DFN-Forum Kommunikations-
technologien
27.–28. Juni 2018, Günzburg
- P-284 Detlef Krömker, Ulrik Schroeder (Hrsg.)
DeLFI 2018 – Die 16. E-Learning
Fachtagung Informatik
10.–12. September 2018, Frankfurt a. M.
- P-285 Christian Czarnecki, Carsten Brockmann,
Eldar Sultanow, Agnes Koschmider,
Annika Selzer (Hrsg.)
Workshops der INFORMATIK 2018 -
Architekturen, Prozesse, Sicherheit und
Nachhaltigkeit
26.–27. September 2018, Berlin
- P-286 Martin Mikusz, Alexander Volland, Martin
Engstler, Masud Fazal-Baqaie, Eckhart
Hanser, Oliver Linssen (Hrsg.)
Projektmanagement und
Vorgehensmodelle 2018
Der Einfluss der Digitalisierung auf
Projektmanagementmethoden und
Entwicklungsprozesse
Düsseldorf 2018

- P-287 A. Meyer-Aurich, M. Gandorfer, N. Barta,
A. Gronauer, J. Kantelhardt, H. Floto (Hrsg.)
Informatik in der Land-, Forst- und
Ernährungswirtschaft
Fokus: Digitalisierung für
landwirtschaftliche Betriebe in
kleinstrukturierten Regionen – ein
Widerspruch in sich?
Referate der 39. GIL-Jahrestagung
18.–19. Februar 2019, Wien
- P-288 Arno Pasternak (Hrsg.)
Informatik für alle
18. GI-Fachtagung
Informatik und Schule
16.-18. September 2019 in Dortmund
- P-289 Torsten Grust, Felix Naumann, Alexander
Böhm, Wolfgang Lehner, Jens Teubner,
Meike Klettke, Theo Härder, Erhard
Rahm, Andreas Heuer, Holger Meyer
(Hrsg.)
Datenbanksysteme für Business,
Technologie und Web (BTW 2019)
4.–8. März 2019 in Rostock
- P-290 Holger Meyer, Norbert Ritter, Andreas
Thor, Daniela Nicklas, Andreas Heuer,
Meike Klettke (Hrsg.)
Datenbanksysteme für Business,
Technologie und Web (BTW 2019)
Workshopband
4.–8. März 2019 in Rostock
- P-291 Michael Räckers, Sebastian Halsbenning,
Detlef Rätz, David Richter,
Erich Schweighofer (Hrsg.)
Digitalisierung von Staat und Verwaltung
Gemeinsame Fachtagung
Verwaltungsinformatik (FTVI) und
Fachtagung Rechtsinformatik (FTRI) 2019
6.–7. März 2019 in Münster
- P-292 Steffen Becker, Ivan Bogicevic, Georg
Herzwurm, Stefan Wagner (Hrsg.)
Software Engineering and Software
Management 2019
18.–22. Februar 2019 in Stuttgart
- P-293 Heiko Roßnagel, Sven Wagner, Detlef
Hühnlein (Hrsg.)
Open Identity Summit 2019
28.–29. März 2019
Garmisch-Partenkirchen

GI-Edition Lecture Notes in Informatics – Seminars

- | | | | |
|------|---|------|--|
| S-1 | Johannes Magenheimer, Sigrid Schubert (Eds.):
Informatics and Student Assessment
Concepts of Empirical Research and
Standardisation of Measurement in the
Area of Didactics of Informatics | S-13 | Gesellschaft für Informatik (Hrsg.)
Informatiktage 2014
Fachwissenschaftlicher Informatik-
Kongress |
| S-2 | Gesellschaft für Informatik (Hrsg.)
Informationstage 2005
Fachwissenschaftlicher Informatik-
Kongress | S-14 | Gesellschaft für Informatik (Hrsg.)
SKILL 2018
Studierendenkonferenz Informatik |
| S-3 | Gesellschaft für Informatik (Hrsg.)
Informationstage 2006
Fachwissenschaftlicher Informatik-
Kongress | S-15 | Gesellschaft für Informatik (Hrsg.)
SKILL 2019
Studierendenkonferenz Informatik |
| S-4 | Hans Hagen, Andreas Kerren, Peter
Dannenmann (Eds.)
Visualization of Large and Unstructured
Data Sets
First workshop of the DFG's International
Research Training Group "Visualization
of Large and Unstructured Data Sets –
Applications in Geospatial Planning,
Modeling and Engineering" | | |
| S-5 | Gesellschaft für Informatik (Hrsg.)
Informationstage 2007
Fachwissenschaftlicher Informatik-
Kongress | | |
| S-6 | Gesellschaft für Informatik (Hrsg.)
Informationstage 2008
Fachwissenschaftlicher Informatik-
Kongress | | |
| S-7 | Hans Hagen, Martin Hering-Bertram,
Christoph Garth (Eds.)
Visualization of Large and Unstructured
Data Sets | | |
| S-8 | Gesellschaft für Informatik (Hrsg.)
Informatiktage 2009
Fachwissenschaftlicher Informatik-
Kongress | | |
| S-9 | Gesellschaft für Informatik (Hrsg.)
Informatiktage 2010
Fachwissenschaftlicher Informatik-
Kongress | | |
| S-10 | Gesellschaft für Informatik (Hrsg.)
Informatiktage 2011
Fachwissenschaftlicher Informatik-
Kongress | | |
| S-11 | Gesellschaft für Informatik (Hrsg.)
Informatiktage 2012
Fachwissenschaftlicher Informatik-
Kongress | | |
| S-12 | Gesellschaft für Informatik (Hrsg.)
Informatiktage 2013
Fachwissenschaftlicher Informatik-
Kongress | | |

The titles can be purchased at:
Köllen Druck + Verlag GmbH
 Ernst-Robert-Curtius-Str. 14 · D-53117 Bonn
 Fax: +49 (0)228/9898222
 E-Mail: druckverlag@koellen.de

Gesellschaft für Informatik e.V. (GI)

publishes this series in order to make available to a broad public recent findings in informatics (i.e. computer science and information systems), to document conferences that are organized in co-operation with GI and to publish the annual GI Award dissertation.

Broken down into

- seminars
- proceedings
- dissertations
- thematics

current topics are dealt with from the vantage point of research and development, teaching and further training in theory and practice. The Editorial Committee uses an intensive review process in order to ensure high quality contributions.

The volumes are published in German or English.

Information: <http://www.gi.de/service/publikationen/lni/>

ISSN 1614-3213

ISBN 978-3-88579-449-3