
AUTOMATED DISCUSSION ANALYSIS IN ONLINE PARTICIPATION PROJECTS

Inaugural-Dissertation

zur

Erlangung des Doktorgrades der
Mathematisch-Naturwissenschaftlichen Fakultät
der Heinrich-Heine-Universität Düsseldorf

vorgelegt von

Matthias Liebeck

aus Düsseldorf

13. Dezember 2017

aus dem Institut für Informatik der
Heinrich-Heine-Universität Düsseldorf

Gedruckt mit der Genehmigung der
Mathematisch-Naturwissenschaftlichen Fakultät der
Heinrich-Heine-Universität Düsseldorf

Referent: Prof. Dr. Stefan Conrad

Koreferent: Prof. Dr. Martin Mauve

Tag der mündlichen Prüfung: 6. April 2018

Ich versichere an Eides Statt, dass die Dissertation von mir selbstständig und ohne unzulässige fremde Hilfe unter Beachtung der *Grundsätze zur Sicherung guter wissenschaftlicher Praxis an der Heinrich-Heine-Universität Düsseldorf* erstellt worden ist.

Die hier vorgelegte Dissertation habe ich eigenständig und ohne unerlaubte Hilfe angefertigt. Die Dissertation wurde in der vorgelegten oder in ähnlicher Form noch bei keiner anderen Institution eingereicht. Ich habe bisher keine erfolglosen Promotionsversuche unternommen.

Düsseldorf, Deutschland
13. Dezember 2017

Matthias Liebeck

Dedicated to my parents.

ACKNOWLEDGEMENTS

My research formed part of the PhD-program *Online Participation*, supported by the North Rhine-Westphalian funding scheme *Fortschrittskollegs*. The unique shape of the PhD-program allowed me to pursue interdisciplinary work and engage closely with city administrations and technical service providers. My discussions with these contacts have deeply influenced the direction of my research.

First, I would like to thank my advisor, Prof. Dr. Stefan Conrad, for the opportunity to work in his research group. He provided an excellent research atmosphere, was very interested in my research, and was always available for my questions.

I also thank Prof. Dr. Martin Mauve for his close interest in my research topic, for heading the Fortschrittskolleg, and for being my second assessor, and Prof. Dr.-Ing. Torsten Zesch for reviewing my thesis.

My research would have been impossible without the aid and support of Katharina Esau, Pashutan Modaresi, and Alexander Askinadze. I thank each of you for our deep friendship, and will always remember our long and productive work sessions. I hope that we will have the opportunity to work together again in the future.

I am grateful for the work of Dr. Tobias Escher and Dr. Lars Heilsberger in coordinating the PhD-program. Furthermore, I want to thank all my research colleagues of the Fortschrittskolleg for our pleasant discussions, events, and business trips.

I want to thank my fellow research colleagues in our research group for all our interesting discussions and the time we shared inside and outside of the university: Marc Büngener, Christian Bock, Kirill Bogomasov, Daniel Braun, Janine Golov, Ludmila Himmelpach, Gerhard Klassen, Magdalena Rischka, Julia Romberg, Michael Singhof, Martha Tatusch, and Matthäus Zloch.

I want to express my gratitude to Sabine Freese, Angela Rennwanz, and Guido Königstein. I am very grateful for your help and support in administrative and technical matters.

I thank Ivan Habernal for our long and fruitful discussions, and his guidance in my early steps of argument mining. Computational support and infrastructure were provided by the Centre for Information and Media Technology (ZIM). Here I want to thank Philipp Rehs for all his technical support regarding our cluster.

Last but not least, I especially thank my family and all of my friends for their years of support. Without your constant help, my research would not have been possible.

ABSTRACT

Cities and municipalities in Germany are more frequently using online participation projects to incorporate the opinion of their citizens into political decision-making processes. Citizens are able to voice their opinions, ideas, and comments in text form on online-based, forum-like platforms. The evaluation of these projects is conducted manually and can be very time consuming if the participants have written thousands of text contributions. In cooperation with technical service providers and cities as part of the PhD program *Online Participation*, we identified a need for automated approaches that assist in the manual evaluation.

First, we focused on argument mining. On the basis of the project *Tempelhofer Feld*, we identified a suitable argument model for online participation projects, annotated text content from a part of the project with three annotators, and achieved a high inter-annotator agreement. Then, we worked on the two machine learning tasks of automatically identifying argumentative content and classifying argument components. In our approach, we evaluated a classical machine learning approach with feature engineering as well as deep learning techniques.

Afterwards, we focused on online participation projects with a high number of text contributions and the task of automatically creating a broad overview of the discussion topics. We started by creating a new lemmatizer for German based on Wiktionary. After a fundamental debate about which text content should be considered for a topic extraction method and how the extracted topics should be visualized, we applied different topic extraction methods to several online participation projects and discussed their results.

Finally, we used text content from citizens involved in the discussion and dealt with the task of automatically inferring demographic attributes in order to identify underrepresented population strata. We developed a multi-lingual author profiling approach for the PAN author profiling challenge in 2016 and achieved first place out of 22 participating teams for gender detection in English text.

ZUSAMMENFASSUNG

In Deutschland setzen Städte und Kommunen vermehrt Online-Partizipationsprojekte ein, um die Meinung ihrer Bürger in politischen Entscheidungsprozessen berücksichtigen zu können. Dazu werden onlinebasierte, forenähnliche Plattformen verwendet, auf denen die Bürger ihre Meinungen, Ideen und Kommentare in Textform äußern. Die Auswertung solcher Projekte erfolgt manuell und führt bei einer sehr hohen Anzahl an Textbeiträgen schnell zu einer Überlastung personeller Kapazitäten. In Kooperation mit technischen Dienstleistern und Städten im Rahmen des NRW Fortschrittskollegs Online-Partizipation haben wir einen Forschungsbedarf an automatisierten Verfahren in mehreren Bereichen identifizieren können.

Zunächst haben wir im Bereich des Argument Minings auf Grundlage des Beteiligungsprojekts *Tempelhofer Feld* und anhand von bestehenden Argumentationsmodellen ein geeignetes Modell für Online-Partizipationsverfahren identifiziert und mit drei Annotatoren einen Teil des Beteiligungsprojekts mit hoher Übereinstimmung annotiert. Anschließend haben wir auf diesem Datensatz die beiden Machine Learning-Aufgaben der Erkennung von argumentativem Textinhalt und der Klassifikation von Argumentationskomponenten bearbeitet. Dabei haben wir sowohl klassische Machine Learning-Verfahren mit Feature Engineering als auch Deep Learning-Verfahren ausführlich evaluiert.

Danach widmeten wir uns der Problemstellung, wie bei einer sehr großen Anzahl an Textbeiträgen automatisch ein Überblick über die Diskussionsthemen erstellt werden kann. Hierfür haben wir zunächst ein paar notwendige Vorarbeiten, wie die Erstellung eines neuen Verfahrens zur Grundformreduktion, durchgeführt. Nach grundlegenden Diskussionen über die für die Extraktion zu berücksichtigen Textinhalte und die Darstellungsform der extrahierten Themen, haben wir Verfahren der Themenextraktion auf mehrere Online-Partizipationsverfahren angewendet und ihre Ergebnisse diskutiert.

Abschließend haben wir uns mit der Aufgabe beschäftigt, wie automatisiert Verteilungen von demografischen Angaben über die an der Diskussion beteiligten Personen anhand von Textbeiträgen bestimmt werden können, um unterrepräsentierte Bevölkerungsschichten identifizieren zu können. Hierzu haben wir einen multilingualen Ansatz entwickelt, der für englische Texte in der PAN author profiling challenge 2016 für die Vorhersage des Geschlechts eines Autors den ersten Platz von 22 beteiligten Teams erreichen konnte.

CONTENTS

1	Introduction	1
1.1	Online Participation Processes	1
1.2	Research Goal	2
1.3	Publication List	3
1.4	Thesis Layout	6
2	Argument Mining in Online Participation Processes	7
2.1	Introduction into Argument Mining	7
2.2	Motivation	10
2.3	Argument Model for Online Participation	11
2.4	Corpus	12
2.5	Multi-level Classification Process	14
2.6	Classical Machine Learning Approach	16
2.6.1	Features	16
2.6.2	Argument Identification and Argument Classification	18
2.6.3	Differentiating Between Claim Types	24
2.7	Using Deep Learning for Argument Mining	26
2.7.1	Introduction	26
2.7.2	Architectures	27
2.7.3	Results	29
2.8	Future Work	30
3	Topic Extraction	33
3.1	Using Wiktionary to Reduce the Vocabulary Size with a New German Lemmatizer	33
3.2	Textual Similarity	36
3.2.1	Textual Similarity of Two Sentences	37
3.2.2	Textual Similarity in Obfuscated Texts	39
3.3	Extracting Topics in Online Participation Processes	42
3.3.1	Datasets	44
3.3.2	Topic Extraction	46
3.3.3	Topic Labeling	50
3.4	Future Work	52

4	Author Profiling	55
4.1	Introduction into Author Profiling	55
4.2	Use Cases for Online Participation Processes	56
4.3	Our Multilingual and Cross-domain Approach	56
4.4	Experiments on a German Online Participation Process	57
4.4.1	Dataset Creation and Annotation	57
4.4.2	Evaluation	57
4.5	Future Work	58
5	Conclusion and Outlook	61
6	Publications	65
6.1	What to Do with an Airport? Mining Arguments in the German Online Participation Project Tempelhofer Feld	67
6.2	Mining Arguments in Online Participation: Möglichkeiten und Grenzen manueller und automatisierter Inhaltsanalyse zur Erhebung von Argumentkomponenten	69
6.3	Text Mining für Online-Partizipationsverfahren: Die Notwendigkeit einer maschinell unterstützten Auswertung	71
6.4	IWNLP: Inverse Wiktionary for Natural Language Processing	73
6.5	HHU at SemEval-2016 Task 1: Multiple Approaches to Measuring Semantic Textual Similarity	75
6.6	Evaluating Safety, Soundness and Sensibleness of Obfuscation Systems	77
6.7	Understanding Trending Topics in Twitter	79
6.8	Exploring the Effects of Cross-Genre Machine Learning for Author Profiling in PAN 2016	81
	References	83
	List of Figures	95
	List of Tables	97

1

INTRODUCTION

This thesis begins with an introduction to online participation processes, describing the research goal, listing all publications, and outlining the layout.

1.1 Online Participation Processes

In the last couple of years, more and more cities in Germany have offered their citizens online-based discussion platforms on which the citizens are able to participate in decision-making processes (Gladitz et al., 2017), for instance on how the budget of the next fiscal period should be allocated. With the terms *online participation processes* and *online participation projects*, we refer to online platforms that include people into decision-making processes. They are usually forum-like and revolve around a predefined topic. The typical ways of participating usually include the possibility to propose ideas, to write comments, and to upvote and downvote texts of other users.

In this thesis, we focus on online participation in local political issues where the users are typically citizens who are affected by the topic of discussion. Our work is part of the PhD program *Online Participation*, funded by the North Rhine-Westphalian funding scheme *Fortschrittskollegs*.

In North Rhine-Westphalia, almost a third of the cities and municipalities already have experience with online participation, and participatory budgetings (*Bürgerhaushalte*) are most popular (Gladitz et al., 2017). On some of these platforms, citizens are engaged in extensive discussion, for instance by arguing for and against a certain topic from different points of view, which is sometimes called citizen-to-citizen-communication.

Besides politics, there are other application domains in which online participation is applied, such as policy drafting in universities (Escher et al., 2017) or companies. According to Rohmann and Schumann (2017), many of the online participation processes in companies fail because of a low participation rate.

In a political setting, online participation processes sometimes attract a lot of users who contribute a high amount of text content, while at other times the participa-

tion in these processes is very low. Figuring out which factors are responsible for the participation rate is still an open research question. For example, Zepic et al. (2017) investigate reasons for low participation rates based on expert interviews and a literature review and categorize them into five categories, including that the process is unknown to the citizens, that citizens are unable to participate, that they have no interest in participating, and that they refuse to participate.

In this dissertation, we are particularly interested in online participation processes in which a large number of texts is overwhelming for both the administration and the participants. In these cases, it becomes very difficult to quickly obtain an overview of the topics that have been and are being discussed. It is conceivable that the following effects described by Jones et al. (2004) will also occur in online participation processes: “(1) users are more likely to respond to simpler messages in overloaded mass interaction; (2) users are more likely to end active participation as the overload of mass interaction increases; and (3) users are more likely to generate simpler responses as the overloading of mass interaction grows.” Therefore, we would like to use natural language processing techniques to counteract these effects.

For a more detailed introduction to online participation, relevant text mining tasks, and the political and legal situation in Germany, we refer the reader to Liebeck et al. (2017).

1.2 Research Goal

Online participation processes with a high number of text contributions are difficult to analyze manually. As of now, all these processes are evaluated manually, which is time-consuming and costly. We would like to help with the analysis of these texts by assisting the manual analysis with natural language processing techniques.

It is important to us that our research addresses current problems in real online participation processes. In order to ensure this, we have conducted extensive discussions with several firms that provide technical solutions for online participation processes as well as with municipalities in which these processes are carried out. This allowed us to identify multiple problems, both during a process and for the evaluation after completion of a process.

For proposal-based online participation projects, the aim is to arrive at technical solutions for the automatic extraction of suggestions and their related justifications from text content written by citizens and for the creation of an overview of the discussed topics. For example, an output of such technical approaches could be that many citizens complain about a parking situation in a district or that more cultural programs should be offered. Additionally, information about the demography of the discussion participants was described as interesting in order to determine whether certain population strata are underrepresented.

For these desired technical solutions, several automated steps are conceivable which should ultimately lead to a reduction in manual work. In this thesis, we focus on the three areas of argument mining, topic extraction, and author profiling. Our research approach is interdisciplinary, combining the strengths of two disciplines: computer science and the social sciences.

The practical application of our research areas is becoming increasingly important as the state government in North Rhine-Westphalia is encouraging its municipalities

to carry out online participation processes, while the municipalities are, at the same time, legally required to evaluate the processes.

1.3 Publication List

This cumulative dissertation is based on multiple previously published papers which appeared in national and international peer-reviewed conferences, workshop proceedings, and journals. The following list comprises all publications that were published during the pursuit of the PhD degree. The publications that make up the core contributions to the automated discussion analysis in the application domain of online participation are included in Chapter 6. The sections that are built upon these contributions are also indicated in the publication list.

2017:

1. Matthias Liebeck, Katharina Esau, and Stefan Conrad (2017). “Text Mining für Online-Partizipationsverfahren: Die Notwendigkeit einer maschinell unterstützten Auswertung”. In: *HMD Praxis der Wirtschaftsinformatik* 54.4. Schwerpunktheft „Online Participation“, pp. 544–562.
Contributions: Matthias Liebeck designed the experiments, evaluated features for the machine learning tasks, and implemented the topic modeling approach. The manuscript was prepared jointly by Matthias Liebeck and Katharina Esau.
Sections: 2.6.2.3, 3.3.2
Status: Published.
2. Pascal Hirmer, Tim Waizenegger, Ghareeb Falazi, Majd Abdo, Yuliya Volga, Alexander Askinadze, Matthias Liebeck, Stefan Conrad, Tobias Hildebrandt, Conrad Indiono, Stefanie Rinderle-Ma, Martin Grimmer, Matthias Kricke, and Eric Peukert (2017). “The First Data Science Challenge at BTW 2017”. In: *Datenbank-Spektrum* 17.3, pp. 207–222.
Contributions: Matthias Liebeck and Alexander Askinadze participated in the BTW 2017 Data Science Challenge and achieved 2nd place which was awarded with 300 euros prize money. The third section of the manuscript was prepared jointly by Alexander Askinadze and Matthias Liebeck under the supervision of Stefan Conrad.
Status: Published.
3. Roland Kahlert, Matthias Liebeck, and Joseph Cornelius (2017). “Understanding Trending Topics in Twitter”. In: *Datenbanksysteme für Business, Technologie und Web (BTW 2017) - Workshopband*. GI, pp. 375–384.
Contributions: The research was conducted jointly by Matthias Liebeck, Roland Kahlert and Joseph Cornelius. Matthias Liebeck designed the experiments, especially for the *Topic Detection* section (which were implemented by Joseph Cornelius), and supervised Roland Kahlert. The manuscript was prepared jointly by Matthias Liebeck and Roland Kahlert.
Sections: 3.3
Status: Published.

4. Katharina Esau, Matthias Liebeck, and Christiane Eilders (2017). “Mining Arguments in Online Participation: Möglichkeiten und Grenzen manueller und automatisierter Inhaltsanalyse zur Erhebung von Argumentkomponenten”. In: *Polkomm 2017 - „Disliken, diskutieren, demonstrieren – Politische Partizipation im (Medien-)Wandel“*.

Contributions: Matthias Liebeck performed simulations for the evaluation in the manuscript. Katharina Esau and Matthias Liebeck contributed equally to the preparation of the manuscript.

Sections: 2.6.2.2

Status: Published.

2016:

5. Matthias Liebeck, Pashutan Modaresi, Alexander Askinadze, and Stefan Conrad (2016b). “Pisco: A Computational Approach to Predict Personality Types from Java Source Code”. In: *Working notes of FIRE 2016 - Forum for Information Retrieval Evaluation*, pp. 43–47.

Contributions: Matthias Liebeck and Pashutan Modaresi applied pair programming to create the code. The manuscript was prepared jointly by Matthias Liebeck, Pashutan Modaresi, and Alexander Askinadze.

Status: Published.

6. Pashutan Modaresi, Matthias Liebeck, and Stefan Conrad (2016b). “Neural Classification of Linguistic Coherence Using Long Short-Term Memories”. In: *Proceedings of the 8th Annual Meeting of the Forum on Information Retrieval Evaluation*. FIRE ’16. ACM, pp. 28–31.

Contributions: The research and the preparation of the manuscript was done jointly by Pashutan Modaresi and Matthias Liebeck under the supervision of Stefan Conrad.

Status: Published.

7. Pashutan Modaresi, Matthias Liebeck, and Stefan Conrad (2016a). “Exploring the Effects of Cross-Genre Machine Learning for Author Profiling in PAN 2016”. In: *Working Notes of CLEF 2016 - Conference and Labs of the Evaluation Forum*, pp. 970–977.

Contributions: Matthias Liebeck contributed with the feature engineering and the implementation of features. Pashutan Modaresi and Matthias Liebeck contributed equally to the preparation of the manuscript.

Sections: 4.3

Status: Published.

8. Matthias Liebeck, Pashutan Modaresi, and Stefan Conrad (2016c). “Evaluating Safety, Soundness and Sensibleness of Obfuscation Systems”. In: *Working Notes of CLEF 2016 - Conference and Labs of the Evaluation Forum*, pp. 920–928.

Contributions: Matthias Liebeck contributed in the design of the evaluation methods for the dimensions *soundness* and *sensibleless*. The manuscript was prepared jointly by Matthias Liebeck and Pashutan Modaresi.

Sections: 3.2.2

Status: Published.

9. Matthias Liebeck, Katharina Esau, and Stefan Conrad (2016a). “What to Do with an Airport? Mining Arguments in the German Online Participation Project Tempelhofer Feld”. In: *Proceedings of the Third Workshop on Argument Mining (ArgMining2016)*. Association for Computational Linguistics, pp. 144–153.
Contributions: The annotation scheme was developed jointly by Matthias Liebeck and Katharina Esau. Matthias Liebeck implemented the machine learning approach and prepared the manuscript.
Sections: 2.3, 2.4, 2.5, 2.6.1, 2.6.2.1
Status: Published.
 10. Matthias Liebeck, Philipp Pollack, Pashutan Modaresi, and Stefan Conrad (2016d). “HHU at SemEval-2016 Task 1: Multiple Approaches to Measuring Semantic Textual Similarity”. In: *Proceedings of the 10th International Workshop on Semantic Evaluation*, pp. 595–601.
Contributions: The research was conducted jointly by Matthias Liebeck, Philipp Pollack, and Pashutan Modaresi. Matthias Liebeck contributed the *Overlap method*, participated in the feature engineering, and supervised Philipp Pollack. The manuscript was prepared by Matthias Liebeck.
Sections: 3.2.1
Status: Published.
- 2015:**
11. Matthias Liebeck and Stefan Conrad (2015). “IWNLP: Inverse Wiktionary for Natural Language Processing”. In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Association for Computational Linguistics, pp. 414–418.
Contributions: The research and the preparation of the manuscript was done entirely by Matthias Liebeck under the supervision of Stefan Conrad.
Sections: 3.1
Status: Published.
 12. Matthias Liebeck (2015a). “Ansätze zur Erkennung von Kommunikationsmodi in Online-Diskussionen”. In: *Proceedings of the 27th GI-Workshop Grundlagen von Datenbanken*, pp. 42–47.
Contributions: The research and the preparation of the manuscript was done entirely by Matthias Liebeck.
Status: Published.
 13. Matthias Liebeck (2015b). “Aspekte einer automatischen Meinungsbildungsanalyse von Online-Diskussionen”. In: *Datenbanksysteme für Business, Technologie und Web (BTW 2015) - Workshops und Studierendenprogramm*, pp. 203–212.
Contributions: The research and the preparation of the manuscript was done entirely by Matthias Liebeck.
Status: Published.

1.4 Thesis Layout

This thesis comprises three main chapters. For each of these sections, we provide a brief overview over the section's content:

Chapter 2: Argument Mining in Online Participation Processes

In Chapter 2, we focus on argument mining. We introduce an argument model suitable for online participation projects and describe our annotated corpus. Afterwards, we focus on two machine learning tasks and evaluate our classical machine learning approach with feature engineering and deep learning approaches.

Chapter 3: Topic Extraction

In Chapter 3, we focus on extracting discussion topics from online participation projects with a high number of text contributions. We start by introducing a new German lemmatizer and address textual semantic similarity. Afterwards, we use two approaches to extract topics. We discuss their results and outline a topic labeling approach.

Chapter 4: Author Profiling

In Chapter 4, we address author profiling, which is the task of predicting demographic attributes of users based on their text content. We present our multi-lingual author profiling approach (Modaresi et al., [2016a](#)) and apply it to a German online participation project.

2

ARGUMENT MINING IN ONLINE PARTICIPATION PROCESSES

In this chapter, we focus on mining arguments from German online participation projects. After an introduction to argument mining, we describe why argument mining is useful for online participation projects. Afterwards, we describe an argument model that is suitable for online participation projects. Then, we describe our annotated dataset and evaluate our approaches to the two machine learning tasks of automatically identifying argumentative content and classifying argument components in detail. The code for our argument mining pipeline and our deep learning experiments is open-sourced on GitHub¹.

2.1 Introduction into Argument Mining

On the Internet, many people discuss and argue about a variety of topics. When purchasing a product online, for example, a camera, reasons for or against the purchase can be read in advance in product reviews. These reasons are usually not available as a compact list but are scattered in continuous text. An example of the application domain online participation is a discussion about whether a playground should be constructed, where reasons for or against the construction are expressed. For a small number of text documents, such reasons can be quickly read manually. However, if the number of text documents is high, e.g., several hundred or even several thousand, a significant investment of tedious manual effort is required to sift through the text documents in order to condense and summarize their content. For precisely such cases, an automated analysis system would be desirable. The development of such techniques falls within the area of argument mining.

Argument mining is a trending research field that focuses on the identification and classification of argument components in texts and on the extraction of their relations. Until today, argument mining has been used on texts in different languages (mostly in

¹<https://github.com/Liebeck/ArgMining>

English and German, but also in Greek (Goudas et al., 2014), Japanese (Reisert et al., 2014), Chinese (Chow, 2016; Li et al., 2017), and in Spanish (Fierro et al., 2017)) and in several text domains, including the legal domain (Palau and Moens, 2009; Houy et al., 2013), online participation (Park and Cardie, 2014; Liebeck et al., 2016a), persuasive essays (Stab and Gurevych, 2014b), news (Eckle-Kohler et al., 2015), artificially created microtexts (Peldszus and Stede, 2013), reviews (Schneider and Wyner, 2012; Rajendran et al., 2016), social media (Goudas et al., 2014; Addawood and Bashir, 2016), web discourse (Habernal and Gurevych, 2015; Habernal and Gurevych, 2017), and scientific publications (Kirschner et al., 2015; Green, 2015).

There are three common tasks in argument mining which are performed one after the other in a real-world application:

1. Argument identification

As a first step, non-argumentative text content is separated from argumentative text content.

Example:

Hello, my name is Matthias.
Cigarettes are bad for your health.
Studies show that cigarettes can
cause cancer.
Thanks for reading my text.



Hello, my name is Matthias.
Cigarettes are bad for your health.
Studies show that cigarettes can
cause cancer.
Thanks for reading my text.

Legend:

— argumentative content

... non-argumentative content

2. Argument classification

Argumentative text content can then be classified according to a specified argument model. In the following example, the *claim-premise family* is used as argument model which comprises claims (“*a controversial statement that is either true or false*” (Stab and Gurevych, 2014a)) and premises (reasons that either support or attack a claim).

Example:

Cigarettes are bad for your health.
Studies show that cigarettes can
cause cancer.



Cigarettes are bad for your health.
Studies show that cigarettes can
cause cancer.

Legend:

— argumentative content

Legend:

--- claim

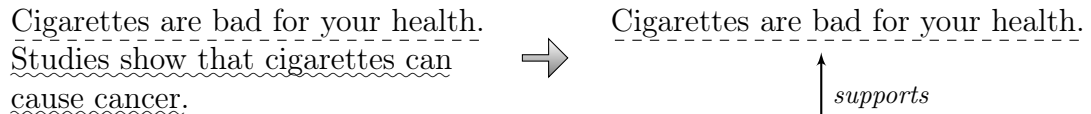
~ ~ ~ premise

3. Argument linking

After individual argument components have been determined, relations between these components can be identified (e.g., a premise supporting a claim).

Example:

Cigarettes are bad for your health.
 Studies show that cigarettes can
 cause cancer.



Legend:

--- claim

~~~ premise

Cigarettes are bad for your health.  
 Studies show that cigarettes can  
 cause cancer.

Legend:

--- claim

~~~ premise

Argument mining is applied to text documents. In order to measure the effectiveness of a machine learning approach, an argument mining system is benchmarked on an annotated corpus. Such corpora comprise text documents that were previously annotated according to an argument model selected by the respective authors of the respective publications. Currently, the argument mining community has not agreed on a single argumentation model. The choice of the argument model depends on a variety of factors, including the text domain (e.g., legal field vs. web discourse) and the way in which different texts are connected (monologue texts vs. discourse with reciprocity). Especially for different types of web content, we agree with Habernal et al. (2014) that there is no single argumentation model that fits every application domain and that the choice of an argumentation model depends on the respective text domain.

However, most argument models contain two common components called *claim* and *premise*. There are different definitions of a claim: Toulmin (2003, p. 90) describes a claim as the statement of the argument “*whose merits we are seeking to establish.*” Stab and Gurevych (2014a) define a claim as a “*controversial statement that is either true or false*” and that it “*should not be accepted by readers without additional support.*” Palau and Moens (2009) use the term claim as “*an idea which is either true or false.*” Premises are reasons that either support or attack a claim.

Argument mining corpora are rare due to the time-consuming nature of annotating a corpus, including the creation of annotation guidelines, annotator training, measurement of inter-annotator agreement, and finally the annotation process itself. For a detailed literature review of previous work in the field of argument mining and of existing corpora, we refer the reader to the extensive works from Lippi and Torroni (2016) and Habernal and Gurevych (2017). At this point, we briefly mention only a few notable corpora:

Stab and Gurevych (2014a) were most influential in the choice of our argumentation model. They annotated a corpus of 90 persuasive essays with a new three-part argumentation model in the claim-premise family, comprising *major claims*, *claims*, and *premises*. In addition, they followed the approach employed by Peldszus and Stede (2013) and annotated argumentative relations in the form of supports and attacks. In Stab and Gurevych (2014b), they published their machine learning approach for determining argument components (the identification of argumentative sentences and argument components were merged into a 4-class classification problem (none, major

claims, claims, and premises)) and argument relations. They achieved macro-averaged F_1 scores (see Section 2.5) of 72.6% and 72.2%, respectively.

Peldszus and Stede (2016) released a corpus of 112 microtexts that each contains approximately five discourse segments. Their corpus is notable for being the first parallel argument mining corpus which was originally written in German and then professionally translated into English.

With Liebeck et al. (2016a), we started to address the research gap of automatically mining arguments for German political online participation projects. Further details of our work are provided in the rest of this section.

A new subtask of determining which one of two given arguments about the same topic and stance is more convincing was introduced by Habernal and Gurevych (2016) by releasing a crowdsourced corpus of almost 17,000 argument pairs. Their contribution is also notable for being one of the first publications in the argument mining community that uses recurrent neural networks for classification tasks.

2.2 Motivation

In the workflow of many online participation processes, the citizens first engage in online discussion and submit their suggestions or ideas. Subsequently, their text contents are read manually and summarized in the form of a list of proposals. Each of these proposals is checked to determine whether it can be implemented in consideration of legal and financial possibilities. As a final step, this list of proposals will be presented to politicians at a meeting of the municipal council and each of the proposals can be voted on. For example, such a list may contain that many citizens may wish to see a revision of the parking fees in the city center or that many citizens support the planting of new trees in a park.

We have a very practical point of view on what kind of results we want to automatically extract by using argument mining technologies on online participation processes. We want to automatically support the person that creates the list of the proposals since the manual summarization of the citizens' text contents, particularly in the case of participation processes with several thousand text contents, is complex and, therefore, takes a long time.

Politicians have a good basis for discussion when they know the concerns of their citizens in the form of proposals and justifications. An automated summary of the discussion should answer the following three questions, which we formulated in Liebeck et al. (2016a):

1. Which suggestions and ideas are contributed by the citizens? What do people want to be built or decided upon?
2. Which reasons do the citizens provide for the realization of their suggestions? How do they argue for and against these ideas?
3. How many people in the discussion say that they agree or disagree with them?

From our point of view, we are most interested in mining suggestions. Although the third question can partially be answered by taking upvotes and downvotes into account, citizens also tend to express their agreement and disagreement in text form,

which must be (automatically) interpreted in order to obtain a full overview of all expressed opinions.

2.3 Argument Model for Online Participation

In Liebeck et al. (2016a), we presented our first approach on argument mining for online participation projects that focus on gathering proposals in the form of options for actions or decisions (e.g., “We should build a playground.” or “Should dogs be banned from the park?”). We looked at the online participation project *Tempelhofer Feld*² in order to determine an argument scheme that is suited for online participation and started to apply existing argument models to our dataset: (i) Toulmin’s model (Toulmin, 1958) and (ii) the claim-premise family.

By analyzing the participation project in the scope of Liebeck et al. (2016a), we identified the following behavior: (i) we have discourse between different users; (ii) attacks on logical conclusions are rather rare; (iii) users frequently express their wishes; (iv) users participate by providing reasons for and against other suggestions; (v) suggestions cannot be classified as true or false; and (vi) suggestions can be accepted without additional support. Taking this behavior into account, existing models did not fit exactly. Based on the user behavior and our practical point of view, we also quickly realized that we needed to distinguish between at least three different categories of argumentative content.

For the choice of our argument model, the claim-premise family was a good starting point. The modifications from Stab and Gurevych (2014a) into a three-part model were most influential for our argument model. We developed an argument model with three argument components for proposal-based online participation projects in Liebeck et al. (2016a) and made the following definitions:

- **major position:** Major positions are options for actions or decisions that occur in the discussion (e.g., “*We should build a playground with a sandbox.*” or “*The opening hours of the museum must be at least two hours longer.*”). They are most often someone’s vision of something new or of a policy change. If another user suggests a modified version by changing some details, the new suggestion is a new major position (e.g., “*We should build a playground with swings.*”). In our practical view, major positions are unique suggestions from citizens that politicians can decide on.
- **claim:** A claim is a pro or contra stance toward a major position (e.g., “*Yes, we should definitely do that!*”). In our model, claims are text passages in which users express their personal positions (e.g., “*I dislike your suggestion.*”). For a politician, the text content of a claim in our definition does not serve as a basis for decision making because claims do not contain justifications upon which decisions can be backed up. The purpose of mining these claims is a conversion into two numbers that indicate how many citizens are for or against a suggestion.
- **premise:** The term premise is defined as a reason that attacks or supports a major position, a claim or another premise. Premises are used to make an

²<https://tempelhofer-feld.berlin.de/>

argumentation comprehensible for others, by reasoning why a suggestion or a decision should be realized or why it should be avoided (e.g., “*This would allow us to save money.*”). We use the term premise in the same way as the claim-premise model and as Toulmin with *grounds*.

We do not evaluate if a reason is valid. We only determine the user’s intent: If an annotator perceives that a user is providing a reason, we annotate it as such (see Section 2.4). Otherwise, we would have to evaluate each statement on a semantic level. For example, if a user argues that a suggestion violates a building law, the annotators would need to check this statement. A verification of all reasons for correctness would require too much expertise from annotators or a very large knowledge database. In our application domain, we leave the evaluation to human experts who advise politicians.

Our argument model is illustrated in Figure 2.1.

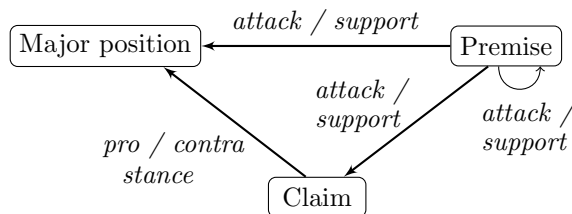


Figure 2.1: Our argumentation model for political online participation (Liebeck et al., 2016a)

2.4 Corpus

In Liebeck et al. (2016a), we applied our argument model to the Tempelhofer Feld project. By the time we annotated the corpus, the Tempelhofer Feld project consisted of 340 ideas and 1389 comments. The Tempelhofer Feld was chosen as the corpus for our research since the participation rate is high and its text content is licensed under the Creative Commons License, which allows us to redistribute the corpus along with our annotations.

Figure 2.2 shows a few German examples and their translations into English from the Tempelhofer Feld project, to which we have applied our argument model. The discussion thread began with one citizen proposing to build children’s playgrounds ① and stating his reason for the proposal ②. Another citizen replied with his stance ③ and backed it up ④. During the discussion, two modified ideas were posted ⑤,⑥.

We did not annotate relations between argument components. Since the Tempelhofer Feld consists of proposals that can be commented on, we have discourse between the users, meaning one user can refer to the argumentation of another user, and, therefore, argument relations can link between argument components of the same document or between argument components of different documents. Especially for proposals with more than 30 comments, this leads to a large number of possible relations that require examination. In addition to the challenge posed by quadratic complexity, it is difficult to create subjective annotation guidelines that allow annotations without

- | | |
|--|---|
| ① [<u>major position</u> : <i>Wir hoffen, dass Kinderspielplätze zukünftig mit dem THF-Gesetz im Aussenbereich vereinbar sind.</i>]
② [<u>premise</u> : <i>Als Familienvater von 2 Kindern fehlt mir auf dem Feld der ein oder andere Kinderspielplatz.</i>]
③ [<u>claim</u> : <i>Ich sehe das Anlegen von einfachen Spielplätzen eher kritisch und das obwohl ich selber Kinder habe.</i>]
④ [<u>premise</u> : <i>Im Umkreis des Feldes sind bereits viele zum Teil sehr schöne Spielplätze vorhanden. Dafür muss meiner Ansicht nach das Feld nicht bebaut werden.</i>]
⑤ [<u>major position</u> : <i>zum beispiel könnte man eine art mini flughafen machen oder so.</i>]
⑥ [<u>major position</u> : <i>was ich auch gut fände, wäre zum beispiel eine kletterwand in form des luftbrücken-denkmals.</i>]
⑦ [<u>claim</u> : <i>ich finde die idee mit dem spielplatz gut und sinnvoll, [...]</i> | ① [<u>major position</u> : <i>We hope that children's playgrounds in the open-air space will be compatible with the ThF law in the future.</i>]
② [<u>premise</u> : <i>As a father of two children I am missing one or two children's playgrounds on the field.</i>]
③ [<u>claim</u> : <i>I am rather critical of the creation of simple playgrounds even though I have children myself.</i>]
④ [<u>premise</u> : <i>In the vicinity of the field there are already plenty of playgrounds, some of which are very nice. Therefore, I don't think it's necessary to build another one on it.</i>]
⑤ [<u>major position</u> : <i>For example a miniature airport could be built.</i>]
⑥ [<u>major position</u> : <i>I would also like a climbing wall shaped like the Airlift Memorial.</i>]
⑦ [<u>claim</u> : <i>[Having] a playground would be a good and sensible idea. [...]</i> |
|--|---|

Figure 2.2: Examples of our argumentation model applied to excerpts of the Tempelhofer Feld

overinterpreting argument relations. This is especially the case for argument components that require domain knowledge in order to determine whether a premise is attacking or supporting. We decided not to annotate argument relations in order to avoid wrong and incomplete annotations.

We want to point out that it is not impossible to annotate argument relations. The research design of Peldszus and Stede (2013) allowed for annotating argument relations within document boundaries since their text content was tailor-made to contain multiple discourse segments that relate to each other. However, since we dealt with genuine web content containing discourse across document boundaries, the annotation of argument relations turned out to be much more difficult than we initially expected.

As a result, we decided to further divide claims into *pro claims* and *contra claims* and annotated four different argument components (major claim, pro claim, contra claim, and premise) in our *THF Airport ArgMining Corpus*. These two subtypes are identified by the most dominant position of a claim based on the wording and the content. For example, claim ③ from Figure 2.2 contains a contra stance and claim ⑦

is formulated positively and, therefore, is considered to be pro claim.

We created annotation guidelines and used the brat rapid annotation tool (Stenetorp et al., 2012) for the annotation process. The annotators were instructed to use freely assignable spans. This allows for multiple annotations per sentence and annotations spanning multiple sentences. In total, we annotated a subset of 72 proposals and 575 comments with three annotators. Further details about the corpus’ class distribution are discussed in the next section. We measured our inter-annotator agreement with *Krippendorff’s unitized alpha* α_u (Krippendorff, 2004) on a subset of the corpus and achieved $\alpha_u = 92.4\%$ for argumentative vs. non-argumentative spans and $\alpha_u = 78.0\%$ for argument components, which is in line with previous annotation studies³.

Another approach we could have pursued to annotate our corpus would have been crowdsourcing, e.g., Amazon’s Mechanical Turk⁴. In crowdsourcing, the annotation process is broken down into multiple single tasks, so-called *human intelligent tasks*, which are solved by humans, called *workers*, who receive a small financial reward for their work. Crowdsourcing has attracted a lot of attention in a variety of natural language processing tasks, for instance in annotating the convincingness of arguments (Habernal and Gurevych, 2016), in creating translations (Zaidan and Callison-Burch, 2011) and language resources (Mohammad and Turney, 2013), and in the evaluation of topic models (Chang et al., 2009). The characteristic of outsourcing the annotation process into a crowd makes crowdsourcing especially attractive for the annotation of large datasets. We decided against the use of crowdsourcing since we first wanted to gain experience with annotations from experts.

2.5 Multi-level Classification Process

The annotations from the THF Airport ArgMining Corpus allow us to evaluate two machine learning tasks: (i) argument identification by treating all annotated text spans as argumentative text content and text parts that have no annotations are considered to be non-argumentative text; (ii) argument classification of the argumentative text sentences that have exactly one annotation.

For now, we use individual sentences as the granularity level of our classification tasks. We treat both above-mentioned tasks as a multi-level classification process which is illustrated in Figure 2.3. The identification of argumentative text parts is called *subtask A*. Since the number of pro claims and contra claims is very low compared to the other two classes (see Table 2.1), we decided to group pro claims and contra claims as *claims* for a ternary classification problem denominated as *subtask B*. In the following step, called *subtask C*, we can further differentiate claims into their two subtypes.

Now, we want to outline the structure of our evaluation framework. First of all, the annotated corpus has been randomly split into an 80% training set and 20% test set for subtasks A and B. The exact distribution of both sets is listed in Table 2.1. The parameters of our features and of our classifiers are selected by a grid search on a 10-fold cross-validation on the training set. The test set is only touched once per

³Habernal and Gurevych (2017) provide a good overview of inter-annotator agreement scores in the field of argument mining.

⁴<https://www.mturk.com>

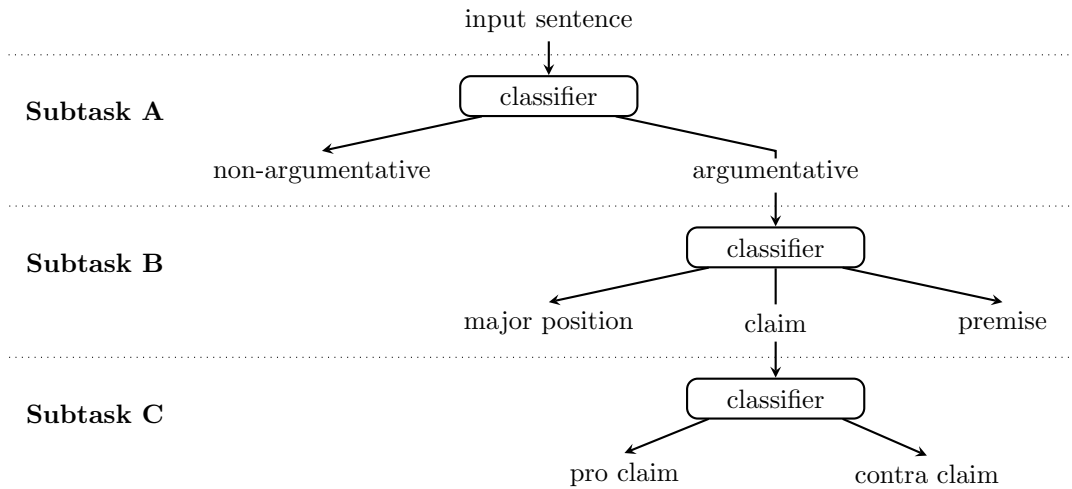


Figure 2.3: Multi-level classification process for an input sentence (Liebeck et al., 2017)

| Subtask | Training set | Test set |
|-----------|---|--|
| Subtask A | 1667 argumentative
280 non-argumentative | 411 argumentative
75 non-argumentative |
| Σ | 1947 sentences | 486 sentences |
| Subtask B | 951 premises
399 major positions
242 claims | 219 premises
110 major positions
69 claims |
| Σ | 1592 sentences | 398 sentences |
| Subtask C | 185 pro claims
57 contra claims | 56 pro claims
13 contra claims |
| Σ | 242 sentences | 69 sentences |

Table 2.1: Class distribution for all three subtasks in the THF Airport ArgMining Corpus

feature to calculate the results. Therefore, no knowledge from the test set influences or *bleeds*⁵ into the parameter selection.

The design of our evaluation framework is influenced by our previous work in author profiling (Modaresi et al., 2016a) and experience gained in the domain of personality recognition (Liebeck et al., 2016b). Our framework comprises two main scripts named *grid search* and *evaluate*. As the name suggests, the grid search script is responsible for determining the best parameters for the features and the classifiers on the training set by a grid search on a cross-validation. The resulting parameters and random states are serialized into a config file. The evaluate script then reads the config file, sets all the specified parameters and random states for the classification pipeline, and predicts reproducible results for the test set. The resulting predictions and confusion matrix are saved in order to make an error analysis possible.

As evaluation metric, we chose the macro-averaged F_1 score. Since we have a class

⁵Vanhoucke (2017) uses the term *bleed* to describe this effect.

imbalance, accuracy would not be a good choice as evaluation metric since a classifier that always predicts the majority class would achieve high scores. The F_1 score for a class c is defined as the harmonic mean of precision and recall of class c :

$$F_1(c) = 2 \cdot \frac{\text{precision}(c) \cdot \text{recall}(c)}{\text{precision}(c) + \text{recall}(c)} \quad (2.1)$$

In order to weight every class $c \in C$ the same, the macro-averaged F_1 score calculates the average of all F_1 scores:

$$\text{macro-averaged } F_1 = \frac{1}{|C|} \sum_{c \in C} F_1(c) \quad (2.2)$$

2.6 Classical Machine Learning Approach

We will now focus on a classical machine learning approach with feature engineering for argument mining. We start by describing our features and reporting already published results from Liebeck et al. (2016a), Esau et al. (2017), and Liebeck et al. (2017). Then, we evaluate additional features in the scope of this thesis and report their results as well.

2.6.1 Features

We experimented with a multitude of features for our machine learning algorithms:

- **Lexical features:**

- word unigrams and bigrams (raw, lowercased, lemmatized with IWNLP (Liebeck and Conrad, 2015) (see Chapter 3.1), and combinations thereof)
- character n-grams
- token shapes (e.g., representing ‘Berlin’ with ‘Xxxxx’)

- **Structural and syntactic features:**

- L2-normalized part-of-speech tag distribution of the STTS tags (Schiller et al., 1999) and the more coarse universal part-of-speech tagset (Petrov et al., 2012)
- L2-normalized distribution of the dependencies in the TIGER annotation scheme (Albert et al., 2003)
- punctuation features:
 - * relative count of commas and dots
 - * last token of a sentence as a one-hot encoding (‘.’, ‘!’, ‘?’, ‘OTHER’)
- number of URLs
- index number of the sentence in the document
- number of tokens in the sentence

- number of named entities
- depth of a text comment in the tree hierarchy of the discussion
- **Topic features:**
 - LDA (Blei et al., 2003) topic distribution trained on a German Wikipedia dump
 - LDA topic distribution trained on the whole THF corpus
- **Sentiment features:**
 - SentiWS (Remus et al., 2010) (averaged polarity and discretized distribution)
- **Embedding features:**
 - **Word embeddings:** Word embeddings (Mikolov et al., 2013) are a commonly used way to overcome the sparsity problem that occurs when dealing with a big vocabulary of size $|V|$ in a bag-of-words (BoW) model. Instead of representing a word with a very long vector only containing zeros except for one dimension, word embeddings allow for a representation in a continuous vector space with a much lower dimension $k \ll |V|$ (for instance $k \in \{100, 200, 300\}$) that also captures semantics by embedding semantically related words closer together than words that are not semantically related. Since word embeddings have evolved to be a standard representation for words in the last three years, we trained word embeddings with gensim (Řehůřek and Sojka, 2010) on a German Wikipedia dump.

Aside from a lower dimensionality, word embeddings do have an additional advantage over classical BoW models since out-of-vocabulary (OOV) terms (words that were not present during the creation of the vocabulary in the training phase of a classifier) cannot be accurately dealt with using BoW models. Word embeddings, however, can deal with an OOV term w in a classification task as long as w has a vectorial representation in the word embedding. Since semantically related words should be close to each other in a word embedding, an OOV term a should behave similarly to a word synonym b that is present in the training set.

One disadvantage of word embeddings (that standard BoW models also have) is that words are each usually represented by exactly one embedding, although the word might have more than one meaning. However, this issue is already being addressed by Fadaee et al. (2017) who proposed a method to create multiple word embeddings per word with the intention of representing each meaning with a separate vector.
 - **Character embeddings:** In Liebeck et al. (2017), we evaluated the common way to represent words with word embeddings. However, this approach has the disadvantage of dealing with out-of-vocabulary words (in terms of the embedding vocabulary) since they do not have a vector representation. Instead of looking up a vectorial representation based on the full lexical form

of the word, Bojanowski et al. (2017) proposed a new approach that incorporates subword information. In their approach, each word is represented by a bag of character n-grams and the word embedding of a given word is the sum of the word embeddings for the containing n-grams. As a result, it is possible to obtain word embeddings for unknown words. Using these subword representations, Bojanowski et al. (2017) were able to improve upon several baselines of multiple tasks (and in several languages) that do not use subword information. Bojanowski et al. (2017) reported that n-gram sizes of length 5 or 6 are good choices to capture subword information for German compound nouns and that smaller n-gram sizes resulted in slightly smaller results. In our evaluation in Section 2.6.2.3, we experimented with a range of n-gram sizes and did not observe the same effect.

2.6.2 Argument Identification and Argument Classification

In this subsection, we discuss the classification results for subtask A and subtask B using a classical machine learning approach including feature engineering. We begin with our initial approach from Liebeck et al. (2016a). Afterwards, we report our observations of the effects of changing the training size from our work in Esau et al. (2017). Subsequently, we outline our experiments from Liebeck et al. (2017) on word embeddings. Finally, we experiment with additional features and report their results.

2.6.2.1 Initial Approach

In Liebeck et al. (2016a), we presented our corpus and our initial machine learning approach. We evaluated three different classifiers: support vector machines (SVM) (Cortes and Vapnik, 1995) with an RBF kernel, k-nearest neighbor (k-NN), and random forests (RF) (Breiman, 2001). As features, we evaluated lexical features (unigrams and bigrams), part-of-speech tag and dependency distributions (denoted as *grammatical features*), and structural features. The classification results of our initial approach are listed in Table 2.2.

| Feature Set | Subtask A | | | Subtask B | | |
|------------------------------------|--------------|-------|-------|--------------|-------|-------|
| | SVM | RF | k-NN | SVM | RF | k-NN |
| Unigram | 65.99 | 68.13 | 61.00 | 64.40 | 59.41 | 40.30 |
| Unigram, lowercased | 66.69 | 64.53 | 62.26 | 65.32 | 53.35 | 38.25 |
| Bigram | 41.79 | 50.48 | 16.25 | 46.62 | 50.42 | 11.51 |
| Grammatical | 55.88 | 52.24 | 48.52 | 59.54 | 47.89 | 46.81 |
| Unigram + Grammatical | 69.77 | 58.39 | 64.87 | 68.50 | 57.13 | 35.90 |
| Unigram + Grammatical + Structural | 67.50 | 61.14 | 54.07 | 65.99 | 59.46 | 47.27 |

Table 2.2: Macro-averaged F_1 scores for subtask A and subtask B in Liebeck et al. (2016a)

The best results for both subtasks were at almost 70% macro-averaged F_1 and were both achieved by an SVM with unigrams and distributions of part-of-speech tags and dependencies. To put these numbers into perspective, at the time of publication, the quality of the classification results was comparable to other studies that also used

classical machine learning approaches on their respective datasets. Upon analyzing the results for subtask B and taking a closer at the confusion matrix of the classification results (Liebeck et al., 2016a), we can see that premises can be classified quite well but major positions are often wrongly classified as premises.

2.6.2.2 Effects of the Training Size

In our THF Airport ArgMining Corpus, we only annotated about one-third of the whole online participation process due to time and budgetary constraints. A valid question regarding the size of our dataset or in any annotation process in general is: How much data do we need to annotate for machine learning in order to achieve good classification results? This question is difficult to answer because of several influential factors, such as (i) “*How representative is the annotated subset of the whole dataset?*”, (ii) “*How high is the annotation quality?*”, and (iii) “*Does the intended classifier require a somewhat minimum number of training instances to be able to learn?*”.

In Esau et al. (2017), we focused on a very similar question: How good would our result have been if we would have only annotated a smaller percentage of our dataset? To answer this question, we decided for the following evaluation setup: Given a fixed test dataset of annotated argument components (the same one we used earlier for subtask B), we measured how the quality of our predictions change if the training set is a randomized subset of the initial training set. For each subset size, we repeated the experiment three times with different random states. In Esau et al. (2017), we reported our results as an inter-annotator agreement score between two annotators, regarding the machine prediction as one annotator and treating the manually annotated gold-standard from three human annotators as the second annotator. In this thesis, we additionally plot an empirical curve of training size vs. macro-averaged F_1 score of the normalized *unigram + grammatical* feature in Figure 2.4.

First of all, we can observe that the results of the SVM are always better than KNN. If we increase the number of training data, the results of the SVM improve further. The trend suggests that more training data could possibly increase the results even more. In addition, the standard deviation (not shown here) decreases with an increasing number of training data. Only at 60% of the training data, there is a slight variation. The figure also shows that KNN clearly behaves differently since about 30% of the training data already leads to a saturation of the results. Since the data points are the same for both classifiers, KNN is not able to use the additional training data as effective as the SVM.

2.6.2.3 Further Evaluations

For our work in Liebeck et al. (2017), we ported our feature engineering from C# to Python and improved the overall reusability of our implementation by using a Docker⁶ container. In Liebeck et al. (2017), we evaluated word embeddings with dimensionality $k \in \{100, 200, 300\}$ as a standalone feature and in combination with unigrams, as reported in Table 2.3. Unigrams achieve better results than word embeddings alone. By combining both features, the results can be increased slightly, but it should be noted that the results for subtask B vary, depending on the dimensionality of the embeddings.

⁶<https://docker.com>

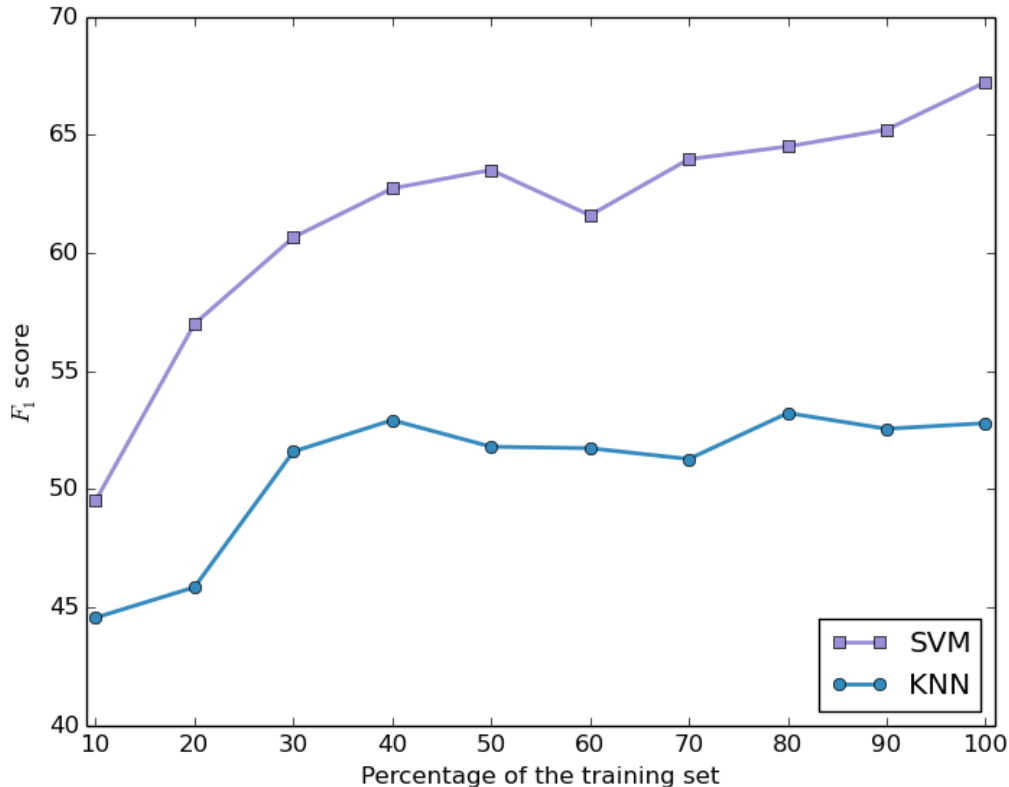


Figure 2.4: Empirical curve of training size vs. macro-averaged F_1 score

| Features | Subtask A | | | Subtask B | | |
|----------------------|--------------|-----------|-----------|-----------|--------------|-----------|
| | $k = 100$ | $k = 200$ | $k = 300$ | $k = 100$ | $k = 200$ | $k = 300$ |
| Unigram | 64.71 | | | 64.3 | | |
| Embeddings | 61.58 | 64.50 | 64.23 | 60.00 | 60.80 | 62.09 |
| Unigram + embeddings | 67.50 | 67.03 | 67.38 | 55.10 | 64.58 | 54.65 |

Table 2.3: Macro-averaged F_1 scores for subtask A and subtask B in Liebeck et al. (2017)

Afterwards, we modernized our underlying natural language processing pipeline and moved to spaCy⁷, the leading industrial-strength natural language processing pipeline for Python. It is important to use a pipeline with high accuracy because errors in the beginning of the pipeline can negatively affect subsequent pipeline steps and since errors in natural language processing usually negatively influence the classification results.

We experimented with different n-gram sizes for the character embeddings and report our results in Table 2.4. For the training of the character embeddings, we used Facebook’s fastText implementation⁸ and trained 100-dimensional character embed-

⁷<https://spacy.io/>

⁸<https://github.com/facebookresearch/fastText>

dings on the same German Wikipedia dump that we used for the word embeddings. We evaluated different numbers of training iterations $i \in \{5, 10, 20, 50, 100\}$ (with the default number of iterations being 5) and each cell in Table 2.4 displays the best result for the respective n-gram size and classifier. As we can see, there is no real difference between different n-gram sizes although $n = 4$ achieved the best results in four out of six cases. FastText uses (3,6) as default n-gram size which means that n-grams with the lengths 3 up to 6 are combined. The random forest classifier performed the best results for subtask A and the SVM achieved the highest results for subtask B.

| n-gram size | | 3 | 4 | 5 | 6 | (3,6) |
|------------------|---------|--------------|--------------|-------|--------------|-------|
| Subtask A | SVM-RBF | 58.26 | 61.47 | 60.12 | 59.74 | 58.65 |
| | RF | 58.76 | 62.26 | 60.76 | 61.60 | 58.11 |
| | KNN | 53.92 | 57.28 | 56.25 | 55.69 | 52.61 |
| Subtask B | SVM-RBF | 57.41 | 55.29 | 55.58 | 57.29 | 56.84 |
| | RF | 51.37 | 52.21 | 53.00 | 54.26 | 52.84 |
| | KNN | 48.59 | 53.21 | 52.04 | 50.06 | 51.23 |

Table 2.4: Macro-averaged F_1 scores of subtask A and B for different character embedding n-gram sizes

We now present the latest results for subtask A and B in Table 2.5 which include the features presented in Section 2.6.1. The parameters for the features and the classifiers were estimated with a tenfold cross-validation on the training set. The listed scores are macro-averaged F_1 scores for the test set. Rows marked with subscript *max* indicate that multiple models with different dimensions were tested. Each cell reports the highest value of all tested models. In this thesis, we explored more features in the evaluation than in our previous publications. Additionally, we also evaluated further feature combinations by concatenating their respective vector representations. Their results can be seen in lower half of the table.

In addition to the three classifiers we used in our previous publications, we now also evaluate SVMs with a linear kernel. In our evaluation, experiments with a linear kernel were much faster to compute for features with a high dimensionality. We did not report results for an SVM with a linear kernel for the comparison of character embeddings (Table 2.3) since the runtime for our grid search was very high with our embedding dimension of 100.

By comparing the result of unigrams with an SVM for subtask B with the results of previous publications, we notice a drop in the evaluation results on the test set with spaCy’s tokenizer. The results in Table 2.5 were achieved by a tenfold cross-validation on the training set. Upon manually investigating this performance drop, we determined that the performance can be increased from 55.98% to 61.11% by changing γ from 0.1 to 0.001 and C from 10 to 10000. The same behavior can be observed for *unigram + grammatical* in subtask B where the SVM results are 56.51% instead of 64.26% with the tuned parameters. It is interesting to see such a difference in the performance. As we mentioned earlier, we do not want that knowledge bleeds from the test set into our parameter selection which is why we list the lower scores in Table 2.5. However, it is interesting that $\gamma = 0.001$ and $C = 100000$ were not selected to be the best performing parameters in the tenfold cross-validation on the training set.

This leads to the conclusion that the dataset is sensitive to its choice of classification parameters and that the randomly selected test dataset might not be as similar to the training set as desired. We are curious about what observations we will make on future datasets.

If we compare the different classifiers in Table 2.5, we can see that the linear SVM is almost always better than the k-nearest neighbor classifier. In view of this results, we will not use k-NN in the future anymore. An SVM with an RBF kernel was still the best choice in both subtasks. If we take a look at the different rows or features, we can see that the *shape* feature always performed worse than *unigrams* from which we deduce that it is not a good feature for argument mining. By comparing the results of *LDA* trained on Wikipedia and on the whole THF corpus, we see that the performance of the THF LDA distribution is better in six out of eight cases. However, if we combine the LDA distribution with unigrams, we see almost no difference between the F_1 scores of both corpora. For the SVM with an RBF kernel, *word embeddings* and *character embeddings* have both proven to be good features since they were both able to improve the results of *unigrams* and of *unigrams + grammatical*.

In total, we performed an extensive parameter search and evaluated a multitude of features with four classifiers. In our latest implementation, an SVM with an RBF kernel achieved the best performances of 69.71% and 67.26% for subtasks A and B, respectively. At the end of this chapter, we will outline multiple ideas which might increase the overall performance.

| Feature Set | Subtask A | | | | Subtask B | | | |
|---------------------------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | SVM-RBF | SVM-lin | k-NN | RF | SVM-RBF | SVM-lin | k-NN | RF |
| Unigram ① | 65.04 | 65.15 | 62.41 | 66.69 | 55.98 | 61.26 | 43.59 | 58.24 |
| Grammatical ② | 54.76 | 56.10 | 48.93 | 58.33 | 53.49 | 56.98 | 51.84 | 51.81 |
| Shape | 57.97 | 60.80 | 55.60 | 60.44 | 51.91 | 48.51 | 47.51 | 49.22 |
| LDA Wikipedia _{max} ③ | 61.37 | 54.35 | 54.30 | 55.51 | 40.23 | 44.10 | 37.50 | 41.12 |
| LDA THF _{max} ④ | 54.41 | 52.44 | 57.38 | 60.23 | 46.21 | 44.12 | 45.39 | 47.49 |
| Character n-grams _{max} | 55.63 | 54.35 | 51.40 | 57.83 | 47.40 | 49.66 | 43.14 | 49.52 |
| Word embeddings _{max} ⑤ | 63.51 | 64.47 | 60.25 | 65.52 | 61.88 | 61.06 | 53.52 | 56.25 |
| Character embeddings _{max} ⑥ | 61.47 | 58.33 | 57.28 | 62.26 | 57.41 | 54.51 | 53.21 | 54.26 |
| ① + ② | 66.20 | 68.14 | 63.59 | 60.83 | 56.51 | 65.55 | 47.15 | 59.60 |
| ① + ③ | 68.36 | 65.43 | 63.44 | 66.21 | 65.02 | 66.28 | 50.77 | 56.89 |
| ① + ④ | 68.15 | 67.38 | 63.91 | 65.15 | 54.88 | 65.35 | 50.78 | 59.19 |
| ① + ⑤ | 67.68 | 68.25 | 65.81 | 69.36 | 57.28 | 64.75 | 47.01 | 56.40 |
| ① + ⑥ | 69.71 | 66.37 | 57.65 | 64.25 | 64.23 | 63.65 | 51.98 | 54.40 |
| ① + ② + ⑤ | 66.90 | 68.85 | 65.32 | 65.93 | 67.26 | 64.46 | 50.01 | 55.64 |
| ① + ② + ⑥ | 68.15 | 66.83 | 58.52 | 65.59 | 63.40 | 63.15 | 53.30 | 56.73 |
| ① + ② + ⑤ + ⑥ | 68.79 | 67.49 | 58.33 | 67.32 | 64.50 | 63.66 | 52.31 | 59.68 |

Table 2.5: Macro-averaged F_1 scores for further evaluations of subtask A and subtask B

2.6.3 Differentiating Between Claim Types

After identifying claims in subtask B, we can now focus on differentiating them further into pro claims and contra claims. Since subtask C is a sentiment task and a typical approach for sentiment tasks is to include sentiment lexicons as features, we additionally experimented with SentiWS (Remus et al., 2010) as a discretized distribution of sentiment scores.

The low number of annotated claims (see Table 2.1) is going to be problematic for training a classifier and even worse for the evaluation. With this low number of training instances, classifiers will most likely overfit and not generalize well. As there are only 13 instances of contra claims in the test set of subtask C, wrong predictions will have a huge influence on the classification result and features that capture contra claims well will likely achieve the best classification results. Nevertheless, we want to report results for subtask C to show complete results for all three steps in our classification hierarchy on our corpus. The results are reported in Table 2.6. Since the distribution of pro claims and contra claims is skewed, the first row represents a majority baseline of 44.8%, in which each sentence of the test set is predicted to be a pro claim. For a classifier to be feasible, it must at least beat this baseline. As we can see, the shape feature twice achieved worse results than the baseline. This also happened once for character n-grams and once for LDA trained on Wikipedia.

| Feature Set | Subtask C | | | |
|--|--------------|--------------|--------------|--------------|
| | SVM-RBF | SVM-lin | k-NN | RF |
| Majority baseline | 44.80 | 44.80 | 44.80 | 44.80 |
| Unigram | 62.15 | 63.49 | 63.10 | 74.78 |
| Bigram | 51.38 | 49.68 | 44.80 | 53.22 |
| Grammatical | 57.07 | 54.20 | 58.53 | 53.17 |
| Shape | <i>43.90</i> | 58.93 | <i>41.46</i> | 64.53 |
| LDA Wikipedia _{max} | 66.15 | 60.10 | 55.28 | <i>44.52</i> |
| LDA THF _{max} | 63.38 | 59.68 | 67.55 | 59.81 |
| Character n-grams | 51.02 | 61.28 | <i>44.49</i> | 59.81 |
| Word embeddings _{max} | 63.36 | 64.64 | 56.62 | 59.65 |
| SentiWS distribution | 44.80 | 44.80 | 44.80 | 44.80 |
| Character embeddings _{max} | 74.44 | 70.95 | 69.93 | 68.25 |
| Unigram + Grammatical | 67.55 | 60.87 | 63.49 | 55.19 |
| Unigram + LDA Wikipedia _{max} | 71.30 | 66.09 | 69.11 | 60.87 |
| Unigram + LDA THF _{max} | 71.30 | 72.96 | 69.11 | 66.83 |
| Unigram + word embedding _{max} | 72.96 | 66.09 | 60.33 | 57.41 |
| Unigram + character embedding _{max} | 71.57 | 71.62 | 66.15 | 62.09 |

Table 2.6: Macro-averaged F_1 scores for subtask C

Our first attempt at capturing sentiment with SentiWS failed because all classifiers only learned to predict the majority class. At the end of this chapter, we will outline our thoughts about other sentiment features in our future work.

For subtask C, the best result of 74.78% was achieved with the random forest classifier and unigrams. For RF it is interesting to see, that all tested combinations

with other features only worsened the results. In a direct comparison, character n-grams were always worse than unigrams. Character embeddings achieved the best results for the SVM and k-NN and they produced high results for all classifiers.

In summary, the results of subtask C are not that meaningful since the dataset is simply too small to allow for a reasonable machine learning task. This is also the reason why we do not show the results of all the features we evaluated for subtask A and B. We hope to revisit subtask C with a larger dataset in the future.

2.7 Using Deep Learning for Argument Mining

We will now investigate whether we can use deep learning to achieve better classification results than our classical machine learning approach with feature engineering.

2.7.1 Introduction

Deep learning techniques have shown great promise in the past. They are most notable in the field of computer vision. In the computer linguistic community around the *Association for Computational Linguistics* (ACL) conferences, deep learning approaches in the form of variations of recurrent neural networks (RNN) are becoming increasingly popular and are probably going to be established as a new baseline technique in the near future.

By now, deep learning has surpassed human performance in multiple tasks: (i) image classification, (ii) face recognition, and (iii) speech recognition.

Deep learning performs very good in the *ImageNet Large Scale Visual Recognition Challenge* (ILSVRC). The challenge consists of a test set comprising 100,000 images where each image is labeled with one label out of 1000 classes. Participating teams have to predict five labels that they are most confident of per image. The results are evaluated as top-5 error score (the number of images where none of the five labels equal the gold label). The human performance of 5.1% was broken by He et al. (2015) with a top-5 error rate of 4.94%. Further deep learning approaches have lowered the error score to 3.57% (He et al., 2016), 3.08% (Szegedy et al., 2017) and even down to 2.991% (ILSVRC2016, 2017). For face recognition, the *Labeled Faces in the Wild* (LFW) (Huang et al., 2007) dataset comprises pairs of images depicting faces. The classification task for the dataset is to identify whether the persons in two presented images are the same. According to Kumar et al. (2009), the human performance is 97.53% accuracy. Deep learning approaches, such as Schroff et al. (2015) with 99.63%, surpass this performance. For speech recognition, a combination of convolutional neural networks (CNNs) (LeCun et al., 1989; LeCun et al., 1998) and long short-term memories (LSTMs) (Hochreiter and Schmidhuber, 1997) is able to achieve slightly lower error rates than humans (Xiong et al., 2016).

For natural language processing, deep learning approaches achieve good or even state-of-the-art results in a variety of tasks and domains, including parsing (Fried et al., 2017), emotion detection (Abdul-Mageed and Ungar, 2017), topic classification and news categorization (Zhang et al., 2015), part-of-speech tagging (P. Wang et al., 2015), and sentiment analysis on Amazon reviews (Conneau et al., 2017).

The strong performance of deep learning approaches in such a variety of tasks motivates us to evaluate whether deep learning approaches can achieve better results than our machine learning approach with a classical feature engineering. However, the number of training instances is crucial since an unduly low number of training instances is challenging and will not allow for a correct training or a training that can generalize from the training instances. In this section, we want to experiment if the low number of training instances (see Table 2.1) is high enough to train neural networks and able to improve on previous results.

We also want to highlight the work of Y. Goldberg (2016), who explains what multiple deep learning components mean for NLP and discusses a multitude of deep

learning design decisions and their effects on NLP tasks.

In argument mining, deep learning methods have been becoming more and more popular since 2016: CNNs were used by Stab and Gurevych (2017) and Aker et al. (2017). Moreover, LSTMs were also utilized (Eger et al., 2017; Hou and Jochim, 2017) in argument mining systems. An LSTM variant, called bidirectional LSTMs, was used often (Habernal and Gurevych, 2016; Niculae et al., 2017; Ajjour et al., 2017). Argument mining architectures also included deep average networks (DAN) (Iyyer et al., 2015) (e.g., (Fierro et al., 2017)) and employed other types of RNNs, e.g., bidirectional RNNs (Schuster and Paliwal, 1997) in Koreeda et al. (2016) or Grid-LSTMs (Kalchbrenner et al., 2015) in Ouchi et al. (2017).

2.7.2 Architectures

We now benchmark deep learning architectures that are common for NLP. For our implementation, we rely on Keras (Chollet, 2015) as a deep learning library with Theano (Theano Development Team, 2016) as the backend.

We start to describe the different architectures by describing the preprocessing that all architectures have in common. First, each sentence is treated as a sequence of tokens with the same tokenization from spaCy, which we also used in the latest evaluations with the classical feature engineering in Section 2.6.2.3. A frequency map of all words in the corpus is constructed, sorted in descending order, and each word is from now on represented by its index number in the frequency map. We feed these index numbers, which are each mapped to a corresponding row in an embedding matrix, as inputs into our neural networks. For all but the first architecture, this embedding matrix is initialized with pre-trained word embeddings from a 300-dimensional word2vec embedding that was trained on a German Wikipedia dump. As our architectures consist of recurrent neural networks, the tokens are pre-padded or trimmed to a fixed length. The embedding matrix contains the zero vector as a special row for the padding token. Words that do not have a word embedding in our trained model are assigned a randomly generated out-of-vocabulary vector of the same length.

As the last layer of each neural network, we condense our output to two or three neurons, depending on the subtask. A prediction for an input sentence is obtained by using `argmax` on the output layer to get the predicted class label. What distinguishes our benchmarked neural networks from each other is how they deal with the sequence of embeddings. The joint architecture of the models is visualized in Figure 2.5. In total, we benchmark six different deep learning architectures:

1. **Embeddings with random weights + LSTM:**

As a baseline, we start with an embedding layer whose weights are randomly initialized from a uniform distribution. The sequence of embedded words is then fed into a vanilla LSTM.

2. **Pre-trained embeddings + LSTM:**

Instead of relying on a random initialization, we now initialize the embedding matrix with the weights from the word2vec model. The sequence of embedded words is again fed into a vanilla LSTM.

3. **Pre-trained embeddings + two stacked LSTMs:**

Additionally, we evaluate the *stacking* (Graves et al., 2013) of two LSTMs on

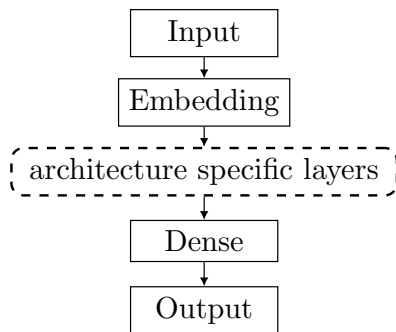


Figure 2.5: Joint architecture of the benchmarked neural networks

top of each other since stacked LSTMs have repeatedly been reported to be able to improve the classification results for multiple tasks. In our previous work on sentence ordering (Modaresi et al., 2016b), we were able to achieve excellent classification results with stacked LSTMs.

We would like to quote from Goldberg’s primer about stacked recurrent neural networks: “*While it is not theoretically clear what is the additional power gained by the deeper architecture, it was observed empirically that deep RNNs work better than shallower ones on some tasks*” (Y. Goldberg, 2016).

4. **Pre-trained embeddings + two stacked bidirectional LSTMs:**

We replace our unidirectional LSTMs with bidirectional LSTMs (BI-LSTMs) (Schuster and Paliwal, 1997; Graves and Schmidhuber, 2005). For each timestep, a BI-LSTM has access to past and future inputs in contrast to unidirectional LSTMs which can only access past inputs. According to Graves et al. (2005), bidirectional LSTMs can outperform unidirectional LSTMs. We evaluate whether this effect will also occur for our corpus.

5. **Pre-trained embeddings + CNN:**

Convolutional neural networks have become an integral part of neural networks for image classification problems but they can also be used for text classification (Kalchbrenner et al., 2014). Y. Goldberg (2016) describes CNNs as “*useful for classification tasks in which we expect to find strong local clues regarding class membership, but these clues can appear in different places in the input.*” Since we noticed strong lexical clues during the annotation of the corpus for our class *claim*, we want to benchmark how good a combination of embeddings and a CNN with max pooling can predict argument components.

6. **Pre-trained embeddings + CNN + LSTM:**

The last architecture that we are testing is a combination of a CNN with an LSTM where the output of the CNN serves as the input for the LSTM. Such an architecture has been successfully used for the sentiment classification of short texts (X. Wang et al., 2016) where the combination of both models achieved better results than a CNN or an LSTM alone.

In our evaluation, we experimented with a small grid search for some of the hyperparameters, such as the number of training epochs, LSTM layer dimensionalities, CNN output dimensionality and kernel sizes, and various levels of dropout to prevent overfitting. For now, we limited our evaluations to 300-dimensional word2vec embeddings.

2.7.3 Results

The macro-averaged F_1 scores for subtasks A and B are shown in Table 2.7 for each benchmarked architecture. The results were predicted by models after their last training epoch and each cell depicts the maximum score of all tested hyperparameter configurations. Our assumption that the number of training data is insufficient to train the neural networks for both subtasks turned out to be unfounded since all deep learning architectures were able to improve with each training epoch. We omitted subtask C from the deep learning experiments due to its small size.

| Architecture | Subtask A | Subtask B |
|---|--------------|--------------|
| Embeddings with random weights + LSTM | 64.95 | 64.68 |
| Pre-trained embeddings + LSTM | 68.51 | 63.77 |
| Pre-trained embeddings + two stacked LSTMs | 67.73 | 64.20 |
| Pre-trained embeddings + two stacked BI-LSTMs | 67.61 | 66.18 |
| Pre-trained embeddings + CNN | 67.54 | 68.59 |
| Pre-trained embeddings + CNN + LSTM | 66.03 | 65.91 |

Table 2.7: Macro-averaged F_1 scores for the benchmarked deep learning architectures

The best result for subtask A of 68.51% was achieved with pre-trained embeddings and a unidirectional LSTM. We can also observe for subtask A that pre-trained embeddings were more effective in detecting argumentative content than randomized weights. In a comparison with the results of the classical feature engineering in Table 2.5, we can see that the best result of 69.71% was not outperformed by our initial deep learning evaluations.

For subtask B, pre-trained embeddings with a CNN yielded the best score of 68.59% which is a slight improvement compared to the 67.20% of the best result from classical feature engineering. We can also observe that two stacked BI-LSTMs predicted better results than two stacked unidirectional LSTMs for subtask B but not for subtask A.

Overall, it was an interesting and worthwhile comparison between classical feature engineering and deep learning models. We are not able to declare a “winner” since each approach was each able to yield better predictions in one subtask. We are eager to see how both fundamentally different approaches perform on other, larger online participation datasets.

2.8 Future Work

We would like to conclude this chapter by outlining further research ideas:

- **Classical machine learning:**

We would like to evaluate more feature combinations. So far, we performed an extensive grid search for the parameter estimation of the classifiers and the features. We limited the number of feature combinations since the parameter estimation was time-consuming and took multiple weeks although we already parallelized the search on several hundred CPU cores.

In our grid search, we included a parameter for normalizing our bag-of-words models in order to reduce the influence of very long sentences. We experimented with tf-idf weighting but our results worsened. We would like to benchmark the effects of using tf-idf as weights for word embeddings in the future.

So far, we have conducted a couple of experiments to reduce the dimensionality of our bag-of-words models by filtering out words that are too frequent or too rare. In addition, we have also experimented with feature selection techniques. Unfortunately, both attempts to reduce the dimensionality had slightly negative effects on the quality of the classification results. In the future, we would like to tackle this point more extensively.

In our evaluation, we only used word embeddings from word2vec. We will also benchmark word embeddings from GloVe (Pennington et al., 2014).

Additionally, we want to evaluate whether the combination of multiple classifiers into an ensemble can improve the classification results.

For subtask C, we would like to evaluate more sentiment lexicons, such as *SB10k* (Cieliebak et al., 2017), *EmoLex* (Mohammad and Turney, 2013), and *Potsdam Twitter Sentiment Corpus* (Sidarenka, 2016). Additionally, we plan to use distant supervision to create our own sentiment lexicons.

- **Dataset:**

All the experiments in this chapter have been focused on a moderately sized dataset. In order to increase the number of training instances, we could replace words with their synonyms to artificially create new instances with the same label. Another viable approach includes crowdsourcing where the workers are asked to paraphrase the sentences in our corpus to obtain more labeled data.

As we will describe later in our conclusion in Chapter 5, we also plan to carry out experiments on other online participation datasets.

We did not annotate relations between argument components in our corpus due to the difficulty of correctly annotating each relation in our multi-document discourse between users. However, we would at least like to revisit the annotation and prediction of intra-document argument relations in the future.

- **Sequence tagging:**

A limitation of our approaches is the classification granularity, which currently is on a sentence level. However, a sentence can contain multiple argument components (e.g., a claim followed by a premise) which is the case for 3.6% of the

sentences in our corpus (Liebeck et al., 2016a). It is also possible that a single argument component might span several consecutive sentences. Therefore, we would like to use sequence tagging to address the issue. With sequence tagging, we can group consecutive tokens with the same predicted label regardless of sentence boundaries.

We will build on previous work on sequence tagging in argument mining (Goudas et al., 2014; Habernal and Gurevych, 2017) and will evaluate different coding schemes since the choice of the coding scheme can have an influence on the classification results for other NLP tasks (Ratinov and Roth, 2009).

- **Deep learning:**

There are also several ways to explore deep learning in more depth using our corpus. First of all, we could experiment with different ways to reduce the input vocabulary, e.g., by only using words with specific POS tags or by filtering too infrequent words or stop words. Then, we would like to do a more systematic and exhaustive grid search for hyperparameters, such as batch size, padding length, layer sizes, and dropout values. Additionally, we will benchmark more embeddings, such as fastText or GloVe, and different embedding sizes. Furthermore, we will explore which effects occur when combining multiple embeddings as the input of our models. We would also like to evaluate more deep learning architectures, for instance, LSTMs with attention (Bahdanau et al., 2014) and gated recurrent units (GRU) (Cho et al., 2014).

3

TOPIC EXTRACTION

In this chapter, we focus on one of the most important tasks for an automated text analysis system of online participation processes, more specifically on the automatic extraction of topics or subjects of debate. This is especially important for popular processes which attract a lot of users and thousands of text contributions. Since it is neither easy nor feasible to quickly achieve a broad overview of the discussed topics, it is inevitable for such huge text collections to use an automated approach in order to extract the discussed topics and to present them online to all users of the platform, including moderators, politicians, and citizens. Although it is possible to quickly read most of the upvoted ideas in an online participation process, they cannot be considered to be representative of the whole process.

We address this problem by focusing on multiple subproblems. Our work begins with the reduction of the vocabulary size by creating a new resource-based lemmatizer, considers textual similarity and text obfuscations, includes multiple results of topic extraction approaches, and concludes with an outline of using topic labeling as the next future step in our extraction pipeline.

3.1 Using Wiktionary to Reduce the Vocabulary Size with a New German Lemmatizer

Classical machine learning approaches often include a Bag-of-Words (BoW) approach as a feature that is defined by a vocabulary, a set of unique words in the training set. As a result, the dimensionality of the resulting feature vector is linearly dependent on the vocabulary size (ignoring that the vocabulary size is sometimes set to a fixed number). Since German is a morphologically rich language, words or certain part-of-speech tags can be conjugated or declined, respectively. For instance words can appear in singular or plural forms, e.g., *Schwimmbad* (swimming pool) and *Schwimmbäder* (swimming pools). In terms of classification problems, both forms almost convey the same semantics and can be regarded as the same word. The process of replacing a word with its lemma (e.g., *Schwimmbäder* \mapsto *Schwimmbad* and *gespielt* (played) \mapsto *spielen*

(to play)) is called lemmatization. Using lemmatization reduces the vocabulary size of a BoW model and can also be useful for methods that determine the most frequent words in a text corpus.

We would like to point out that lemmatization is dependent on the context and the part-of-speech since an inflected form can map to more than one lemma. For example, *Kohle* maps to *Kohle* (coal) and *Kohl* (cabbage).

Most of the lemmatization methods rely on statistical or rule-based approaches. Although these methods are able to predict a lemma for a previously unknown word, it is possible that these methods make errors by predicting a wrong lemma. We formulate the hypothesis that such approaches can be improved by first looking up an inflected form in a dictionary (containing mappings to their lemmas) and only resorting to the aforementioned approaches if an inflected form is not present in the dictionary. This should prove to be true as long as the dictionary entries are correct.

In order to validate this hypothesis, we created IWNLP¹ (Liebeck and Conrad, 2015), an open-source parser for the German version of Wiktionary that provides a mapping from inflected forms to their lemmas. Wiktionary is a MediaWiki-based platform comprising a large dictionary in which each entry can also include inflections. Information on Wiktionary is entered by users into so-called templates, which are written in the MediaWiki markup language. The template for nouns is rather simple since the template uses a fixed number of key-value parameters which are then rendered as an HTML table. Figure 3.1a shows the syntax for the word *Haus* (house) and Figure 3.1b illustrates the browser-rendered template. Templates for conjugated verbs and declined adjectives, on the other hand, are far more complex since they only require principal parts which are combined with suffixes. Figure 3.1c shows an example for *stehen* (to play) and an excerpt of the rendered HTML output is given in Figure 3.1d. Instead of stressing the official Wiktionary server by crawling every entry, we decided to parse the monthly Wiktionary XML dump (containing the raw input of each page) and to reimplement specific parts of MediaWiki’s rendering engine in order to obtain pairs of (inflected form, lemma).

In Liebeck and Conrad (2015), we combined IWNLP with two existing software for lemmatization, namely Mate Tools (Björkelund et al., 2010) and TreeTagger (Schmid, 1994). We were able to increase their results for nouns, verbs, and adjectives across three corpora and verified our hypothesis. Compared with Morphy (Lezius et al., 1998), a lemmatizer that also provides pairs of (inflected form, lemma), IWNLP is also able to achieve better results in all cases.

We would like to emphasize that IWNLP can also be used as a standalone lemmatizer. In Liebeck and Conrad (2015), we evaluated IWNLP as a standalone lemmatizer on three German corpora. The evaluation results are listed in Table 3.1 as accuracy scores for all words that were correctly lemmatized. Words for which we predict more than one lemma are treated as a wrong lemmatization. The first column reports the corrected benchmark for the initial results from Liebeck and Conrad (2015). The following columns show the lemmatization results for subsequent XML dumps with a delta calculated with respect to the corrected paper version. In Liebeck and Conrad (2015), we predicted that the results will improve as Wiktionary continues to grow. This proved to be true. In particular, this can be observed by the improvement between the dumps *20151123* and *20151226* which is only due to the work of the Wiktionary

¹<http://www.iwnlp.com>


```

{{Deutsch Substantiv Übersicht
|Genus=n
|Nominativ Singular=Haus
|Nominativ Plural=Häuser
|Genitiv Singular=Hauses
|Genitiv Plural=Häuser
|Dativ Singular=Haus
|Dativ Singular*=Häuser
|Dativ Plural=Häusern
|Akkusativ Singular=Haus
|Akkusativ Plural=Häuser
}}

```

(a) Template code for *Haus* (house)

| | Singular | Plural |
|-----------|-----------------------|-------------|
| Nominativ | das Haus | die Häuser |
| Genitiv | des Hauses | der Häuser |
| Dativ | dem Haus
dem Hause | den Häusern |
| Akkusativ | das Haus | die Häuser |

(b) Rendered view of *Haus*

```

{{Deutsch Verb unregelmäßig
|2=steh
|3=stand
|4=ständig
|5=gestanden
|7=em
|9=stünd
|vp=sie_werden
|zp=sie_sind
|gerund=ja
}}

```

(c) Template code for *stehen* (to stand)

| Präsens | | |
|--------------------|-----------------|-----------------|
| Person | Aktiv | |
| | Indikativ | Konjunktiv I |
| 1. Person Singular | ich stehe | ich stehe |
| 2. Person Singular | du stehst | du stehest |
| 3. Person Singular | er/sie/es steht | er/sie/es stehe |
| 1. Person Plural | wir stehen | wir stehen |
| 2. Person Plural | ihr steht | ihr stehet |
| 3. Person Plural | sie stehen | sie stehen |

(d) Excerpt of the rendered view of *stehen*

Figure 3.1: Examples of two Wiktionary entries and their rendered HTML outputs

community. In the last two years, the number of covered words increased but the number of lexically ambiguous words, i.e., words for which more than one lemma exists in our mapping, has also risen. This resulted in a slight drop for the adjective results because some adjectives have Wiktionary entries for their base form, their comparative, and their superlative.² Our evaluation results would be higher if our evaluation metric would also measure cases where one of the proposed lemmas of a given word would match the gold lemma instead of an exact match. However, we decided to treat such cases as wrong lookups in order to not prettify our results since techniques,

²We construct our lemmatization mapping in the following way: For each Wiktionary entry l that contains one of the templates we support, we add mappings for $inflected\ form \mapsto l$ for all inflected forms that can be extracted from l . Additionally, we add $l \mapsto l$ to include words that are already present as their lemmas. For some adjectives in the last dump, the superlative s for an adjective a is also present as a Wiktionary entry which makes the lookup of s ambiguous due to $s \mapsto \{a, s\}$. We plan to filter out such comparatives and superlatives in order to reduce the number of ambiguous adjectives in the future.

such as word sense disambiguation, would be required to choose the correct lemma in a given context and our goal was to convey a realistic impression of an easy-to-use dictionary-based lemmatizer for German.

| Corpus | POS | 20150407
corrected | 20151123 | 20151226 | 20161020 | 20170501 |
|----------|------------|-----------------------|-----------------|------------------|------------------|------------------|
| Tiger | Nouns | 0.738 | 0.748
+1.0% | 0.749
+1.1% | 0.758
+2.0% | 0.763
+2.5% |
| | Verbs | 0.837 | 0.897
+6.0 % | 0.897
+6.0 % | 0.917
+8.0 % | 0.921
+8.4 % |
| | Adjectives | 0.636 | 0.732
+9.6 % | 0.751
+11.5 % | 0.778
+14.2 % | 0.775
+13.9 % |
| TüBa-D/Z | Nouns | 0.721 | 0.747
+2.6 % | 0.747
+2.6 % | 0.756
+3.5 % | 0.761
+4.0 % |
| | Verbs | 0.81 | 0.859
+4.9 % | 0.86
+5.0 % | 0.876
+6.6 % | 0.879
+6.9 % |
| | Adjectives | 0.57 | 0.662
+9.2 % | 0.684
+11.4 % | 0.71
+14.0 % | 0.713
+14.3 % |
| HDT | Nouns | 0.607 | 0.627
+2.0 % | 0.627
+2.0 % | 0.63
+2.3 % | 0.631
+2.4 % |
| | Verbs | 0.864 | 0.924
+6.0 % | 0.925
+6.1 % | 0.939
+7.5 % | 0.943
+7.9% |
| | Adjectives | 0.627 | 0.708
+8.1 % | 0.715
+8.8 % | 0.72
+9.3 % | 0.71
+8.3 % |

Table 3.1: Percentages of correctly lemmatized words with IWNLP as a standalone lemmatizer

IWNLP is available as a standalone Python implementation³. Currently, IWNLP is being implemented as an extension to spaCy.

3.2 Textual Similarity

We now focus on measuring the similarity between text passages. This task, also known as semantic textual similarity (STS), is helpful for a variety of NLP tasks, such as information retrieval and text summarization. STS is usually measured on an ordinal scale ranging from totally unrelated over somewhat related to conveying the same meaning. Although detecting textual similarity is easy for humans, it is not easy to determine it automatically since contextual knowledge has a significant influence on deciding how closely related two text passages are.

Although resources for determining synonyms and antonyms exist, such as WordNet (Miller, 1995) for English and GermaNet (Hamp and Feldweg, 1997) for German, they only work for single words. If we want to measure the STS between longer text passages, techniques that work on a broader granularity level, e.g., on sentences, are required.

³IWNLP-py, <https://github.com/Liebeck/IWNLP-py>

For the analysis of online participation processes, measuring STS is useful for at least two subtasks: (i) while extracting discussion topics (see Section 3.3), terms that are semantically close together can be grouped and treated as the same topic; and (ii) for grouping argument components. Let us assume we applied the techniques from Chapter 2 to an online participation process and we extracted multiple premises. If we measure the pairwise STS of these premises, we might be able to group identical reasons together even if they are formulated differently.

In this section, we start by addressing the semantic similarity of two sentences and then move over to the highly related field of detecting paraphrases in obfuscated texts.

3.2.1 Textual Similarity of Two Sentences

In order to engage into measuring STS, we decided to participate (Liebeck et al., 2016d) in the English subtask of the *SemEval-2016 Task 1: Semantic Textual Similarity* challenge (Agirre et al., 2016). The challenge focused on the development of systems that can predict the semantic similarity of two given input sentences in the continuous interval $[0, 5]$ where 0 is defined as complete dissimilarity and 5 represents a complete semantic equivalence.

The task organizers provided a test set comprising 1186 sentences pairs. The task participants had to predict the STS score for all of these test instances. Their gold standard was created via crowdsourcing and was only known to the task organizers, so that systems could not be overfitted toward the test set. Each team was allowed to submit the predictions of up to three systems, which allowed the participants to try different parameters for their models or even different approaches altogether. In order to report a final ranking for all the participant’s systems, the task organizers calculated the Pearson correlation between the predicted STS scores and the gold STS scores for each team. In 2016, 43 teams participated with 119 submissions in total.

In our approach (Liebeck et al., 2016d), we pursued three different approaches, one unsupervised and two supervised, and submitted one set of predictions per approach. In the final evaluation ranking of the challenge, our approaches ranked 33rd, 66th, and 85th place, respectively. In this section, we only outline our best working approach, named *Overlap*, which is unsupervised. We also evaluated our three approaches on the available test dataset of 2015, where we achieved much better results and our best working supervised approach even outperformed the *Overlap* method by quite a bit. However, in 2016, our unsupervised approached surprisingly outperformed both supervised approaches. Since the task organizers did not provide domain-specific training data for 2016 and since most of the text domains in the test data were not present in the available datasets from the previous years, it becomes apparent that our unsupervised approach was able to generalize better.

We will now outline how our unsupervised *Overlap* approach works. We started by defining a similarity function $\text{sim}(t_1, t_2)$ between two tokens t_1 and t_2 as

$$\text{sim}(t_1, t_2) := \begin{cases} 1 & \text{if } t_1.\text{lemma} == t_2.\text{lemma} \\ 1 & \text{if } t_1 \text{ and } t_2 \text{ have the same most common synset} \\ 0.5 & \text{if } t_1 \text{ and } t_2 \text{ share any synset} \\ d(t_1, t_2) & \text{if } t_1 \text{ and } t_2 \text{ have word embeddings} \\ \text{default} & \text{otherwise} \end{cases} \quad (3.1)$$

where $d(t_1, t_2)$ denotes the cosine similarity between the word2vec embeddings of t_1 and t_2 , synonyms are looked up in synsets from Wordnet (Miller, 1995), and the *default* value can be 0 or manually tuned for a given dataset with gold labels. Given two sentences s_1 and s_2 , we can then define their similarity $\text{ssim}(s_1, s_2)$ by calculating the similarity between the tokens of s_1 and s_2 :

$$\text{ssim}(s_1, s_2) := 5 \cdot \left(\frac{\sum_{t_1 \in s_1} \max_{t_2 \in s_2} \text{sim}(t_1, t_2)}{2 \cdot |s_1|} + \frac{\sum_{t_2 \in s_2} \max_{t_1 \in s_1} \text{sim}(t_2, t_1)}{2 \cdot |s_2|} \right) \quad (3.2)$$

Table 3.2 shows two example pairs from the 2016 test dataset. The third column lists the gold standard STS from the task organizers and the last column shows the results calculated by our *Overlap* method. As we can see, the prediction for the first example is almost perfect, whereas the prediction for the second example is not accurate although both sentences share a high number of words after stop word filtering.

| Sentence 1 | Sentence 2 | STS | Overlap |
|---|--|-----|---------|
| Unfortunately the answer to your question is we simply do not know. | Sorry, I don't know the answer to your question. | 4 | 4.05800 |
| Unfortunately the answer to your question is we simply do not know. | My answer to your question is "Probably Not". | 1 | 3.70982 |

Table 3.2: Example results of the Overlap method compared to the gold standard (Liebeck et al., 2016d)

The best-ranking systems in the last couple of years usually utilize deep learning techniques. In our future work, we will also evaluate the use of recurrent neural networks for calculating STS.

So far, we have only evaluated our approaches for English. Although there are three small German corpora of manually annotated word pairs (Gurevych, 2005; Zesch and Gurevych, 2006), there is currently no German dataset for textual similarity that is comparable to the SemEval textual similarity monolingual task. Unfortunately, we are, therefore, currently unable to evaluate our approaches for the semantic textual similarity of two German sentences.

A concept closely related to textual semantic similarity is linguistic coherence. In Modaresi et al. (2016b), we analyzed German and English news articles and experimented with an approach to order the sentences of news articles. Given two sentences

s_1 and s_2 , we addressed the binary classification task of whether s_1 directly precedes s_2 or vice versa. Additionally, we experimented with a third label, indicating whether we cannot determine the coherence due to missing context. Using two stacked layers of LSTMs, we were able to achieve F_1 scores of over 94% in both classification tasks for both languages. As a baseline, we also predicted the linguistic coherence with an SVM using a BoW model. In a direct comparison, our LSTM approach greatly outperformed the baseline although we have to admit that SVMs are not particularly suited to dealing with sequential data.

3.2.2 Textual Similarity in Obfuscated Texts

In the context of online participation processes, especially for processes where citizens can register anonymously and, therefore, can create multiple accounts, automatically determining textual similarity is also significant. With multiple user accounts, a citizen could strengthen his point of view by simply posting the same content multiple times, although slightly obfuscated in order to make these posts look like they were created by different users. Interest groups that organize their members to participate in an online participation process with a predefined list of arguments make up another use case for automated obfuscation detection.

Two online participation projects come into mind for these use cases: (i) *Online-Konsultation zur Leitentscheidung Braunkohle*⁴ and (ii) *Beteiligungsportal Baden-Württemberg: Jagd- und Wildtiermanagementgesetz*⁵:

In 2015, the state government of North Rhine-Westphalia presented a new draft about the future of lignite mining and opened it up for discussion. Contrary to previous planning, the state government decided to reduce the planned area for surface mining. As a result, the resettlement of three villages was no longer necessary. Many citizens who would be affected by the changes participated in the online discussion and frequently argued for adherence to the previous draft from 1991, which included the resettlement and financial compensation that would no longer be necessary. In the discussion, one group of users stood out, as they frequently advanced arguments in favor of the old plans and shared the same surname. This could be a family with several family members participating on the platform or a single person posting the same content via multiple user accounts.

In the second example, the state government of Baden-Württemberg presented pending changes to hunting laws in 2014 and asked interested citizens to participate in a discussion on their online participation platform regarding the future of hunting in Baden-Württemberg. One hunting association, in particular, felt that it had been negatively impacted and, therefore, created an argumentation aid (Landesjagdverband Baden-Württemberg e.V., 2014), a list of predefined arguments for use in the discussion, and mobilized its members to join the discussion and to protest against the changes to the law. The discussion gave the impression that the members of the hunting association each used a couple of arguments from their argumentation aid. In doing so, they did not copy and paste the arguments, but rather changed the wording while retaining the same content of the statement.

We would like to point out that, whatever the differences may be in terms of the

⁴<https://www.leitentscheidung-braunkohle.nrw>

⁵Unfortunately, the discussion is not publically available anymore.

motivation and the execution of user participation between a malicious citizen and an interest group organizing themselves to engage in a political discussion, there is at least the similarity that rephrased texts are written in both use cases.

In order to study the task of automatically identifying intentionally rephrased texts more intensively, we participated in the *obfuscation evaluation task* in PAN 2016⁶. The aim of the shared task was to evaluate three different obfuscation systems from the same workshop's *author masking challenge* (Potthast et al., 2016). The goal of the author masking challenge was to develop a system that takes a text document as input and paraphrases it to make the resulting *obfuscated document* appear to be written by another author. The teams were provided with so-called *problems*, groups of multiple documents from a single author where one document per group is marked as *original*. The development of such an automated approach is challenging and the three participating teams in the 2016 challenge submitted results from rather simple systems.

But in the context of online participation, a citizen does not need to use an automated system since he can post his opinion and then simply post several paraphrased or rather obfuscated versions (with his intent being to conceal that he is the author of all of these posts). In the case of the *Jagd- und Wildtiermanagementgesetz*, we can consider the argumentation aid to be the original text and the reformulations to be the obfuscated texts and treat the identification whether a user is reusing arguments from the argumentation aid as an obfuscation detection task. Both cases motivated us to investigate whether we could automatically detect text obfuscations.

For the obfuscation evaluation task, the organizers provided three categories in which the task participants were asked to come up with their own evaluation metrics: (i) safety (whether a forensic analysis software can be deceived in determining a document's author), (ii) soundness (whether the obfuscated texts are textually entailed with their originals), and (iii) sensibleness (whether the obfuscated documents are linguistically understandable and do not stand out due to bad language). These dimensions were to be evaluated independently from each other although certain dependencies exist, for instance linguistically incomprehensible obfuscations can lead to a high semantical difference.

In our approach (Liebeck et al., 2016c), we decided to evaluate each dimension with one metric:

- safety: In order to determine how good an obfuscation software can deceive a forensic analysis software, we used the forensic software GLAD (Hürlimann et al., 2015), one of the top-ranked systems at PAN 2015. GLAD uses a support vector machine for a binary classification with multiple features: n-grams, visual features (punctuation, line endings, upper and lower case ratio, text sizes), compression feature, and distributions of part-of-speech tags and dependencies. We trained GLAD on training data from the previous years, namely PAN 2013, PAN 2014, and PAN 2015. First, we calculated a baseline value by running GLAD for each of the problems, measuring if the original document of each problem is being identified as being written by the same author. Afterwards, we used GLAD to determine whether the obfuscated documents (each of which was created by the task participants based on the original document in each group)

⁶<http://pan.webis.de/clef16/pan16-web/author-obfuscation.html>

can still be identified as being written by the same author as the respective original document.

As evaluation metric for the dimension safety, we calculated the $c@1$ scores (Peñas and Rodrigo, 2011) for the obfuscations of each team and compared them with of the $c@1$ score of the baseline run. The $c@1$ score is defined as follows

$$c@1 = \frac{1}{n} \cdot \left(n_c + \frac{n_u n_c}{n} \right) \quad (3.3)$$

where n is the number of problems, n_c is the number of correct answers, and n_u is the number of unanswered problems. In our case, a correct answer indicates that the forensic software can determine that the author of the obfuscated document is still the same author as the author from the other documents in the same problem. Since the motivation of creating obfuscations is to conceal the identity of an author, a low $c@1$ score is better than a higher score and represents how good the obfuscations are. For the obfuscation evaluation task, the score should also be lower than the baseline score because, otherwise, it would mean that an obfuscation system increases the chance of detecting a document’s author instead of concealing the author.

- soundness: We decided to interpret the soundness dimension as a measure of semantic similarity since obfuscations are created by paraphrasing a document while retaining the meaning of the text content. Therefore, we decided to use the *Overlap method* mentioned in Section 3.2.1. For each pair (original, obfuscation), we calculated a semantic similarity score in $[0, 5]$ and reported the average as the metric for the obfuscation of each teams. This allowed us to use an automated approach with external training data from the STS tasks and to evaluate all obfuscations instead of only a subset.
- sensibleness: The sensibleness dimension measures how linguistically understandable obfuscations are. It is important to evaluate this dimension since obfuscations might fool a forensic analysis software but under a severe loss of language quality. After looking at the obfuscations of the three participating teams, we decided to manually evaluate the language quality with a score $s \in \{0, 1, 2\}$. Table 3.3 defines our three labels and shows an example for each label. We randomly drew 60 obfuscations and manually annotated each with three annotators. As the final metric in the sensibleness dimension, we averaged over all annotations per team. We observed a high agreement on the label *incomprehensible* but it turned out to be more difficult to achieve a good agreement on whether an obfuscation is partially or fully comprehensible due to the subjective nature of text understanding.

We would like to point out that there are multiple ways of dealing with reformulations and obfuscations once they are detected. This is because it is difficult to decide whether two people use the same argument with different wordings or one person is posting from multiple accounts. In the context of online participation processes, it is difficult to train automated systems to decide whether two text posts are written by the same user since the amount of training data per user is usually very small. Apart from technical problems and solutions to authorship detection, we think that it is important to inform the organizers of an online participation process if we automatically

| Score | Name | Definition |
|-------|---------------------------------|--|
| 2 | <i>comprehensible</i> | The paraphrase can be understood immediately.
<u>Example</u> : “ <i>These things are deeply rooted in the Swedish people.</i> ” |
| 1 | <i>partially comprehensible</i> | The paraphrase can be understood with some restrictions. It can contain smaller errors or some smaller parts that are incomprehensible.
<u>Example</u> : “ <i>they him. But ignored</i> ” |
| 0 | <i>incomprehensible</i> | The language quality of the paraphrase is too poor to allow any understanding of the content.
<u>Example</u> : “ <i>I a In certain years in a bookstore can help , than English , Frenche English. French</i> ” |

Table 3.3: Labels for the sensibleness dimension (Liebeck et al., 2016c)

detect text passages that indicate that one user might be posting with multiple user accounts. We leave the investigation and the decision of how to deal with such cases up to the city administration.

We would like to mention that some online participation processes restrict the submissions of ideas to local citizens, e.g., Liquid Friesland⁷ in 2012. In this particular case, citizens filled out a registration form with their personal information (name, address, and date of birth) and received an access code via mail that was required to complete the registration process (Diefenbach, 2013).

3.3 Extracting Topics in Online Participation Processes

We will now address online participation processes in which the users contribute such a high number of texts that it makes the platform incomprehensible, and we can no longer expect that the users will read several hundred texts just to understand which topics have already been discussed and what the discussion is about. In order to facilitate the participation in the discussion, we would like to utilize natural language processing techniques to provide a broad overview of the discussed topics.

For the municipal administration, politicians, or the project organizers in general, it is of course also advantageous to have an overview of what the citizens are talking about. In addition, the project organizers might also reference this information with the topics that the discussion was intended to be about and guide the discussion by bringing up new perspectives or subtopics that have not been talked about yet.

There are multiple approaches to the goal of providing an overview. Keep in mind, that we would like to have these approaches to be able to scale up for processes with more than ten thousand text comments. While choosing a technique to tackle this problem, there are multiple aspects that have to be considered:

1. **Content:** What content should be the basis for the extraction? Should the whole text content be considered or only parts (for instance limited to certain

⁷<http://www.liquidfriesland.de/>

part-of-speech-tags, e.g., nouns)?

2. **Method:** Which automated approach should be used to extract and summarize the topics discussed?

The simplest way to begin carrying out topic extraction is to extract words or phrases that occur most frequently. This makes it possible to find the most dominant discussion topic. While determining the most frequent words, it is reasonable to filter out stop words or to only use certain part-of-speech tags, e.g., nouns. However, only extracting the most frequent words can cause the effect that words from only a few, or even just one, discussion topic are extracted as a summary. This is especially the case if there is a single discussion topic that is dominating the whole discussion.

In the case of participatory budgetings, such an approach would most likely lead to suboptimal results since citizens usually submit proposals from a variety of different topics. Therefore, we think that topic modeling is a better approach to capture multiple discussion topics. In Section 3.3.2, we use the popular method *Latent Dirichlet Allocation* (LDA) (Blei et al., 2003) and show the most frequent words per topic.

Although our task is related to multi-document summarization, it is still different since we want to generate an overview of multiple documents that are about different topics. Classical multi-document summarization deals with the summarization of one story or event that is being described in multiple documents, for instance in news articles.

3. **Visualization:** How should the extracted information be visualized for the end user?

A recent user behavior study (Smith et al., 2017) has focused on the question of what makes a topic visualization good in terms of topic understanding. In a two-phased evaluation, Smith et al. (2017) evaluated four different types of topic visualizations: word lists, word lists with bars, word clouds, and network graphs. Crowdworkers were asked to label topics (comprising k most frequent words) with one to three words on the basis of topic visualizations. For 71% of the labels, at least one word of the manually produced labels is also present in the most frequent words from that topic. This also means that 29% of the labels generated by users do not include any word taken from the most frequent terms of a topic. Additionally, user-generated labels were unsurprisingly rated better than automatically generated labels that were also evaluated for a comparative baseline. The observed behavior shows that topic labeling approaches still need a lot of research to outperform human performance.

We can transfer some of their insights to the field of online participation in terms of using a dashboard-like topic overview if we think of a person that is shown different visualizations instead of a crowdworker. For topic understanding, Smith et al. (2017) did not find a significant difference between visualization techniques in the resulting label quality. This means for our dashboard use case that the

choice of the visualization technique should not be important. However, if we want to provide an overview of the discussed topics as quickly as possible (in terms of the person seeing the dashboard is able to grasp the topics quicker), we can rely on simple word lists since Smith et al. (2017) measured how long it took crowdworkers to come up with a label for a topic and they found out that topics represented by word lists allow for a significantly faster labeling than more complex visualization techniques.

We started to investigate different methods in Kahlert et al. (2017) where we addressed the task of extracting information from multiple hashtags on Twitter in order to get an understanding of the topics discussed in the hashtags. We compared multiple intuitive ways to filter words before determining the most frequent words in a hashtag. More specifically, we examined (i) no filtering at all, (ii) filtering of stop words, (iii) using part-of-speech-tags as a filter (nouns and verbs separately), (iv) only words in the first sentence of a tweet, (v) using dependencies to extract verb-noun pairs, and (vi) n-grams. Additionally, we trained LDA models with bigrams. In our opinion, the most frequent terms usually did not provide enough information to allow for an understanding of the hashtags, while methods capturing more semantics like trigrams and LDA models were more preferable.

3.3.1 Datasets

In the scope of this thesis, we will focus on the extraction of topics from three different geographical areas in Germany and their respective online participation projects. They differ in the number of texts and objective of the projects. For the city *Bonn*, we are able to compare participatory budgetings for multiple consecutive years which allows us to check whether the citizens talked about the same topics repeatedly or whether they talked about different topics. These three different regions should allow us to get a feeling for the types of projects our approach works well with, or for the types of projects that might be better handled using other approaches. We are extracting topics from online participation processes in these areas:

- **Berlin:** We begin with the dataset from the *Tempelhofer Feld* project that we already introduced extensively in Chapter 2. It revolves around the collection of ideas for the large open field of the former airport Berlin Tempelhof.
- **Tagebau Garzweiler:** As briefly mentioned in Section 3.2.2, the online participation process *Online-Konsultation zur Leitentscheidung Braunkohle* is about the reduction of the surface mining area of the *Tagebau Garzweiler* and its effects on people living in the area (with the expectation of being relocated and reimbursed for it), the environment, and the power supply of Germany.
- **Bonn:** The city of Bonn has about 300,000 inhabitants and is very experienced with online participation. In our analysis, we focus on participatory budgetings from the years 2011, 2015, and 2017.

The online participation processes also differ in the amount of text content, as listed in Table 3.4. The *Braunkohle* corpus has a low number of proposals, which were created by the state government in advance, and citizens were not able to create

new proposals. We can also observe a decreasing rate of text contributions for the participatory budgetings in Bonn. Figure 3.2 shows that the texts in the THF corpus are shorter than in the *Braunkohle* corpus.

| Corpus | # Proposals | # Comments | # Total |
|------------------|-------------|------------|---------|
| Tempelhofer Feld | 340 | 1389 | 1729 |
| Braunkohle | 7 | 1296 | 1303 |
| Bonn 2011 | 1015 | 8903 | 9918 |
| Bonn 2015 | 335 | 2937 | 3272 |
| Bonn 2017 | 55 | 109 | 164 |

Table 3.4: Number of proposals and comments per corpus

The data protection laws in Germany make it difficult to share the text contents of the processes directly although they are already publicly viewable. This is especially the case for most of the past online participation processes, including projects with a high participation rate such as the participatory budgeting in Bonn 2011 and the *Online-Konsultation zur Leitentscheidung Braunkohle*, neither of which is directly downloadable. This also makes it difficult to annotate the corpora with additional information since the texts cannot be shared. In order to bypass this restriction and to allow these past datasets to be more accessible in the future, we decided to create multiple open-source crawlers that are available on GitHub⁸. With the help of these crawlers, we are able to share the datasets indirectly by sharing the crawlers.

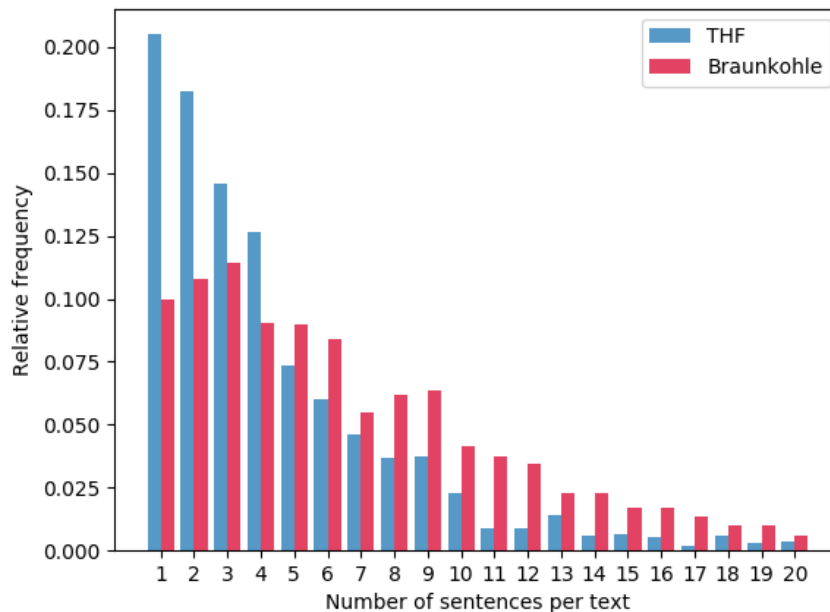


Figure 3.2: Comparison of the corpora’s sentence lengths

⁸<https://github.com/Liebeck/OnlineParticipationDatasets>

3.3.2 Topic Extraction

In Liebeck et al. (2017), we compared two different methods to extract the topics discussed in the Tempelhofer Feld project. We began by demonstrating that a word cloud displaying the most frequent words is not sufficient enough to capture an overview of the discussion. In our opinion, a better way to capture different discussion topics is topic modeling. We used the most popular topic modeling algorithm, called *Latent Dirichlet Allocation* (LDA) (Blei et al., 2003), to capture semantic topics. With LDA, each topic is a probability distribution over a fixed vocabulary that usually consists of the words in the corpus that LDA is being used upon. A typical way to represent trained LDA models is to list the most frequent $n \in \{5, 10\}$ words per topic.

We would like to point out that LDA does not have any knowledge about the meaning of individual words. The topics found by LDA result from co-occurring words in the corpus. However, LDA is often able to find several semantically coherent topics in large corpora.

Before running LDA, we must define a value for the number of topics K that LDA is supposed to find. In all of our attempts, we empirically determined the number of topics. Although there has been some work on automatically determining the number of topics (Griffiths and Steyvers, 2004), the opinion whether the retrieved topics are good or coherent is subjective, depending on the corpus, and may require background information. Chang et al. (2009) introduced the crowdsourced *word intrusion task* to evaluate how coherent topic models are. In their observation, they found that humans often agree which topic is coherent or “good” and that traditional evaluation metrics may even contradict the human judgment on topic quality. Regarding non-coherent or “bad” topics, we quote Hu et al. (2014): “*The topics discovered by topic modeling do not always make sense to end users. From the users’ perspective, there are often ‘bad’ topics. These bad topics can confuse two or more themes into one topic; two different topics can be (near) duplicates; and some topics make no sense at all.*”

In the following, we will apply LDA to the datasets from Section 3.3.1 and will assess how suitable LDA is in capturing the content of the online participation projects.

Tempelhofer Feld:

In Liebeck et al. (2017), we empirically determined $K = 15$ to yield the best LDA topics. Table 3.5 shows several good topics that reflect some of the proposals made by the citizens. Unfortunately, our LDA model also contains bad topics that we do not show here.

| Topic | Words |
|-------|--|
| #1 | Gärtner, Nutzung, Allmende, Beet, Raum |
| #2 | Zugang, Kind, Spaß, Spielplatz, Nutzung |
| #4 | Baseball, Softball, Amerikaner, Team, Trinkwasserbrunnen |
| #7 | Harald Juhnke, Straße, Weg, Way, Zeit |
| #10 | Musik, Kultur, Bühne, Wäldchen, Eingang |
| #11 | Biergarten, Bier, Jahr, Tankstelle, Form |

Table 3.5: Excerpt from the topics identified by LDA on the Tempelhofer Feld project and their corresponding five most frequent words (Liebeck et al., 2017)

For our LDA model in Liebeck et al. (2017), we removed common stop words and corpus-specific stop words before training our LDA model. Schofield et al. (2017) investigated whether stop words should be removed before or after the training an LDA model and concluded that there is almost no difference. Therefore, we remove the corpus-specific stop words from the LDA models of the other corpora after the training in order to allow for a faster and easier curation of stop words.

Braunkohle:

For the THF corpus, LDA seemed to be a good choice since there are multiple discussion topics present in the dataset. Upon manually investigating the *Braunkohle* corpus, we can see that the users were not allowed to submit proposals on their own. Although multiple proposals on the website were pre-submitted by the organizers, there seems to be only one interconnected topic, namely the reduction of the planned area for surface mining, as decided by the state government. The opinion of the citizens voiced in the comments is quite polarized into people who advocate for the reduction due to environmental reasons and into people who categorically reject it. For such a monothematic online participation project, topic models might not be a suitable choice but rather for processes with a variety of different proposals that usually occur in participatory budgetings. Nevertheless, we wanted to see whether LDA might produce meaningful topics regardless.

For the corpus, $K = 6$ turned out to be the best choice. The LDA topics are listed in Table 3.6. With knowledge of the proposals and some of the comments of the online participation process, it is possible to differentiate the topics: renewable energy in general (#1), the decision of the state government to not relocate the place *Holzweiler* (#3), the distance of the residual lake to *Holzweiler* (#5), and renewable energy's problems with the security of supply (#6).

| Topic | Words |
|-------|---|
| #1 | Klimawandel, Zukunft, Meinung, Alternative, Verfügung, Energieträger, Wind, Kraftwerk |
| #2 | Land, Entscheidung, Unternehmen, Kohle, Region, Erde, Politik, Heimat |
| #3 | Bürger, Land, Umsiedlung, Dorf, Entscheidung, CO2, Ort, Seite |
| #4 | Kohle, Energieträger, Energieversorgung, Nutzung, Kraftwerk, Zukunft, Land, Erneuerbare |
| #5 | Abstand, m, Ort, Ortschaft, L19, Grund, Restsee, Meter |
| #6 | Netz, Problem, CO2, Zufallsstrom, Energiewende, Kraftwerk, Versorgungssicherheit, Politik |

Table 3.6: LDA Topics for Braunkohle

However, we have to admit that it is not easy to understand these topics without background knowledge and that the topics appear to be very similar which is due to the very narrow topical focus of the platform. Upon comparing multiple LDA models, we noticed that the topic about the residual lake can be found consistently for different values of K . We also noticed that it was not sufficient enough to only display the five most frequent words per topic and, therefore, increased the number of words to achieve a better differentiation between the topics.

Let us now take a look at Bonn and its participatory budgetings from 2011, 2015, and 2017.

Bonn 2011:

The LDA topics for 2011 are listed in Table 3.7. Given the large size of the dataset, LDA can indeed capture multiple good topics: the Bonner *Beethovenhalle* (#3), the availability of schools and kindergartens (#4), cultural topics like the *Frauenmuseum* (#9) and Bonn’s participation in carnival with its associated costs for the city (#15), and traffic (#1 and #10). Topic #5 discusses the music schools and savings opportunities in the musical sponsorships of the city. It is interesting to see that LDA also captured a general topic about the moderation of the online participation process (#2) and a general topic about saving suggestions (#11). Topic #14 treats public transport and swimming pools are being discussed in topic #7.

| Topic | Words |
|-------|---|
| #1 | Straße, Auto, Kontrolle, Parkplatz, Uhr |
| #2 | Moderation, Teilnehmer, Liebe, Teilnehmerin, Buch |
| #3 | Orchester, Beethovenhalle, Oper, Gebäude, Halle |
| #4 | Schule, Kind, Stelle, Schüler, Kindergarten |
| #5 | Kind, Musikschule, Familie, Eltern, Geld |
| #6 | Stadthaus, Wasser, Rheinaue, Fahrt, Gewinn |
| #7 | Bürger, Bad, Geld, Freibad, Verwaltung |
| #8 | Verwaltung, Leistung, Dank, Haushalt, Beitrag |
| #9 | Verein, Frau, Museum, Frauenmuseum, Kunst |
| #10 | Ampel, Verkehr, Rahmen, Kreuzung, Rat |
| #11 | Leistung, Geld, Einsparung, Kürzung, Beispiel |
| #12 | Kürzung, Mensch, Angebot, Kind, Leistung |
| #13 | Oper, Theater, Hund, Preis, Bürger |
| #14 | Bus, Linie, Bahn, Innenstadt, Minute |
| #15 | Karneval, Kultur, Mitarbeiter, Veranstaltung, Leute |

Table 3.7: LDA Topics for Bonn 2011

Bonn 2015:

The Bonn participatory budgeting from 2015 is about one-third of the size of the project from 2011. Given that there are still over 300 proposals, it is reasonable to assume that there are more than ten discussion topics. LDA models with a smaller number of topics yielded topics with dissatisfying semantic coherence. With a higher number of topics, the results become more interpretable. However, it was still difficult to choose the number of topics for Bonn 2015 since the trained models for $K = 14$ and $K = 15$ both found multiple good topics but also mixed topics or incomprehensible ones. In the end, we decided to list an excerpt of the LDA topics for $K = 14$ in Table 3.8. LDA found topics about holiday care (#5), the closing of libraries (#6), two cultural topics (#7 and #9), again a topic about moderation (#10), and citizens that complain about the lack of parking lots (#13).

| Topic | Words |
|-------|---|
| #4 | Kunstverein, Künstler, Rat, Projekt, Ergebnis |
| #5 | Kind, Eltern, Beitrag, Ferienbetreuung, Familie |
| #6 | Kind, Bücherei, Bibliothek, Bildung, Schließung |
| #7 | Festspielhaus, Schule, Kind, Beethovenhalle, Park |
| #9 | Oper, Kultur, Theater, Orchester, Konzert |
| #10 | Moderation, Dank, Thema, Liebe, Dialog |
| #13 | Auto, Gebühr, Platz, Innenstadt, Parkplatz |

Table 3.8: Excerpt from the LDA topics from Bonn 2015 with $K = 14$ **Bonn 2017:**

With only 55 proposals and 109 comments, the participation rate for the participatory budgeting in 2017 was very low. In order to assess how expressive LDA is for this small corpus, we read the proposals in advance before looking at the trained LDA models. Upon investigating the LDA models, we noticed that the topics were not clearly separated, regardless of the number of topics. We also experimented with different weighting schemes but without success. Although there are occasionally good or coherent topics, e.g., the closing swimming pools [*Bad, Schließung, Hallenbad, Frankenhades, Frankenbad*], the expensive renovation of the *Beethovenhalle* [*Sanierung, Halle, Konzept, Beethovenhalle, Kostenobergrenze*], and meta-discussions about the success and acceptance of the online participation process [*Bürger, Haushalt, Dialog, Verwaltung, Bürgerdialog*], they are, however, distributed over different models and cannot be found consistently. In summary, LDA is not suitable for topic extraction on such a small project with many different proposals. For these cases, word clouds or a manual view of the platform will probably be sufficient.

Overall, we have shown that LDA works as a method for topic extraction but that its success depends on the type of online participation process and the size of the dataset. For example, LDA has generally worked well for participatory budgetings, except for Bonn 2017, where the number of text contributions was too low. Despite the monothematic structure of the *Braunkohle* platform, a number of discussion topics could be identified, although their interpretability was only possible with background knowledge of the process. However, citizens and administrators usually have this kind of knowledge when they engage in the discussion.

For all of these corpora, we trained several LDA models using a different number of topics. Lacking a reliable way to automatically determine the best model, we had to manually choose the model with the most “good” topics, which was difficult due to the subjective nature of this approach.

3.3.3 Topic Labeling

Topic labeling is the NLP task of describing a set of words with a superordinate or summarizing term. For topic models, such as LDA, topic labeling can be used to automatically summarize the most frequent words per topic. In a text analysis pipeline for extracting discussion topics, topic labeling can be considered as the next step after topic modeling, as illustrated in Figure 3.3.

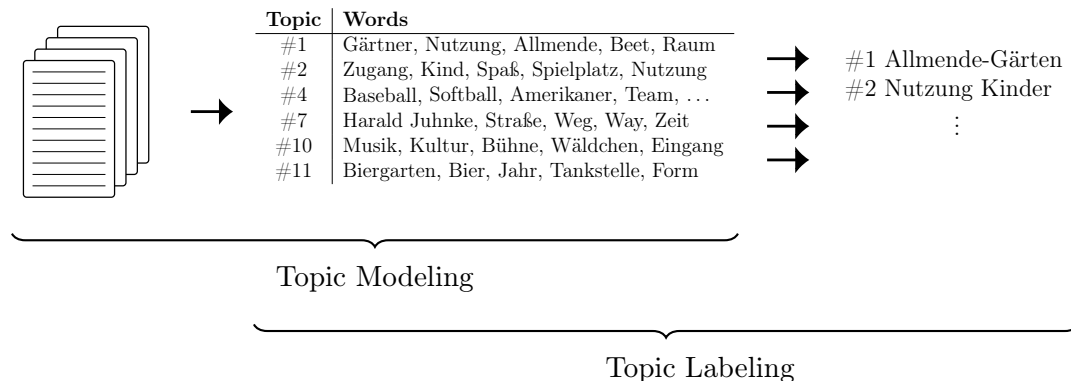


Figure 3.3: Combination of topic modeling and topic labeling

We experimented with a system to predict a topic label for the top n words of an LDA topic. Given that a topic labeling technique should be able to output a good fitting label for a wide variety of top n words from topic models, the output vocabulary that represents all possible labels should be very large. This is especially necessary if the words which are to be labeled can come from a variety of discussion topics. In our prototype, we decided to use a very large output vocabulary based on the titles of Wikipedia articles.

The evaluation of topic labeling approaches is tricky since each possible label for a topic requires a judgment concerning how good the label fits the topic. Otherwise, it is not possible to compare which one of two labels is a better fit. In the current status of the research field, such judgments or annotations of topic labeling approaches are created by humans, usually by crowdworkers (Lau et al., 2011; Bhatia et al., 2016). Although an evaluation is also possible by the researchers involved in the respective research groups, an evaluation of independent crowdworkers is perceived as more impartial. Since such judgments are somewhat subjective, even with annotation guidelines, each label-topic pair is usually rated by multiple annotators and their judgments are averaged.

We are interested in topic labeling for German, especially for topic models generated from online participation processes. In order to allow for a rapid prototyping of topic labeling techniques, we decided to use an annotated English dataset from Bhatia et al. (2016), which is based on the previous work from Lau et al. (2011). The dataset comprises 228 topics with the most ten frequent words each. Bhatia et al. (2016) presented each of these topics to crowdworkers along with 19 possible labels. The workers had to rate each topic-label pair with a score $s \in \{0, 1, 2, 3\}$ indicating how well the label fits the topic. After filtering out bad crowdworkers, the judgments were averaged into a single score. Table 3.9 shows the topic terms of four topics along with some of the 19 labels and their respective crowdsourced scores.

| Topic Terms | Labels | Score |
|---|-----------------------------------|-------------|
| school, student, university, college, teacher, class, education, learn, high, program | school | 2.67 |
| | education | 2.67 |
| | primary education | 2.0 |
| | teacher | 2.0 |
| | pre-medical | 0.71 |
| church, arch, wall, building, window, gothic, nave, side, vault, tower | church architecture | 2.71 |
| | romanesque architecture | 1.67 |
| | barrel vault | 1.0 |
| | rose window | 0.8 |
| san, francisco, diego, los, california, angeles, 2006, 2005, conference, october | san diego | 1.83 |
| | michael williams (defensive back) | 1.22 |
| | southern california | 1.0 |
| | | |
| computer, internet, software, system, microsoft, technology, company, service, network, program | computer network | 2.57 |
| | open source | 2.14 |
| | internet | 1.8 |

Table 3.9: Example topics from Bhatia et al. (2016) with some of their labels and respective annotated scores. The bolded labels represent the predicted labels of our *LDA embedding* prototype.

In the scope of this thesis, we pursued an unsupervised approach for topic labeling which we call *LDA embedding*. Let $topic\ term_{ij}$ denote the i -th term in topic j and $label_{kj}$ is the k -th label for topic j . For each term in the dataset, we retrieved the corresponding Wikipedia article with the same name and represented it with an LDA distribution of the words in the article. We trained LDA models on an English Wikipedia dump and experimented with different dimensionalities $D = \{100, 150, 200, 300\}$. For each topic j , we calculated the score

$$\text{dist}_{LDA}^j(label_{kj}) := \Phi(\{\text{dist}_{LDA}(label_{kj}, topic\ term_{ij})\}_{1 \leq i \leq 10})$$

of how well label $label_{kj}$ fits topic j . The distance between the LDA distributions of the Wikipedia articles for $label_{kj}$ and $topic\ term_{ij}$ is denoted by $\text{dist}_{LDA}(label_{kj}, topic\ term_{ij})$. We evaluated four different ways to calculate the distance: (i) cosine similarity, (ii) Jensen–Shannon divergence (Lin, 1991), (iii) Euclidean distance, and (iv) Kullback–Leibler divergence (Kullback and Leibler, 1951).

For each possible label, we calculated distances to the ten topic terms and aggregated them with Φ into a single score. For Φ , we experimented with three functions: sum, median, and mean. In order to determine the best-fitting label for topic j , we selected the label with the smallest distance to the topic terms.

Unfortunately, our prototype was not able to beat the previous baselines from Lau et al. (2011) and Bhatia et al. (2016). Looking at the dataset and individual topics more closely, we can see that our approach can predict the best-ranked labels in some cases, e.g., for the first and fourth topic in Table 3.9, where our predicted labels are marked as bold. For the second topic, our predicted label *romanesque architecture* has a much lower score than the best-ranked label *church architecture*. We do not agree with this assessment and want to point out that such ratings are subjective. In addition, there

are other instances of scores that we cannot comprehend. If we take a look at the third topic which is about three cities in California, we can see that the best-ranked label *san diego* has a score of 1.83. We cannot understand how the label of an American football player named *Michael Williams* is ranked with 1.22. Our approach predicted the much better fitting label *southern california*, which geographically describes two of the three cities. However, our prediction is labeled with a 1.0 which is worse than the football player. The fourth topic is another example of incomprehensible high judgments from the crowdworkers since *open source* is rated higher than *internet* and the most frequent topic words indicate nothing of source code development or open source licenses.

In summary, our approach still looks promising for further research and we are very curious to see how it performs on other datasets.

3.4 Future Work

We will now outline further research aspects regarding topic extraction:

- **Semantic similarity:**

For our work on semantic similarity, we would like to experiment if the *Word Mover's Distance* (Kusner et al., 2015) can be used as an additional feature for our supervised approaches.

- **Topic modeling:**

In the scope of this thesis, we have limited ourselves to only use LDA as topic modeling method. In our ongoing research, we are currently comparing the results of several other topic modeling methods in order to find the best working method for online participation processes.

We agree with Hu et al. (2014) that the “*users of topic models are the ultimate judge of whether a topic is ‘good’ or ‘bad.’*” Therefore, we want to compare different topic models by showing them to moderators and users who should rate the returned topics in terms of understandability. In addition, we would like to measure the time it takes the users to understand the topics and include this metric into our decision for a suitable topic modeling technique. Inspired by Smith et al. (2017), we might also use crowdsourcing to compare our models.

Another interesting approach to include user feedback and to improve topic models is to make them interactive. Hu et al. (2014) proposed to allow end users to merge two topics into one and to split a mixed topic into two separate ones. We want to evaluate if such an approach is helpful in our application domain as well.

We will also evaluate which topic modeling method works best for the participatory budgetings from *Cologne* and *Darmstadt*.

The output vocabulary plays a huge role in topic extraction. Therefore, we would like to reduce the size of the vocabulary by correcting spelling errors. In addition, we would like to evaluate techniques to further compress our extracted words semantically, for instance with semantic nets.

As mentioned in Section 3.3.2, the manual decision to choose an optimal number of topics for LDA was difficult. We would like to experiment with techniques to automatically find that optimal number. We will evaluate if these techniques are

of any benefit to us and if they also contradict our human judgment, as observed in Chang et al. (2009).

We would also like to pursue a combination of topic modeling and summarization techniques: Given a topic model for a corpus, we could group all documents with the same dominant topic and try to use extractive summarization techniques to create an overview of the particular discussion topic.

- **Visualization:**

Besides topic modeling, we would like to experiment with different visualization techniques that could also provide an overview of the discussion, such as DLATK (Schwartz et al., 2017) and Scattertext (Kessler, 2017). Both visualization packages could be used to compare two online participation processes from different years in the same city, for instance, to identify common topics in successive participatory budgetings.

- **Topic labeling:**

To the best of our knowledge, there is currently no German topic labeling dataset with crowdsourced annotations that is comparable to Bhatia et al. (2016). Therefore, we would like to create a comparable crowdsourced dataset for German and use it to benchmark our approach. In addition, we would like to evaluate existing topic labeling approaches on that dataset, especially the work from Bhatia et al. (2016).

In our approach, we represented each term with the LDA distribution of the corresponding Wikipedia article. We would like to evaluate whether it is beneficial to include Wikipedia’s link structure. To that end, we will try to add more depth to our approach by representing an article by the LDA distributions of the first n linked articles and by adapting our distance calculation accordingly.

4

AUTHOR PROFILING

In this chapter, we focus on author profiling and present our multilingual approach, which achieved an excellent ranking in the PAN author profiling in 2016. Afterwards, we evaluate how our approach performs on a German online participation dataset.

4.1 Introduction into Author Profiling

The research area author profiling focuses on the automatic identification of demographic attributes from an author, given some of his texts, e.g., from social media posts. So far, demographic inference has been applied for a variety of attributes, such as detecting the gender (Rao et al., 2010; Burger et al., 2011), age (Rao et al., 2010), location (Eisenstein et al., 2010), occupation (Preoțiuc-Pietro et al., 2015), personality traits (Mairesse et al., 2007), and political orientation (Conover et al., 2011). Author profiling was also used to detect medical risks by screening for behavior patterns, such as the detection of depression (Schwartz et al., 2014) or heart disease (Eichstaedt et al., 2015) by analyzing posts on Facebook and Twitter, respectively.

In terms of identifying personality traits, the Big Five model (L. Goldberg, 1993) (comprising openness, conscientiousness, extraversion, agreeableness, and neuroticism) is often used. Previous research by J. Chen et al. (2015) showed that predicting personality traits based on Twitter tweets can be used to target advertisements individually as users with high openness and low neuroticism are more likely to respond to advertisements and have a higher conversion rate.

We also experimented with the Big Five model in Liebeck et al. (2016b) for predicting personality types of students based on their Java source code.

We would like to mention that it can be difficult to predict certain user attributes, such as gender or age, on the basis of a single text message. Considering the following example, it is even difficult for humans to infer whether the author's gender is male or female:

I bought an iPhone today 😍

However, the more text content we have of a user, the easier it becomes to predict these traits, as studies such as Burger et al. (2011) and Ungar (2017) have investigated.

Author profiling has attracted a lot of attention in the NLP community, which resulted in shared tasks with a variety of attributes to predict. The most notable author profiling challenge takes place yearly at PAN (Rangel et al., 2013; Rangel et al., 2014; Rangel et al., 2015; Rangel et al., 2016; Rangel et al., 2017). In 2017, a new gender identification shared task for Russian (RusProfiling, 2017) was created.

4.2 Use Cases for Online Participation Processes

There are several use cases for author profiling in online participation projects. Probably the most important of them is to estimate the age and gender distribution of the citizens involved in the discussion since the opinions of each group should be heard. If opinions from a minority are missing in the comments, it would be possible to approach them offline and include their point of view and arguments. Although we technically first have to predict the gender and the age for each individual user, we are not interested in this granularity level but only in the general distribution. A possible output from our system might be that 70% of the participants are male and above age 50. A city administration would now be able to use this information to approach younger citizens and underrepresented minorities.

Author profiling could also be applied to D-BAS (Krauthoff et al., 2016) where a user's gender and age could be inferred from his or her entered text. If an argument in D-BAS is only responded to by a specific user group, for instance, males, author profiling could be used to ask currently underrepresented groups, e.g., females, for their opinions on that argument. Since D-BAS aims to collect short arguments from the users, the training for such a classifier can probably not be conducted on D-BAS data alone but rather on in-domain data.

It is also possible to predict other user attributes or user behavior. In the case of online participation, it might be interesting to predict users that are prone to negatively influence or even bully other discussion participants and alert the moderators to control whether the user in question adheres to the rules of conduct. A study of Ziegele and Jost (2016) showed that a high deliberative discussion atmosphere can lead to a high willingness to participate in a discussion. Therefore, we would like to ensure such a discussion atmosphere by automatically supporting moderators. The research from Cheng et al. (2015) on antisocial behavior and banned users in comment areas of news websites is quite promising and contains features we might adopt for our author profiling approach. Fortunately, users of online participation projects usually demonstrate civil behavior and automated moderation systems are currently not needed.

4.3 Our Multilingual and Cross-domain Approach

We participated (Modaresi et al., 2016a) in the age and gender prediction of the PAN challenge in 2016 (Rangel et al., 2016). The iteration of 2016 comprised three languages (English, Spanish, and Dutch). The difficulty, in comparison with the previous years, was that the training occurred on tweets from Twitter and the evaluation on another domain that was unknown to us and the other participants at the time of the challenge.

Since the length of a tweet is limited and Twitter users tend to use a very specific way of maximizing the amount of content within the length boundaries, we decided to use genre-independent features. Since none of us was able to understand Spanish or Dutch, we also decided not to use language-specific features. Additionally, all our features were normalized in order to account for the different text lengths between the source and the target domain.

Logistic regression was used as a machine learning technique. As features, we used combinations of word unigrams, word bigrams, character 4-grams within word boundaries, punctuation features, and a relative spelling error feature by using Hunspell¹ with LibreOffice dictionaries.

Using our approach, we were able to achieve very good results ranking as the second team out of 22. For English, we achieved first place for gender detection and tied for second place in terms of joint accuracy. For Spanish, we were able to tie for first place.

4.4 Experiments on a German Online Participation Process

So far, our approach has not been tested on German data. To the best of our knowledge, there is currently only one German corpus, named *TwiSty*, for author profiling in German (Verhoeven et al., 2016) comprising social media data. In the scope of this thesis, we will evaluate how good our approach works with an in-domain setting for a German online participation project, namely the *Braunkohle* corpus. A deeper analysis setup, including additional features, the release of our dataset, and cross-domain evaluations are planned for an original publication. Here, we only focus on gender prediction.

4.4.1 Dataset Creation and Annotation

For our evaluation in the online participation domain, we use the text comments from the *Braunkohle* corpus that were retrieved with our crawler². In total, 1296 comments were written by 441 unique users. We have annotated each user name with gender information $g \in \{\text{male, female, unknown}\}$ solely based on their first name. With a distribution of 44 women, 380 men, and 17 authors of unknown gender, the gender distribution is heavily skewed toward men. For our experiment, we can use 144 comments written by women and 1064 comments from men.

4.4.2 Evaluation

For a better comparison with the PAN author profiling challenge, we will also use accuracy as evaluation metric and conduct our experiments on a subset of the annotated data with balanced class distribution. Besides only having a small number of data for training and testing a classifier, an additional problem arises when taking a closer look at the number of comments each user wrote: about a third of the 144 female comments are written by a single user. To prevent this single user from having too

¹<http://hunspell.github.io/>

²<https://github.com/Liebeck/OnlineParticipationDatasets>

great an influence on the classification result, we will perform multiple evaluations with different randomized subsets, each of which contains 120 comments per gender. In total, this leaves us with 240 comments per subset. In order to evaluate how stable our classification results are over the five subsets, we will perform five-fold cross-validations.

| Subset | Scores | | | | | Mean | Variance |
|--------|--------|--------|--------|--------|--------|-------|----------|
| | Fold 1 | Fold 2 | Fold 3 | Fold 4 | Fold 5 | | |
| 1 | 75.00 | 52.08 | 77.08 | 64.58 | 75.00 | 68.75 | 88.54 |
| 2 | 68.75 | 64.58 | 79.17 | 68.75 | 75.00 | 71.25 | 26.74 |
| 3 | 62.50 | 79.17 | 60.42 | 66.67 | 62.50 | 66.25 | 45.83 |
| 4 | 66.67 | 72.92 | 66.67 | 68.75 | 75.00 | 70.00 | 11.46 |
| 5 | 81.25 | 60.42 | 66.67 | 70.83 | 64.58 | 68.75 | 50.35 |

Table 4.1: Evaluation results of our author profiling approach on the *Braunkohle* corpus. Each row represents a different subset of the corpus along with the results for the respective fivefold cross-validation.

The results of the five-fold cross-validation of our approach from Modaresi et al. (2016a) on the *Braunkohle* corpus are listed in Table 4.1. First of all, all results are better than a random classification baseline which would yield $\approx 50\%$ accuracy since the dataset is balanced. If we take a look at the five subsets, we can see that the mean classification accuracy is between 66.25% and 71.25%. If we compare this range with our gender predictions (Modaresi et al., 2016a) in the PAN challenge with 75.65% for English and 69.64% for Spanish, our scores for German are lower than for English and comparable with Spanish although our German dataset was much smaller.

However, if we take a look at the individual subsets and compare the individual accuracy scores with their respective mean, we can see that our accuracy scores are not stable and have a high variance. This is probably due to the small dataset size and too infrequent patterns in the textual data.

Nevertheless, the averaged result of all subsets is 69%, which is satisfactory given the size of the dataset. In the future, we will evaluate how a German gender prediction model trained on another domain will perform on our corpus.

4.5 Future Work

In the future, we would like to evaluate our approach in a cross-domain setting on multiple German datasets. Additionally, we will benchmark some of the features described in Section 2.6.1 as well as state-of-the-art techniques.

Character embeddings (Bojanowski et al., 2017), as explained in Section 2.6.1, have already been used in the author profiling challenge at PAN 2017 (Rangel et al., 2017) by Franco-Salvador et al. (2017) who experimented with character n-gram embeddings and achieved 7th place for the gender prediction of English texts. Miura et al. (2017) combined word embeddings and character n-gram embeddings and achieved 6th place. We intend to extend these works by combining their approaches with classical features.

Spelling errors are also identified to be a helpful feature to detect an author’s native language. L. Chen et al. (2017) found out that character n-grams of misspelled words

can slightly improve previous benchmarks. We will experiment with this feature as well in our future work by searching for spelling errors in texts written by male and female authors in order to determine important character n-grams of misspelled words for gender classification.

5

CONCLUSION AND OUTLOOK

This thesis has focused on automatically analyzing German online participation processes using natural language processing techniques. For successful processes with thousands of text contributions, users can easily be overwhelmed by the sheer amount of information from previous posts. After the participation phase of an online participation process ended, the organizers have to manually evaluate the suggestions and comments from the users or citizens. Since this analysis is very time consuming, it is necessary to automatically assist the organizers in the manual evaluation.

However, software and algorithms specifically tailored for this automatical supported evaluation of German online participation processes did not exist and were a research gap that we addressed with our works in the research fields argument mining, topic extraction, and author profiling. Although methods already existed in all three research fields, they are usually focused on English text data. In the scope of this thesis, we investigated how well existing and new techniques perform on German data in the application domain of online participation.

In Chapter 2, we focused on argument mining. We started by identifying an appropriate argument model for our application domain, created annotation guidelines, and annotated a dataset. Afterwards, we addressed two common machine learning tasks in argument mining: argument identification and argument classification. We have worked on both classification tasks with two different approaches. First, we pursued a classical machine learning approach with feature engineering. We evaluated multiple classifiers, a multitude of features, and incorporated state-of-the-art features into our research. Second, we benchmarked five different deep learning architectures for both subtasks as deep learning is becoming more common in natural language processing. For the identification of argumentative content, the classical machine learning approach yielded the best results with 69.71% macro-averaged F_1 . The classification of argument components was best solved with deep learning and the best model achieved a score of 68.59%.

As our approach is currently limited to classifications on a sentence level, an evaluation of a finer granular sequence tagging approach is the next imminent step. We did not evaluate an argument linking subtask on our dataset due to the difficulty of

correctly annotating relations between argument components with the high number of multi-document discourse between the citizens in the Tempelhofer Feld online participation project. In the future, we plan to revisit argument linking on smaller datasets.

In Chapter 3, we focused on topic extraction. We began with necessary preliminary work by creating the new German lemmatizer IWNLP which is based on the freely available Wiktionary. Then, we worked on semantic textual similarity between two sentences and paraphrase detection. Subsequently, we shifted our focus to extracting topics from online participation projects. In the scope of this thesis, we pursued two approaches to extract discussion topics from online participation projects and we believe that the topic modeling method Latent Dirichlet Allocation is applicable and very promising in our application domain. We applied LDA to several online participation projects and discussed the retrieved topic models. In the remainder of Chapter 3, we concentrated on topic labeling as the subsequent step after topic modeling. We presented an approach based on LDA embeddings and discussed the results of our prototypical implementation which will we expand in the future.

In Chapter 4, we concentrated on author profiling and presented our cross-lingual approach from the PAN author profiling challenge in 2016. We discussed why author profiling in online participation projects can be useful to estimate the demographic distribution of the participants. Then, we applied our approach for gender detection on the *Braunkohle* corpus and achieved accuracy scores between 66.25% and 71.25% for five different subsets of the corpus. In the future, we will conduct further research into cross-domain author profiling on German texts.

Our research on argument mining was carried out as interdisciplinary research between the two disciplines computer science and communication and media studies as part of the PhD program *Online Participation*. Our approach of combining the strengths of both disciplines regarding manual and automatic content analysis has proven to be very useful. Especially in communication and media studies, automatic text analysis is currently becoming popular. We think we have set a good example for interdisciplinary work and encourage more researchers to work in interdisciplinary teams.

In Liebeck et al. (2017), we already outlined some design decisions regarding the automatic prediction of emotions. We believe that further research into how participants of online participation projects respond to each other emotionally is required and we would like to pursue this research direction in the future.

The text content of all platforms we studied was extracted from forum-like platforms where participants were able to enter content in free form. Forums have the advantage of a low technical hurdle and participants can quickly join online discussions. However, forums do not scale as a discussion platform and participants who join the discussion at a later point in time can be negatively affected by the effects described by Jones et al. (2004) due to the structure of forums. A difficulty in natural language processing is the ambiguity of language and the varying quality of the text content that additionally impacts the performance of automatic analysis approaches. An approach fundamentally different from extracting arguments from texts is the already mentioned dialog-based approach with D-BAS (Krauthoff et al., 2016), where users are not allowed to post long text content in a free text format but they are instead forced into a structured discussion by the platform. Such an approach requires argument mining to a much lesser extent and makes it easier to focus on specific arguments and their

relations since they do not have to be mined from text content. Additionally, the online participation projects based on D-BAS can be moderated through a community effort. It will be exciting to observe if this kind of discussion platform can prevail.

In argument mining, we have so far only focused on the online participation process Tempelhofer Feld and on about one-third of its content. We are currently expanding our focus and decided to annotate the entire Tempelhofer Feld process and the entire *Online-Konsultation zur Leitentscheidung Braunkohle* process. This time, we include a lot of other labels, called variables in the context of social sciences, into our annotation. In addition to argument components, we now also capture emotions (binarized into positive and negative emotions), narrations, humor, disrespectfulness, greetings, proposed solutions in terms of a compromise between conflicting points of view, information questions, questions of justifications, empathy, and relations between different comments. From a social science perspective, the research goal is to discern how these factors influence the discourse between the users, e.g., whether negative emotions are expressed after disrespectfulness. For the computer science perspective, the annotated data can be used for machine learning. The annotation of two corpora also allows for a cross-dataset evaluation in terms of training a model on one online participation process and evaluating on the other one. If we are able to achieve good results in this evaluation, we can get an impression of how our trained model generalizes across datasets.

As long as forums dominate the technical online participation platforms, topic extraction is probably the area where we can help best with technical solutions. We intend to integrate feedback from politicians, government employees, and technical service providers on our methods and the retrieved topics from Chapter 3 very soon. In addition, we believe that we can help by providing methods to automatically categorize text comments and to automatically detect duplicate content. Furthermore, we would like to develop an open-source framework for topic extraction, interactive topic modeling, and topic visualizations. We will work together with our technical and administrative contacts to test this framework under live conditions.

6

PUBLICATIONS

6.1 What to Do with an Airport? Mining Arguments in the German Online Participation Project Tempelhofer Feld

Matthias Liebeck, Katharina Esau, and Stefan Conrad (2016a). “What to Do with an Airport? Mining Arguments in the German Online Participation Project Tempelhofer Feld”. In: *Proceedings of the Third Workshop on Argument Mining (ArgMining2016)*. Association for Computational Linguistics, pp. 144–153.

Contributions: The annotation scheme was developed jointly by Matthias Liebeck and Katharina Esau. Matthias Liebeck implemented the machine learning approach and prepared the manuscript.

Sections: 2.3, 2.4, 2.5, 2.6.1, 2.6.2.1

Status: Published.

URL: <https://aclweb.org/anthology/W16-2817>

6.2 Mining Arguments in Online Participation: Möglichkeiten und Grenzen manueller und automatisierter Inhaltsanalyse zur Erhebung von Argumentkomponenten

Katharina Esau, Matthias Liebeck, and Christiane Eilders (2017). “Mining Arguments in Online Participation: Möglichkeiten und Grenzen manueller und automatisierter Inhaltsanalyse zur Erhebung von Argumentkomponenten”. In: *Polkomm 2017 - „Disliken, diskutieren, demonstrieren – Politische Partizipation im (Medien-)Wandel“*.

Contributions: Matthias Liebeck performed simulations for the evaluation in the manuscript. Katharina Esau and Matthias Liebeck contributed equally to the preparation of the manuscript.

Sections: 2.6.2.2

Status: Published.

URL: <https://dbs.cs.uni-duesseldorf.de/publikationen/2017/polkomm2017.pdf>

6.3 Text Mining für Online-Partizipationsverfahren: Die Notwendigkeit einer maschinell unterstütz- ten Auswertung

Matthias Liebeck, Katharina Esau, and Stefan Conrad (2017). “Text Mining für Online-Partizipationsverfahren: Die Notwendigkeit einer maschinell unterstützten Auswertung”. In: *HMD Praxis der Wirtschaftsinformatik* 54.4. Schwerpunktheft „Online Participation“, pp. 544–562.

Contributions: Matthias Liebeck designed the experiments, evaluated features for the machine learning tasks, and implemented the topic modeling approach. The manuscript was prepared jointly by Matthias Liebeck and Katharina Esau.

Sections: 2.6.2.3, 3.3.2

Status: Published.

URL: <https://link.springer.com/article/10.1365/s40702-017-0321-6>

6.4 IWNLP: Inverse Wiktionary for Natural Language Processing

Matthias Liebeck and Stefan Conrad (2015). “IWNLP: Inverse Wiktionary for Natural Language Processing”. In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Association for Computational Linguistics, pp. 414–418.

Contributions: The research and the preparation of the manuscript was done entirely by Matthias Liebeck under the supervision of Stefan Conrad.

Sections: 3.1

Status: Published.

URL: <https://aclweb.org/anthology/P15-2068>

6.5 HHU at SemEval-2016 Task 1: Multiple Approaches to Measuring Semantic Textual Similarity

Matthias Liebeck, Philipp Pollack, Pashutan Modaresi, and Stefan Conrad (2016d). “HHU at SemEval-2016 Task 1: Multiple Approaches to Measuring Semantic Textual Similarity”. In: *Proceedings of the 10th International Workshop on Semantic Evaluation*, pp. 595–601.

Contributions: The research was conducted jointly by Matthias Liebeck, Philipp Pollack, and Pashutan Modaresi. Matthias Liebeck contributed the *Overlap method*, participated in the feature engineering, and supervised Philipp Pollack. The manuscript was prepared by Matthias Liebeck.

Sections: 3.2.1

Status: Published.

URL: <https://aclweb.org/anthology/S16-1090>

6.6 Evaluating Safety, Soundness and Sensibleness of Obfuscation Systems

Matthias Liebeck, Pashutan Modaresi, and Stefan Conrad (2016c). “Evaluating Safety, Soundness and Sensibleness of Obfuscation Systems”. In: *Working Notes of CLEF 2016 - Conference and Labs of the Evaluation Forum*, pp. 920–928.

Contributions: Matthias Liebeck contributed in the design of the evaluation methods for the dimensions *soundness* and *sensibleless*. The manuscript was prepared jointly by Matthias Liebeck and Pashutan Modaresi.

Sections: 3.2.2

Status: Published.

URL: <http://ceur-ws.org/Vol-1609/16090920.pdf>

6.7 Understanding Trending Topics in Twitter

Roland Kahlert, Matthias Liebeck, and Joseph Cornelius (2017). “Understanding Trending Topics in Twitter”. In: *Datenbanksysteme für Business, Technologie und Web (BTW 2017) - Workshopband*. GI, pp. 375–384.

Contributions: The research was conducted jointly by Matthias Liebeck, Roland Kahlert and Joseph Cornelius. Matthias Liebeck designed the experiments, especially for the *Topic Detection* section (which were implemented by Joseph Cornelius), and supervised Roland Kahlert. The manuscript was prepared jointly by Matthias Liebeck and Roland Kahlert.

Sections: 3.3

Status: Published.

URL: http://btw2017.informatik.uni-stuttgart.de/slidesandpapers/F-19-4/paper_web.pdf

6.8 Exploring the Effects of Cross-Genre Machine Learning for Author Profiling in PAN 2016

Pashutan Modaresi, Matthias Liebeck, and Stefan Conrad (2016a). “Exploring the Effects of Cross-Genre Machine Learning for Author Profiling in PAN 2016”. In: *Working Notes of CLEF 2016 - Conference and Labs of the Evaluation Forum*, pp. 970–977.

Contributions: Matthias Liebeck contributed with the feature engineering and the implementation of features. Pashutan Modaresi and Matthias Liebeck contributed equally to the preparation of the manuscript.

Sections: 4.3

Status: Published.

URL: <http://ceur-ws.org/Vol-1609/16090970.pdf>

REFERENCES

- Muhammad Abdul-Mageed and Lyle Ungar (2017). “EmoNet: Fine-Grained Emotion Detection with Gated Recurrent Neural Networks”. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, pp. 718–728.
- Aseel Addawood and Masooda Bashir (2016). ““What Is Your Evidence?” A Study of Controversial Topics on Social Media”. In: *Proceedings of the Third Workshop on Argument Mining (ArgMining2016)*. Association for Computational Linguistics, pp. 1–11.
- Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe (2016). “SemEval-2016 Task 1: Semantic Textual Similarity, Monolingual and Cross-Lingual Evaluation”. In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. Association for Computational Linguistics, pp. 497–511.
- Yamen Ajjour, Wei-Fan Chen, Johannes Kiesel, Henning Wachsmuth, and Benno Stein (2017). “Unit Segmentation of Argumentative Texts”. In: *Proceedings of the 4th Workshop on Argument Mining*. Association for Computational Linguistics, pp. 118–128.
- Ahmet Aker, Alfred Sliwa, Yuan Ma, Ruishen Liu, Niravkumar Borad, Seyedeh Ziyaei, and Mina Ghbadi (2017). “What works and what does not: Classifier and feature analysis for argument mining”. In: *Proceedings of the 4th Workshop on Argument Mining*. Association for Computational Linguistics, pp. 91–96.
- Stefanie Albert, Jan Anderssen, Regine Bader, Stephanie Becker, Tobias Bracht, Sabine Brants, Thorsten Brants, Vera Demberg, Stefanie Dipper, Peter Eisenberg, Silvia Hansen, Hagen Hirschmann, Juliane Janitzek, Carolin Kirstein, Robert Langner, Lukas Michelbacher, Oliver Plaehn, Cordula Preis, Marcus Pussel, Marco Rower, Bettina Schrader, Anne Schwartz, Smith George, and Hans Uszkoreit (2003). *TIGER Annotationsschema*. Tech. rep. Universität Potsdam, Universität Saarbrücken, Universität Stuttgart.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio (2014). “Neural Machine Translation by Jointly Learning to Align and Translate”. In: *CoRR* abs/1409.0473.
- Shraey Bhatia, Jey Han Lau, and Timothy Baldwin (2016). “Automatic Labelling of Topics with Neural Embeddings”. In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 953–963.
- Anders Björkelund, Bernd Bohnet, Love Hafdell, and Pierre Nugues (2010). “A High-Performance Syntactic and Semantic Dependency Parser”. In: *Proceedings of the*

- 23rd International Conference on Computational Linguistics: Demonstrations*. COLING '10. Association for Computational Linguistics, pp. 33–36.
- David M. Blei, Andrew Y. Ng, and Michael I. Jordan (2003). “Latent Dirichlet Allocation”. In: *Journal of Machine Learning Research* 3, pp. 993–1022.
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov (2017). “Enriching Word Vectors with Subword Information”. In: *Transactions of the Association for Computational Linguistics* 5, pp. 135–146.
- Leo Breiman (2001). “Random Forests”. In: *Machine Learning* 45.1, pp. 5–32.
- John D. Burger, John Henderson, George Kim, and Guido Zarrella (2011). “Discriminating Gender on Twitter”. In: *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pp. 1301–1309.
- Jonathan Chang, Jordan Boyd-Graber, Chong Wang, Sean Gerrish, and David M. Blei (2009). “Reading Tea Leaves: How Humans Interpret Topic Models”. In: *Neural Information Processing Systems*, pp. 288–296.
- Jilin Chen, Eben Haber, Ruogu Kang, Gary Hsieh, and Jalal Mahmud (2015). “Making Use of Derived Personality: The Case of Social Media Ad Targeting”. In: *Proceedings of the Ninth International Conference on Web and Social Media, ICWSM 2015, University of Oxford, Oxford, UK, May 26-29, 2015*, pp. 51–60.
- Lingzhen Chen, Carlo Strapparava, and Vivi Nastase (2017). “Improving Native Language Identification by Using Spelling Errors”. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, pp. 542–546.
- Justin Cheng, Cristian Danescu-Niculescu-Mizil, and Jure Leskovec (2015). “Antisocial Behavior in Online Discussion Communities”. In: *Proceedings of the Ninth International Conference on Web and Social Media, ICWSM 2015*, pp. 61–70.
- Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio (2014). “On the Properties of Neural Machine Translation: Encoder-Decoder Approaches”. In: *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*. Association for Computational Linguistics, pp. 103–111.
- François Chollet (2015). *Keras*. <https://github.com/fchollet/keras>.
- Marisa Chow (2016). “Argument Identification in Chinese Editorials”. In: *Proceedings of the NAACL Student Research Workshop*. Association for Computational Linguistics, pp. 16–21.
- Mark Cieliebak, Jan Milan Deriu, Dominic Egger, and Fatih Uzdilli (2017). “A Twitter Corpus and Benchmark Resources for German Sentiment Analysis”. In: *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*. Association for Computational Linguistics, pp. 45–51.
- Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun (2017). “Very Deep Convolutional Networks for Text Classification”. In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. Association for Computational Linguistics, pp. 1107–1116.
- Michael Conover, Bruno Gonçalves, Jacob Ratkiewicz, Alessandro Flammini, and Filippo Menczer (2011). “Predicting the Political Alignment of Twitter Users”. In: *Proceedings of 3rd IEEE Conference on Social Computing (SocialCom)*. IEEE, pp. 192–199.

- Corinna Cortes and Vladimir Vapnik (1995). “Support-Vector Networks”. In: *Machine Learning* 20.3, pp. 273–297.
- Imke Diefenbach (2013). “Mehr Bürgerbeteiligung? Eine empirische Studie zur Online-plattform Liquid Friesland”. MA thesis. Hochschule Emden/Leer.
- Judith Eckle-Kohler, Roland Kluge, and Iryna Gurevych (2015). “On the Role of Discourse Markers for Discriminating Claims and Premises in Argumentative Discourse”. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015*. Association for Computational Linguistics, pp. 2236–2242.
- Steffen Eger, Johannes Daxenberger, and Iryna Gurevych (2017). “Neural End-to-End Learning for Computational Argumentation Mining”. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, pp. 11–22.
- Johannes Eichstaedt, Hansen Schwartz, Margaret Kern, Gregory Park, Darwin Labarthe, Raina Merchant, Sneha Jha, Megha Agrawal, Lukasz Dziurzynski, Maarten Sap, Christopher Weeg, Emily Larson, Lyle Ungar, and Martin Seligman (2015). “Psychological Language on Twitter Predicts County-Level Heart Disease Mortality”. In: *Psychological Science* 26.2, pp. 159–169.
- Jacob Eisenstein, Brendan O’Connor, Noah Smith, and Eric Xing (2010). “A Latent Variable Model for Geographic Lexical Variation”. In: *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pp. 1277–1287.
- Katharina Esau, Matthias Liebeck, and Christiane Eilders (2017). “Mining Arguments in Online Participation: Möglichkeiten und Grenzen manueller und automatisierter Inhaltsanalyse zur Erhebung von Argumentkomponenten”. In: *Polkomm 2017 - „Disliken, diskutieren, demonstrieren – Politische Partizipation im (Medien-)Wandel“*.
- Tobias Escher, Dennis Friess, Katharina Esau, Jost Sieweke, Ulf Tranow, Simon Dischner, Philipp Hagemester, and Martin Mauve (2017). “Online Deliberation in Academia: Evaluating the Quality and Legitimacy of Co-Operatively Developed University Regulations”. In: *Policy & Internet* 9.1, pp. 133–164.
- Marzieh Fadaee, Arianna Bisazza, and Christof Monz (2017). “Learning Topic-Sensitive Word Representations”. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, pp. 441–447.
- Constanza Fierro, Claudio Fuentes, Jorge Pérez, and Mauricio Quezada (2017). “200K+ Crowdsourced Political Arguments for a New Chilean Constitution”. In: *Proceedings of the 4th Workshop on Argument Mining*. Association for Computational Linguistics, pp. 1–10.
- Marc Franco-Salvador, Nataliia Plotnikova, Neha Pawar, and Yassine Benajiba (2017). “Subword-based Deep Averaging Networks for Author Profiling in Social Media”. In: *CLEF 2017 Evaluation Labs and Workshop – Working Notes Papers*. CEUR Workshop Proceedings. CEUR-WS.org.
- Daniel Fried, Mitchell Stern, and Dan Klein (2017). “Improving Neural Parsing by Disentangling Model Combination and Reranking Effects”. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, pp. 161–166.

- Peter Gladitz, Sabrina Schöttle, Malte Steinbach, Nadja Wilker, and Theresa Witt (2017). “DIID Monitor Online Partizipation - Zum Stand von Online-Bürgerbeteiligung in den Kommunen Nordrhein-Westfalens”. In: *Kommunalpraxis Wahlen 1*, pp. 30–34.
- Lewis Goldberg (1993). “The Structure of Phenotypic Personality Traits”. In: *American Psychologist* 48.1, pp. 26–34.
- Yoav Goldberg (2016). “A Primer on Neural Network Models for Natural Language Processing”. In: *J. Artif. Intell. Res.* 57, pp. 345–420.
- Theodosios Goudas, Christos Louizos, Georgios Petasis, and Vangelis Karkaletsis (2014). “Argument Extraction from News, Blogs, and Social Media”. In: *Artificial Intelligence: Methods and Applications*, pp. 287–299.
- Alex Graves, Santiago Fernández, and Jürgen Schmidhuber (2005). “Bidirectional LSTM Networks for Improved Phoneme Classification and Recognition”. In: *Artificial Neural Networks: Formal Models and Their Applications - ICANN 2005, Proceedings, Part II*, pp. 799–804.
- Alex Graves, Abdel-rahman Mohamed, and Geoffrey E. Hinton (2013). “Speech Recognition with Deep Recurrent Neural Networks”. In: *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013*, pp. 6645–6649.
- Alex Graves and Jürgen Schmidhuber (2005). “Framewise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures”. In: *Neural Networks* 18.5-6, pp. 602–610.
- Nancy Green (2015). “Identifying Argumentation Schemes in Genetics Research Articles”. In: *Proceedings of the 2nd Workshop on Argumentation Mining*. Association for Computational Linguistics, pp. 12–21.
- Thomas. L. Griffiths and Mark Steyvers (2004). “Finding scientific topics”. In: *Proceedings of the National Academy of Sciences* 101.Suppl. 1, pp. 5228–5235.
- Iryna Gurevych (2005). “Using the Structure of a Conceptual Network in Computing Semantic Relatedness”. In: *Proceedings of the Second International Joint Conference on Natural Language Processing*. IJCNLP’05. Springer-Verlag, pp. 767–778.
- Ivan Habernal, Judith Eckle-Kohler, and Iryna Gurevych (2014). “Argumentation Mining on the Web from Information Seeking Perspective”. In: *Proceedings of the Workshop on Frontiers and Connections between Argumentation Theory and Natural Language Processing*. CEUR-WS, pp. 26–39.
- Ivan Habernal and Iryna Gurevych (2015). “Exploiting Debate Portals for Semi-Supervised Argumentation Mining in User-Generated Web Discourse”. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015*. Association for Computational Linguistics, pp. 2127–2137.
- Ivan Habernal and Iryna Gurevych (2016). “Which argument is more convincing? Analyzing and predicting convincingness of Web arguments using bidirectional LSTM”. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, pp. 1589–1599.
- Ivan Habernal and Iryna Gurevych (2017). “Argumentation Mining in User-Generated Web Discourse”. In: *Computational Linguistics* 43.1, pp. 125–179.
- Birgit Hamp and Helmut Feldweg (1997). “GermaNet - a Lexical-Semantic Net for German”. In: *Proceedings of ACL workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*, pp. 9–15.

- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2015). “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification”. In: *International Conference on Computer Vision, ICCV 2015*, pp. 1026–1034.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016). “Deep Residual Learning for Image Recognition”. In: *Conference on Computer Vision and Pattern Recognition, CVPR 2016*, pp. 770–778.
- Pascal Hirmmer, Tim Waizenegger, Ghareeb Falazi, Majd Abdo, Yuliya Volga, Alexander Askinadze, Matthias Liebeck, Stefan Conrad, Tobias Hildebrandt, Conrad Indiono, Stefanie Rinderle-Ma, Martin Grimmer, Matthias Kricke, and Eric Peukert (2017). “The First Data Science Challenge at BTW 2017”. In: *Datenbank-Spektrum* 17.3, pp. 207–222.
- Sepp Hochreiter and Jürgen Schmidhuber (1997). “Long Short-Term Memory”. In: *Neural Comput.* 9.8, pp. 1735–1780.
- Yufang Hou and Charles Jochim (2017). “Argument Relation Classification Using a Joint Inference Model”. In: *Proceedings of the 4th Workshop on Argument Mining*. Association for Computational Linguistics, pp. 60–66.
- Constantin Houy, Tim Niesen, Peter Fettke, and Peter Loos (2013). “Towards Automated Identification and Analysis of Argumentation Structures in the Decision Corpus of the German Federal Constitutional Court”. In: *7th IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST)*. IEEE Computer Society.
- Yuening Hu, Jordan Boyd-Graber, Brianna Satinoff, and Alison Smith (2014). “Interactive Topic Modeling”. In: *Machine Learning* 95, pp. 423–469.
- Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller (2007). *Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments*. Tech. rep. University of Massachusetts, Amherst.
- Manuela Hürlimann, Benno Weck, Esther van den Berg, Simon Suster, and Malvina Nissim (2015). “GLAD: Groningen Lightweight Authorship Detection”. In: *Working Notes of CLEF 2015 - Conference and Labs of the Evaluation forum*.
- ILSVRC2016 (2017). *ILSVRC 2016 results*. URL: <http://image-net.org/challenges/LSVRC/2016/results> (visited on 07/17/2017).
- Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III (2015). “Deep Unordered Composition Rivals Syntactic Methods for Text Classification”. In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, pp. 1681–1691.
- Quentin Jones, Gilad Ravid, and Sheizaf Rafaeli (2004). “Information Overload and the Message Dynamics of Online Interaction Spaces: A Theoretical Model and Empirical Exploration”. In: *Information Systems Research* 15.2, pp. 194–210.
- Roland Kahlert, Matthias Liebeck, and Joseph Cornelius (2017). “Understanding Trending Topics in Twitter”. In: *Datenbanksysteme für Business, Technologie und Web (BTW 2017) - Workshopband*. GI, pp. 375–384.
- Nal Kalchbrenner, Ivo Danihelka, and Alex Graves (2015). “Grid Long Short-Term Memory”. In: *CoRR* abs/1507.01526.
- Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom (2014). “A Convolutional Neural Network for Modelling Sentences”. In: *Proceedings of the 52nd Annual Meet-*

- ing of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, pp. 655–665.
- Jason Kessler (2017). “Scattertext: a Browser-Based Tool for Visualizing how Corpora Differ”. In: *Proceedings of ACL 2017, System Demonstrations*. Association for Computational Linguistics, pp. 85–90.
- Christian Kirschner, Judith Eckle-Kohler, and Iryna Gurevych (2015). “Linking the Thoughts: Analysis of Argumentation Structures in Scientific Publications”. In: *Proceedings of the 2nd Workshop on Argumentation Mining*. Association for Computational Linguistics, pp. 1–11.
- Yuta Koreeda, Toshihiko Yanase, Kohsuke Yanai, Misa Sato, and Yoshiki Niwa (2016). “Neural Attention Model for Classification of Sentences that Support Promoting/-Suppressing Relationship”. In: *Proceedings of the Third Workshop on Argument Mining (ArgMining2016)*. Association for Computational Linguistics, pp. 76–81.
- Tobias Krauthoff, Michael Baurmann, Gregor Betz, and Martin Mauve (2016). “Dialog-Based Online Argumentation”. In: *Computational Models of Argument - Proceedings of COMMA 2016*, pp. 33–40.
- Klaus Krippendorff (2004). *Content Analysis: An Introduction to Its Methodology*. second edition. Sage Publications.
- Solomon Kullback and Richard A. Leibler (1951). “On Information and Sufficiency”. In: *The Annals of Mathematical Statistics* 22.1, pp. 79–86.
- Neeraj Kumar, Alexander C. Berg, Peter N. Belhumeur, and Shree K. Nayar (2009). “Attribute and Simile Classifiers for Face Verification”. In: IEEE Computer Society, pp. 365–372.
- Matt J. Kusner, Yu Sun, Nicholas I. Kolkin, and Kilian Q. Weinberger (2015). “From Word Embeddings To Document Distances”. In: *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015*, pp. 957–966.
- Landesjagdverband Baden-Württemberg e.V. (2014). *Argumente zur Diskussion der Novelle des Landesjagdgesetzes*. URL: http://www.landesjagdverband.de/fileadmin/Medien/LJV/Dokumente/Jagdgesetznovelle/Forderungen/506052_flyer_nj_argumentationshilfe_bw_140403_einzel.pdf (visited on 07/03/2017).
- Jey Han Lau, Karl Grieser, David Newman, and Timothy Baldwin (2011). “Automatic Labelling of Topic Models”. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, pp. 1536–1545.
- Yann LeCun, Bernhard Boser, John Denker, Donnie Henderson, Richard Howard, Wayne Hubbard, and Lawrence Jackel (1989). “Backpropagation Applied to Handwritten Zip Code Recognition”. In: *Neural Computation* 1.4, pp. 541–551.
- Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner (1998). “Gradient-Based Learning Applied to Document Recognition”. In: *Proceedings of the IEEE* 86.11, pp. 2278–2324.
- Wolfgang Lezius, Reinhard Rapp, and Manfred Wettler (1998). “A Freely Available Morphological Analyzer, Disambiguator and Context Sensitive Lemmatizer for German”. In: *Proceedings of the 17th International Conference on Computational Linguistics - Volume 2*. COLING ’98. Association for Computational Linguistics, pp. 743–748.
- Mengxue Li, Shiqiang Geng, Yang Gao, Shuhua Peng, Haijing Liu, and Hao Wang (2017). “Crowdsourcing argumentation structures in Chinese hotel reviews”. In:

- 2017 *IEEE International Conference on Systems, Man, and Cybernetics, SMC 2017*, pp. 87–92.
- Matthias Liebeck (2015a). “Ansätze zur Erkennung von Kommunikationsmodi in Online-Diskussionen”. In: *Proceedings of the 27th GI-Workshop Grundlagen von Datenbanken*, pp. 42–47.
- Matthias Liebeck (2015b). “Aspekte einer automatischen Meinungsbildungsanalyse von Online-Diskussionen”. In: *Datenbanksysteme für Business, Technologie und Web (BTW 2015) - Workshops und Studierendenprogramm*, pp. 203–212.
- Matthias Liebeck and Stefan Conrad (2015). “IWNLP: Inverse Wiktionary for Natural Language Processing”. In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Association for Computational Linguistics, pp. 414–418.
- Matthias Liebeck, Katharina Esau, and Stefan Conrad (2016a). “What to Do with an Airport? Mining Arguments in the German Online Participation Project Tempelhofer Feld”. In: *Proceedings of the Third Workshop on Argument Mining (ArgMining2016)*. Association for Computational Linguistics, pp. 144–153.
- Matthias Liebeck, Katharina Esau, and Stefan Conrad (2017). “Text Mining für Online-Partizipationsverfahren: Die Notwendigkeit einer maschinell unterstützten Auswertung”. In: *HMD Praxis der Wirtschaftsinformatik 54.4. Schwerpunktheft „Online Participation“*, pp. 544–562.
- Matthias Liebeck, Pashutan Modaresi, Alexander Askinadze, and Stefan Conrad (2016b). “Pisco: A Computational Approach to Predict Personality Types from Java Source Code”. In: *Working notes of FIRE 2016 - Forum for Information Retrieval Evaluation*, pp. 43–47.
- Matthias Liebeck, Pashutan Modaresi, and Stefan Conrad (2016c). “Evaluating Safety, Soundness and Sensibleness of Obfuscation Systems”. In: *Working Notes of CLEF 2016 - Conference and Labs of the Evaluation Forum*, pp. 920–928.
- Matthias Liebeck, Philipp Pollack, Pashutan Modaresi, and Stefan Conrad (2016d). “HHU at SemEval-2016 Task 1: Multiple Approaches to Measuring Semantic Textual Similarity”. In: *Proceedings of the 10th International Workshop on Semantic Evaluation*, pp. 595–601.
- Jianhua Lin (1991). “Divergence Measures Based on the Shannon Entropy”. In: *IEEE Transactions on Information Theory* 37.1, pp. 145–151.
- Marco Lippi and Paolo Torroni (2016). “Argumentation Mining: State of the Art and Emerging Trends”. In: *ACM Trans. Internet Technol.* 16.2, 10:1–10:25.
- François Mairesse, Marilyn A. Walker, Matthias R. Mehl, and Roger K. Moore (2007). “Using Linguistic Cues for the Automatic Recognition of Personality in Conversation and Text”. In: *J. Artif. Intell. Res.* 30, pp. 457–500.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean (2013). “Efficient Estimation of Word Representations in Vector Space”. In: *CoRR* abs/1301.3781.
- George A. Miller (1995). “WordNet: A Lexical Database for English”. In: *Communications of the ACM* 38.11, pp. 39–41.
- Yasuhide Miura, Tomoki Taniguchi, Motoki Taniguchi, and Tomoko Ohkuma (2017). “Author Profiling with Word+Character Neural Attention Network”. In: *CLEF 2017 Evaluation Labs and Workshop – Working Notes Papers*. CEUR Workshop Proceedings. CEUR-WS.org.

- Pashutan Modaresi, Matthias Liebeck, and Stefan Conrad (2016a). “Exploring the Effects of Cross-Genre Machine Learning for Author Profiling in PAN 2016”. In: *Working Notes of CLEF 2016 - Conference and Labs of the Evaluation Forum*, pp. 970–977.
- Pashutan Modaresi, Matthias Liebeck, and Stefan Conrad (2016b). “Neural Classification of Linguistic Coherence Using Long Short-Term Memories”. In: *Proceedings of the 8th Annual Meeting of the Forum on Information Retrieval Evaluation*. FIRE ’16. ACM, pp. 28–31.
- Saif M. Mohammad and Peter D. Turney (2013). “Crowdsourcing a Word-Emotion Association Lexicon”. In: 29.3, pp. 436–465.
- Vlad Niculae, Joonsuk Park, and Claire Cardie (2017). “Argument Mining with Structured SVMs and RNNs”. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, pp. 985–995.
- Hiroki Ouchi, Hiroyuki Shindo, and Yuji Matsumoto (2017). “Neural Modeling of Multi-Predicate Interactions for Japanese Predicate Argument Structure Analysis”. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, pp. 1591–1600.
- Raquel Palau and Marie-Francine Moens (2009). “Argumentation Mining: The Detection, Classification and Structure of Arguments in Text”. In: *Proceedings of the 12th International Conference on Artificial Intelligence and Law, ICAIL ’09*. ACM, pp. 98–107.
- Joonsuk Park and Claire Cardie (2014). “Identifying Appropriate Support for Propositions in Online User Comments”. In: *Proceedings of the First Workshop on Argumentation Mining*. Association for Computational Linguistics, pp. 29–38.
- Andreas Peldszus and Manfred Stede (2013). “Ranking the annotators: An agreement study on argumentation structure”. In: *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*. Association for Computational Linguistics, pp. 196–204.
- Andreas Peldszus and Manfred Stede (2016). “An Annotated Corpus of Argumentative Microtexts”. In: *Argumentation and Reasoned Action: Proceedings of the 1st European Conference on Argumentation, Lisbon 2015 / Vol. 2*. College Publications, pp. 801–815.
- Anselmo Peñas and Alvaro Rodrigo (2011). “A Simple Measure to Assess Non-response”. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 1415–1424.
- Jeffrey Pennington, Richard Socher, and Christopher Manning (2014). “Glove: Global Vectors for Word Representation”. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, pp. 1532–1543.
- Slav Petrov, Dipanjan Das, and Ryan McDonald (2012). “A Universal Part-of-Speech Tagset”. In: *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*. European Language Resources Association (ELRA), pp. 2089–2096.

- Martin Potthast, Matthias Hagen, and Benno Stein (2016). “Author Obfuscation: Attacking the State of the Art in Authorship Verification”. In: *Working Notes of CLEF 2016 - Conference and Labs of the Evaluation Forum*, pp. 716–749.
- Daniel Preotiuc-Pietro, Vasileios Lampos, and Nikolaos Aletras (2015). “An analysis of the user occupational class through Twitter content”. In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, pp. 1754–1764.
- Pavithra Rajendran, Danushka Bollegala, and Simon Parsons (2016). “Contextual stance classification of opinions: A step towards enthymeme reconstruction in online reviews”. In: *Proceedings of the Third Workshop on Argument Mining (ArgMining2016)*. Association for Computational Linguistics, pp. 31–39.
- Francisco Rangel, Fabio Celli, Paolo Rosso, Martin Potthast, Benno Stein, and Walter Daelemans (2015). “Overview of the 3rd Author Profiling Task at PAN 2015”. In: *Working Notes Papers of the CLEF 2015 Evaluation Labs*. CLEF.
- Francisco Rangel, Paolo Rosso, Irina Chugur, Martin Potthast, Martin Trenkmann, Benno Stein, Ben Verhoeven, and Walter Daelemans (2014). “Overview of the 2nd Author Profiling Task at PAN 2014”. In: *Working Notes for CLEF 2014 Conference*. CLEF, pp. 898–927.
- Francisco Rangel, Paolo Rosso, Moshe Koppel, Efstathios Stamatatos, and Giacomo Inches (2013). “Overview of the Author Profiling Task at PAN 2013”. In: *Working Notes for CLEF 2013 Conference*. CLEF.
- Francisco Rangel, Paolo Rosso, Martin Potthast, and Benno Stein (2017). “Overview of the 5th Author Profiling Task at PAN 2017: Gender and Language Variety Identification in Twitter”. In: *CLEF 2017 Evaluation Labs and Workshop – Working Notes Papers*. CEUR Workshop Proceedings. CEUR-WS.org.
- Francisco Rangel, Paolo Rosso, Ben Verhoeven, Walter Daelemans, Martin Potthast, and Benno Stein (2016). “Overview of the 4th Author Profiling Task at PAN 2016: Cross-genre Evaluations”. In: *Working Notes Papers of the CLEF 2016 Evaluation Labs*. CEUR Workshop Proceedings. CLEF and CEUR-WS.org.
- Delip Rao, David Yarowsky, Abhishek Shreevats, and Manaswi Gupta (2010). “Classifying Latent User Attributes in Twitter”. In: *Proceedings of the 2Nd International Workshop on Search and Mining User-generated Contents*. SMUC ’10. ACM, pp. 37–44.
- Lev Ratinov and Dan Roth (2009). “Design Challenges and Misconceptions in Named Entity Recognition”. In: *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*. Association for Computational Linguistics, pp. 147–155.
- Radim Rehurek and Petr Sojka (2010). “Software Framework for Topic Modelling with Large Corpora”. In: *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. ELRA, pp. 45–50.
- Paul Reisert, Junta Mizuno, Miwa Kanno, Naoaki Okazaki, and Kentaro Inui (2014). “A Corpus Study for Identifying Evidence on Microblogs”. In: *Proceedings of LAW VIII - The 8th Linguistic Annotation Workshop*. Association for Computational Linguistics, pp. 70–74.
- Robert Remus, Uwe Quasthoff, and Gerhard Heyer (2010). “SentiWS - A Publicly Available German-language Resource for Sentiment Analysis”. In: *Proceedings of the Sev-*

- enth conference on International Language Resources and Evaluation (LREC'10)*. European Language Resources Association (ELRA).
- Sebastian Rohmann and Matthias Schumann (2017). “Best Practices für die Mitarbeiter-Partizipation in der Produktentwicklung”. In: *HMD Praxis der Wirtschaftsinformatik* 54.4. Schwerpunktheft „Online Participation“, pp. 575–590.
- RusProfiling (2017). *RusProfiling 2017*. URL: <http://en.rusprofilinglab.ru/rusprofiling-at-pan/> (visited on 08/09/2017).
- Anne Schiller, Simone Teufel, Christine Stöckert, and Christine Thielen (1999). *Guidelines für das Tagging deutscher Textcorpora mit STTS (kleines und großes Tagset)*. Tech. rep. Universität Stuttgart, Universität Tübingen.
- Helmut Schmid (1994). “Probabilistic Part-of-Speech Tagging Using Decision Trees”. In: *Proceedings of the International Conference on New Methods in Language Processing*.
- Jodi Schneider and Adam Z. Wyner (2012). “Identifying Consumers’ Arguments in Text”. In: *Proceedings of the Workshop on Semantic Web and Information Extraction (SWAIE 2012)*, pp. 31–42.
- Alexandra Schofield, Måns Magnusson, and David Mimno (2017). “Pulling Out the Stops: Rethinking Stopword Removal for Topic Models”. In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Association for Computational Linguistics, pp. 432–436.
- Florian Schroff, Dmitry Kalenichenko, and James Philbin (2015). “FaceNet: A Unified Embedding for Face Recognition and Clustering”. In: *CoRR* abs/1503.03832.
- Mike Schuster and Kuldeep Paliwal (1997). “Bidirectional Recurrent Neural Networks”. In: *IEEE Trans. Signal Processing* 45.11, pp. 2673–2681.
- H. Andrew Schwartz, Johannes Eichstaedt, Margaret Kern, Gregory Park, Maarten Sap, David Stillwell, Michal Kosinski, and Lyle Ungar (2014). “Towards Assessing Changes in Degree of Depression through Facebook”. In: *Proceedings of the Workshop on Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality*. Association for Computational Linguistics, pp. 118–125.
- H. Andrew Schwartz, Salvatore Giorgi, Maarten Sap, Patrick Crutchley, Lyle Ungar, and Johannes Eichstaedt (2017). “DLATK: Differential Language Analysis ToolKit”. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, pp. 55–60.
- Uladzimir Sidarenka (2016). “PotTS: The Potsdam Twitter Sentiment Corpus”. In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. European Language Resources Association (ELRA), pp. 1133–1141.
- Alison Smith, Tak Lee, Forough Poursabzi-Sangdeh, Jordan Boyd-Graber, Niklas Elmqvist, and Leah Findlater (2017). “Evaluating Visual Representations for Topic Understanding and Their Effects on Manually Generated Labels”. In: *Transactions of the Association for Computational Linguistics* 5, pp. 1–15.
- Christian Stab and Iryna Gurevych (2014a). “Annotating Argument Components and Relations in Persuasive Essays”. In: *Proceedings of the 25th International Conference on Computational Linguistics (COLING 2014)*, pp. 1501–1510.

- Christian Stab and Iryna Gurevych (2014b). “Identifying Argumentative Discourse Structures in Persuasive Essays”. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*. Association for Computational Linguistics, pp. 46–56.
- Christian Stab and Iryna Gurevych (2017). “Recognizing Insufficiently Supported Arguments in Argumentative Essays”. In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. Association for Computational Linguistics, pp. 980–990.
- Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun’ichi Tsujii (2012). “BRAT: a Web-based Tool for NLP-Assisted Text Annotation”. In: *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*. EACL ’12. Association for Computational Linguistics, pp. 102–107.
- Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi (2017). “Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning”. In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pp. 4278–4284.
- Theano Development Team (2016). “Theano: A Python framework for fast computation of mathematical expressions”. In: *CoRR* abs/1605.02688.
- Stephen Toulmin (1958). *The Uses of Argument*. Cambridge University Press.
- Stephen Toulmin (2003). *The Uses of Argument, Updated Edition*. Cambridge University Press.
- Lyle Ungar (2017). “Measuring Psychological Traits using Social Media”. Invited Talk @ NLP+CSS: Second Workshop on Natural Language Processing and Computational Social Science, 04/08/2017, Vancouver.
- Vincent Vanhoucke (2017). *Udacity course “Deep Learning”*. <https://de.udacity.com/course/deep-learning--ud730>. (Visited on 07/26/2017).
- Ben Verhoeven, Walter Daelemans, and Barbara Plank (2016). “TwiSty: A Multilingual Twitter Stylometry Corpus for Gender and Personality Profiling”. In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016*. URL: <http://www.lrec-conf.org/proceedings/lrec2016/summaries/644.html>.
- Peilu Wang, Yao Qian, Frank K. Soong, Lei He, and Hai Zhao (2015). “Part-of-Speech Tagging with Bidirectional Long Short-Term Memory Recurrent Neural Network”. In: *CoRR* abs/1510.06168.
- Xingyou Wang, Weijie Jiang, and Zhiyong Luo (2016). “Combination of Convolutional and Recurrent Neural Network for Sentiment Analysis of Short Texts”. In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 2428–2437.
- Wayne Xiong, Jasha Droppo, Xuedong Huang, Frank Seide, Mike Seltzer, Andreas Stolcke, Dong Yu, and Geoffrey Zweig (2016). “Achieving Human Parity in Conversational Speech Recognition”. In: *CoRR* abs/1610.05256.
- Omar F. Zaidan and Chris Callison-Burch (2011). “Crowdsourcing Translation: Professional Quality from Non-Professionals”. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, pp. 1220–1229.

- Robert Zepic, Marcus Dapp, and Helmut Krcmar (2017). “E-Partizipation und keiner macht mit”. In: *HMD Praxis der Wirtschaftsinformatik* 54.4. Schwerpunktheft „Online Participation“, pp. 488–501.
- Torsten Zesch and Iryna Gurevych (2006). “Automatically Creating Datasets for Measures of Semantic Relatedness”. In: *Proceedings of the Workshop on Linguistic Distances*. Association for Computational Linguistics, pp. 16–24.
- Xiang Zhang, Junbo Zhao, and Yann LeCun (2015). “Character-level Convolutional Networks for Text Classification”. In: *Advances in Neural Information Processing Systems 28*. Curran Associates, Inc., pp. 649–657.
- Marc Ziegele and Pablo B. Jost (2016). “Not Funny? The Effects of Factual Versus Sarcastic Journalistic Responses to Uncivil User Comments”. In: *Communication Research*.

LIST OF FIGURES

| | | |
|-----|---|----|
| 2.1 | Our argumentation model for political online participation | 12 |
| 2.2 | Examples of our argumentation model applied to excerpts of the Tempelhofer Feld | 13 |
| 2.3 | Multi-level classification process for an input sentence | 15 |
| 2.4 | Empirical curve of training size vs. macro-averaged F_1 score | 20 |
| 2.5 | Joint architecture of the benchmarked neural networks | 28 |
| 3.1 | Examples of two Wiktionary entries and their rendered HTML outputs | 35 |
| 3.2 | Comparison of the corpora's sentence lengths | 45 |
| 3.3 | Combination of topic modeling and topic labeling | 50 |

LIST OF TABLES

| | | |
|-----|---|----|
| 2.1 | Class distribution for all three subtasks in the THF Airport ArgMining Corpus | 15 |
| 2.2 | Macro-averaged F_1 scores for subtask A and subtask B in Liebeck et al. (2016a) | 18 |
| 2.3 | Macro-averaged F_1 scores for subtask A and subtask B in Liebeck et al. (2017) | 20 |
| 2.4 | Macro-averaged F_1 scores of subtask A and B for different character embedding n-gram sizes | 21 |
| 2.5 | Macro-averaged F_1 scores for further evaluations of subtask A and subtask B | 23 |
| 2.6 | Macro-averaged F_1 scores for subtask C | 24 |
| 2.7 | Macro-averaged F_1 scores for the benchmarked deep learning architectures | 29 |
| 3.1 | Percentages of correctly lemmatized words with IWNLP as a standalone lemmatizer | 36 |
| 3.2 | Example results of the Overlap method compared to the gold standard (Liebeck et al., 2016d) | 38 |
| 3.3 | Labels for the sensibleness dimension | 42 |
| 3.4 | Number of proposals and comments per corpus | 45 |
| 3.5 | Excerpt from the topics identified by LDA on the Tempelhofer Feld project and their corresponding five most frequent words (Liebeck et al., 2017) | 46 |
| 3.6 | LDA Topics for Braunkohle | 47 |
| 3.7 | LDA Topics for Bonn 2011 | 48 |
| 3.8 | Excerpt from the LDA topics from Bonn 2015 with $K = 14$ | 49 |
| 3.9 | Example topics from Bhatia et al. (2016) | 51 |
| 4.1 | Evaluation results of our author profiling approach on the <i>Braunkohle</i> corpus | 58 |