Informatics

Analyse politischer Meinungen und Erkennung von Stilfiguren

Klassifizierung basierend auf Themen und Meinungstypen im politischen Kontext; Erkennung von Alliterationen und Hyperbeln

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Data Science

eingereicht von

Christopher Deringer, BSc
Matrikelnummer 01529026

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Ao.Univ.Prof. Ing. Mag. Dr. Horst Eidenberger

Wien, 29. Februar 2024

Christopher Deringer

Horst Eidenberger

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Informatics

Political Opinion Analysis and Figure of Speech detection

Topic and opinion type classification in the political context; alliteration and hyperbole detection

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Data Science

by

Christopher Deringer, BSc
Registration Number 01529026

to the Faculty of Informatics

at the TU Wien

Advisor: Ao.Univ.Prof. Ing. Mag. Dr. Horst Eidenberger

Vienna, 29th February, 2024

Christopher Deringer

Horst Eidenberger

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

TU WIEN Bibliothek Your knowledge hub

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

TU WIEN Bibliothek Your knowledge hub

Erklärung zur Verfassung der Arbeit

Christopher Deringer, BSc

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 29. Februar 2024

Christopher Deringer

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Danksagung

Diese Stelle möchte ich gerne nutzen um mich bei den Personen zu bedanken, die es mir ermöglicht haben, an diesen Punkt zu kommen.

Ich möchte meine tiefste Dankbarkeit gegenüber meinem Betreuer Ao.Univ.Prof. Ing. Mag. Dr. Horst Eidenberger ausdrücken, der mir während dieser Arbeit kontinuierliche Unterstützung zukommen ließ und immer sehr zeitnah geantwortet hat wenn ich eine Frage hatte. Ich bin sehr dankbar für das konstruktive Feedback das ich erhalten habe und für die Freiheit die mir gewährt wurde, ohne ihn wäre diese Diplomarbeit nicht realisiert worden.

Ein besonderer Dank gilt auch meiner Familie und ganz besonders meinen Eltern, die mich während meines Studiums fortwährend unterstützt haben und immer ein offenes Ohr für mich hatten. Danke, dass ihr mir diese Möglichkeit gegeben habt und stets hinter mir gestanden seid.

Abschließend möchte ich mich bei meiner Partnerin bedanken, die stets eine Quelle der Motivation für mich war und es immer schaffte mich aufzuheitern. Danke, dass du immer für mich da bist.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Acknowledgements

Now is the moment to express my appreciation towards the people who made it possible for me to come to this point.

I would like to express my deepest gratitude to my supervisor Ao.Univ.Prof. Ing. Mag. Dr. Horst Eidenberger who supported me throughout this thesis and always responded as quickly as possible. I am very thankful for the constructive feedback that I received and the freedom I was granted, it would not have been possible to write this thesis without him.

I would like to extend my gratefulness to my family and especially my parents who always supported me during my studies and have been there for me. Thank you for giving me this opportunity and for having my back during all this time.

I wish to expand my thankfulness to my partner who was able to motivate me and cheer me up on all occasions. Thank you for always being there for me and for the constant support.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Kurzfassung

Im Rahmen dieser Arbeit wurden vier verschiedene Probleme im Kontext von deutschsprachigen politischen Aussagen und der deutschen Sprache im Allgemeinen behandelt. Diese sind die Klassifizierung von Themen und Meinungstypen und die Erkennung von Alliterationen und Hyperbeln. Die meisten Experimente wurden mit einem Datensatz durchgeführt, der basierend auf den Protokollen des österreichischen Nationalrates erstellt wurde und rund 65.000 politische Aussagen enthält.

Der gewählte Ansatz für die Themenerkennung baut auf der Extrahierung relevanter Begriffe aus dem Wikipedia Artikel zu einem Thema auf. Die Resultate wurden händisch evaluiert, die Genauigkeit war im Falle von den Themen "Feminismus"(36,39%) und "Flüchtlingskrise in Europa"(19,04%) sehr niedrig, im Fall von dem Thema "Klimawandel"(89,02%) jedoch gut.

Der Ansatz, der für die Klassifizierung von Meinungstypen verfolgt wurde, stammt von Othman et al. [Using NLP Approach for Opinion Types Classifier, Othman et al., 2015] und wurde für die englische Sprache entwickelt. Das Ziel der Experimente war es herauszufinden, ob der Ansatz auch für die deutsche Sprache funktioniert. Die Evaluierung zeigte, dass die Genauigkeit im Falle von Meinungen im Positiv vergleichbar ist (76,60% vs 71,00%), das war jedoch nicht der Fall für Meinungen, die den Komparative (78,30% vs 44,00%) oder Superlativ (82,10% vs 44,00%) verwenden.

Die Erkennung von Alliterationen war erfolgreich bei der Verwendung eines Datensatzes der 605 Alliterationen enthält, eine Genauigkeit von 99,33% wurde erreicht. Es wurden noch drei zusätzliche Experimente auf freien Text durchgeführt, hier wurde eine durchschnittliche Genauigkeit von 53,83% erreicht, das Minimum war 30,00%. Der Ansatz verwendet den Kölner Phonetik Algorithmus der von Postel [Die Kölner Phonetik - Ein Verfahren zur Identifizierung von Personennamen auf der Grundlage der Gestaltanalyse, Postel, 196] entwickelt wurde und kombiniert ihn mit zusätzlichen Regeln.

Für die Erkennung der Hyperbeln wurde ein existierender Ansatz für die englische Sprache, der von Troiano et al. [A computational exploration of exaggeration, Troiano et al., 2018] entwickelt wurde und auf semantischen Eigenschaften basiert, für die deutsche Sprache implementiert. An das Problem wurde mit überwachtem Lernen herangegangen, es wurde als ein binäres Klassifikationsproblem definiert. Die Resultate wurden im Hinblick auf Genauigkeit (76,00% vs 52,23%), Trefferquote (76,00% vs 38,52%), Treffergenauigkeit (72,00% vs 68,90%) und F1-Score (76,00% vs 41,11%) verglichen.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Abstract

In the scope of this work, four different problems have been studied in the context of German political statements and the German language in general, namely topic classification, opinion type classification, alliteration detection, and hyperbole detection. Most of the experiments were conducted using a dataset that was created based on protocols of the Austrian national council containing around 65000 political statements.

The topic classification was performed by extracting topic related terms from the Wikipedia article on a certain topic. It was manually evaluated and led to results that leave room for improvement, as the precision regarding the topics feminism (36.39%) and European migrant crisis (19.04%) showed. In the case of climate change, a precision of 89.02% was achieved.

The approach that was implemented for opinion type classification is based on part-of-speech tagging and was proposed and implemented for the English language by Othman et al. [Using NLP Approach for Opinion Types Classifier, Othman et al., 2015]. The goal of the experiments was to show whether the approach is applicable to the German language as well when using a part-of-speech tagger for the German language and the respective tags. The evaluation showed that the performance of this approach is comparable in terms of precision in the case of the opinionated statements (76.60% vs 71.00%). It was not the case for comparative (78.30% vs 44.00%) and superlative opinionated statements (82.10% vs 44.00%).

In the case of alliteration detection, a precision of 99.33% was achieved on an alliteration dataset containing 605 alliterations. Three additional experiments were performed on free text, where an average precision of 53.83% was achieved, with 30.00% being the worst case. The approach utilizes the Cologne Phonetics algorithm by Postel [Die Kölner Phonetik - Ein Verfahren zur Identifizierung von Personennamen auf der Grundlage der Gestaltanalyse, Postel, 1969] and combines it with additional rules.

For hyperbole detection, an existing approach for the English language by Troiano et al. [A computational exploration of exaggeration, Troiano et al., 2018] based on the computation of semantic features has been implemented for the German language. It was defined as a supervised machine learning problem; a binary classification task. The results were compared in terms of precision (76.00% vs 52.23%), recall (76.00% vs 38.52%), accuracy (72.00% vs 68.90%) and F1-score (76.00% vs 41.11%). The performance was only comparable in terms of accuracy.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Contents

Kurzfassung **xi**

Abstract **xiii**

Contents **xv**

1 Introduction **1**
- 1.1 Motivation 1
- 1.2 Problem Statement 2
- 1.3 Research Questions 3
- 1.4 Outline 4

2 Background **7**
- 2.1 Machine Learning 7
- 2.2 Natural Language Processing (NLP) 16
- 2.3 Related Work 26

3 Design **31**
- 3.1 Research Questions 31
- 3.2 Methodological Approach 32
- 3.3 Experiment Design 33
- 3.4 Datasets 35

4 Experiments **39**
- 4.1 Extracting technical terms from Wikipedia 39
- 4.2 Opinion Type Classification 44
- 4.3 Alliteration Detection 46
- 4.4 Hyperbole Detection 55

5 Evaluation **61**
- 5.1 Extracting technical terms from Wikipedia 61
- 5.2 Opinion Type Classification 70
- 5.3 Alliteration Detection 80
- 5.4 Hyperbole Detection 87

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

6 Conclusion **93**

List of Figures **97**

List of Tables **99**

List of Algorithms **101**

Bibliography **103**

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

CHAPTER 1

Introduction

1.1 Motivation

In recent years, the landscape of political discourse has undergone a profound transformation, propelled by the unprecedented surge of information and the growing influence of digital platforms. This dynamic environment has given rise to critical challenges, notably the erosion of public trust in government institutions and the pervasive spread of misinformation and disinformation. As we navigate this complex terrain, there is an urgent need for robust tools and frameworks that can dissect political speech, providing valuable insights into sentiments, opinions, and the underlying narratives shaping our socio-political landscape.

The OECD Trust Survey [OEC], which was conducted in 2021 for the first time, is a cross-national survey which aims to measure the trust in government and public institutions. In the first edition, 22 countries including Austria participated. In the second edition, 30 countries participate. However, the second edition of the survey does not include Austria. The survey collected opinions on a range of questions, some of them relating to the responsiveness of governments to public feedback.

In Austria, 41,36% of the participants said that they think it is unlikely that a public service would be improved in response to public feedback, 37,23% said it is likely, and 18,84% were neutral on the issue, while the remaining 2,57% said they do not know. The OECD report presented the results with the headline "Governments seen by many as unresponsive to public feedback", which might be true in the case of Austria. To work on an issue like this, it is worthwhile to investigate which topics are being discussed in the national council. To solve this problem, I suggest an approach utilizing topic-based information retrieval.

The proliferation of misinformation and disinformation further exacerbates the challenges faced by citizens seeking to make informed decisions. By developing a comprehensive framework for political sentiment analysis, I aim to implement an approach to effectively discern between factual information and opinions of varying effects. Two linguistic elements that hold particular significance in this realm are hyperboles and alliteration, and this thesis also aims to implement a method for reliably detecting them. Hyperboles, characterized by exaggerated statements or claims not meant to be taken literally, are frequently employed by political figures to emphasize a point, elicit an emotional response, or underscore the urgency of an issue. The detection of hyperbolic expressions is crucial as it unveils the rhetorical strategies at play, enabling a more nuanced understanding of the speaker's intent and the potential impact on public sentiment.

Simultaneously, alliteration, the repetition of consonant sounds at the beginning of adjacent or closely connected words, serves as a rhetorical device that enhances the rhythm and resonance of language. In political speech, alliteration can be strategically utilized to amplify key messages, create memorable slogans, and reinforce a sense of unity or urgency. By incorporating the detection of alliterative patterns into my framework, I aim to unveil the stylistic choices made by politicians to craft persuasive narratives.

This thesis draws inspiration from the protocols of the Austrian National Council, recognizing the importance of tailoring natural language processing techniques to the specific nuances of political discourse in the Austrian context.

The intersection between political opinion analysis and figure of speech detection holds the potential to strengthen the understanding of political communication in the sense that rhetoric details are highlighted, which gives interesting insights into how politicians are expressing their ideas and views. Alliterations can be used to draw attention to certain parts of a statement, hyperboles might be used to evoke an emotional response. Both of these rhetorical instruments could be used to influence how a certain issue is perceived and how opinions are formed.

One example is the following sentence: "Statt Mut impft ihr den Leuten Angst ein". This sentence is a quotation of representative Wolfgang Zanger, the statement was included in his speech in the 58. sitting of the national council [nat]. It can be interpreted as stating that the addressed politicians try to scare the public instead of giving them hope. The interesting part in this context is the word "impft". The word "impft" is the inflected form of "impfen", "impfen" stands for "to vaccinate" and could potentially lead to emotional responses as the topic of vaccination played an important role during the COVID-19 pandemic. This type of analysis can lead to a deeper understanding of political speeches.

1.2 Problem Statement

This experimental study aims to propose further approaches for recognizing opinions and the way they are expressed in written text with the help of various natural language processing techniques. During an election a lot of different politicians and parties are

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

trying to win the favor of the voters. One important factor for the individual voting decisions might be the opinions which are communicated by the politicians whom are associated with a certain party. It is hard to be aware of everything that has been said by members of a certain party, generally it is hard to identify whether the individual politicians of a party are pulling in the same direction.

One subproblem of this larger problem is opinion type classification, where one wants to distinguish between non-opinionated statements, opinions, comparative opinions and superlative opinions, where the opinions are expressed with either the positive, comparative or superlative form.

Figures of speech are deviating from ordinary language and have the goal to produce a certain rhetorical effect. Two popular figures of speech are hyperbole and alliteration. It is known that these figures of speech are sometimes used in marketing as an attempt to win the attention of a potential customer. The hyperbole is used in the political context as well. An automatic detection of these figures of speech would make it easier to detect their usage and pave the way for a better understanding.

The paragraphs prior to this one state the two problems which will be the main focus of this thesis. The first problem is the opinion type classification, the second problem is alliteration and hyperbole detection.

In addition a rule-based approach is going to be implemented which aims at providing the capabilities to search for political statements based on their topic.

1.3 Research Questions

The following four research questions were defined to tackle the problems that were described in the problem statement:

RQ1: *How high is the precision of a rule-based approach for topic classification in political statements using corpora extracted from Wikipedia?*

This research question is answered by implementing a simple form of information retrieval, an inverted index is created which contains political statements from the Austrian national council protocols. The idea is to extract topic-related terms from a Wikipedia article, which are then used to search for statements containing the topic-related terms. The precision is calculated based on the amount of relevant and irrelevant statements.

RQ2: *How does an approach using a German tag set and a tagger for the German language, instead of an English tag set and a tagger for the English language, for opinion type classification in German sentences, hold up against the approach developed by Othman et al. regarding opinion type classification in English sentences? This question will be evaluated using precision*

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Othman et al. [OHMI15] defined four different classes, namely opinionated, comparative opinionated, superlative opinionated and non-opinionated. The different classes are assigned based on certain part of speech tags (POS-tags). This research question is answered by implementing the same approach for the German language, the precision is calculated based on whether the statement received the correct class from the implemented algorithm.

RQ3: *The Cologne phonetics algorithm can be used to transform a word into a numerical representation depicting the underlying phonetics. Which additional constraints need to be added so that the numerical representation resulting from the Cologne phonetics algorithm is feasible for alliteration detection with 95 percent precision?*

The third research question is answered by implementing an algorithm for alliteration detection which uses the Cologne phonetics algorithm by Postel [Pos69] and additional constraints which are worked out by performing experiments on an alliteration dataset containing 605 alliterations. The precision is calculated based on whether the alliterations are detected or not.

RQ4: *Troiano et al. developed an approach for classifying English sentences as either hyperbolic or not. Is this approach applicable to the German language as well? This question will be evaluated using accuracy, precision, recall and the F1-Score.*

Troiano et al. [TSÖT18] developed an approach for hyperbole detection for the English language, which uses semantic features that are computed by using pre-trained models or specific datasets. The goal of this question is to implement the same approach for the German language. It requires research regarding datasets and models which are suitable for calculating the same semantic features for German statements. It is a classification task with two classes, multiple supervised machine learning algorithms are used and the metrics accuracy, precision, recall and F1-score are used to evaluate the results, which are then compared to the best results achieved by Troiano et al. [TSÖT18].

1.4 Outline

This section gives an overview regarding the remaining chapters. Chapter 2 is the literature chapter of the thesis and contains sections on machine learning 2.1, natural language processing 2.2 and the related work 2.3. The machine learning section 2.1 contains subsections on supervised learning 2.1.1 and deep learning 2.1.2. The supervised learning subsection 2.1.1 is further split into by different topics, namely regression 2.1.1.1, classification 2.1.1.2 and transformers 2.1.2.1.

The section on natural language processing contains subsections on tokenization 2.2.1, phonetic algorithms 2.2.2, part-of-speech tagging 2.2.3, sentiment analysis 2.2.4 and BERT 2.2.5.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

The related work can be found in section 2.3, this section has a subsection on alliterations and hyperboles in marketing an politics (2.3.1), a subsection on hyperbole detection (2.3.2) and a subsection on opinion type classification (2.3.3).

In the third chapter (3) the research questions are defined and the methodological approach and experiment design are described. The first section (3.1) shows the requirements that need to be fulfilled so that each research question can be answered, section 3.2 shows the scientific methodology that is followed throughout the thesis and 3.3 describes how the experiments are designed. Section 3.4 provides information regarding the datasets that were created in the scope of this thesis.

The fourth chapter (4) describes the experiments that were conducted in the scope of this thesis. Section 4.1 contains the description for the Wikipedia-based topic classification approach that was implemented to find political statements based on the topic. It describes the overall process that the algorithm follows and the datasets that were used

The second section (4.2) of chapter 4 describes the opinion type classification approach that was implemented by Othman et al. [OHMI15] and shows how it can be applied to the German language, by using a part-of-speech tagger and a tagset for the German language. In addition it describes a rule-based approach and a dictionary-based approach, which was implemented in addition so that the results can be compared to the results achieved by using the approach which is based on the part-of-speech tags. Section 4.3 describes the algorithm that was implemented for alliteration detection and section 4.4 describes how the hyperbole detection approach by Troiano et al. [TSÖT18] was re-implemented for the German language.

Chapter 5 contains the evaluation of all the experiments that were conducted in the scope of this thesis. The sections in the chapter are again separated by topic. Section 5.1 shows the evaluation of the experiments regarding the Wikipedia-based approach for topic classification, section 5.2 shows the evaluation of the three different approaches that were implemented for opinion type classification. The third section (5.3) of chapter 5 shows the results that were achieved using the alliteration detection algorithm and section 5.4 shows the results of the hyperbole detection algorithm for the German language.

The final chapter (6) summarizes the thesis by giving answers to the research questions and by making suggestions for future work.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

CHAPTER 2

Background

2.1 Machine Learning

2.1.1 Supervised Learning

In machine learning, a supervised learning problem is a problem that is tackled by providing a dataset that already contains input and output variables. According to the book "The Elements of Statistical Learning" by Hastie, Tibshirani and Friedman [HTFF09] it is called "supervised" as there already is an outcome variable available to guide the learning process.

One can distinguish between two types of supervised learning tasks, one being regression and the other one being classification. The distinction is made based on the output [HTFF09], where regression is used in tasks with quantitative output and classification is used in tasks with qualitative output. Both tasks can be seen as a task in function approximation.

2.1.1.1 Regression

Regression is one of the two types of approaches that are available in supervised learning. It is used if the input variables of a dataset are used to compute a quantitative output variable.

One popular teaching example for a regression task uses a dataset [Wic19] about diamonds, which contains the four C's of diamond quality, being carat, cut, color and clarity and five physical measurements.

The dataset was created by Hadley Wickham [Wic19] based on data from a diamond search engine and it contains more than 50000 entries. The dataset is included in the ggplot2 R package and it is described in the book "ggplot2: Elegant Graphics for Data Analysis" by Hadley Wickham [WW16].

The output variable is the price of a diamond, the goal is to train a model that can successfully calculate the price of a diamond based on the given input variables or a subset of them.

Linear Regression The linear regression model, as shown in Hastie et al. [HTFF09], has the form

$$f(X) = \beta_0 + \sum_{j=1}^{p} X_j \beta_j \tag{2.1}$$

with an input vector $X^T = (X_1, X_2, ..., X_n)$ and an output Y.

The assumption which is made by the model is that the regression function $E(Y|X)$ is linear or that a linear model can be seen as an approximation. The idea is to estimate the coefficients β by using the available training data.

The most popular method for estimating β is least squares, where the coefficients $\beta = (\beta_0, \beta_1, ..., \beta_p)^T$ are picked with the intention of minimizing the residual sum of squares (RSS) as described by Filzmoser [Fil20]:

$$\begin{aligned} RSS(\beta) &= \sum_{i=1}^{N} (y_i - f(x_i))^2 \\ RSS(\beta) &= \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij} \beta_j)^2 \end{aligned} \tag{2.2}$$

2.1.1.2 Classification

Classification is the second type of approach that is available in supervised learning, next to regression. It is used if the input variables of a dataset are used to compute a qualitative output variable.

A popular example for a classification task, which is described in Hastie et al. [HTFF09], is an email spam filter which is used to decide whether an email is spam or a normal email that a person actually wants to read. There are two qualitative categories, being $C = \{spam, email\}$ and the assumption is that an input email can either be one or the other.

Naive Bayes Naive Bayes classifiers all share the property that they assume that the value of an input variable is independent of the value of any other input variable in regard

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

of the output variable. Li and Jain [LJ98] explain it in the context of a text classification system where $C = (c_1, ..., c_m)$ are the m document classes. An unlabeled document, or input text, D is transformed into a list of words $W = (w_1, ..., w_d)$, which is then used in the naive Bayes approach to assign a class to document D as follows:

$$c^*_{NB} = argmax_{c_j \in C} P(c_j) \prod_{i=1}^{d} P(w_i|c_j) \tag{2.3}$$

with $P(c_j)$ being the a priori probability of class c_j and $P(w_i|c_j)$ being the conditional probability of word w_i given class c_j. In this case the underlying assumption is that each word in the document occurs independent of every other word in the document.

This approach is yet quite simple and does not perform very well with small training sets, as the relative frequency estimate of a word will be zero if it does not appear in any document in the training set [LJ98]. Li and Jain [LJ98] applied Laplace law of succession, which is explained by Ristad [Ris95] in "A Natural Law of Succession", to tackle this problem.

Logistic Regression Logistic Regression is a model where the probability of certain outputs is computed based on the input which can be used to build a classifier.

Hastie et al. [HTFF09] explain it as a model that was created for the purpose of modeling posterior probabilities of multiple classes by using linear functions under the consideration that the posterior probabilities sum to one and are in the interval $[0, 1]$.

The logistic regression model has the following form [HTFF09]:

$$\begin{aligned} log\frac{Pr(G=1|X=x)}{Pr(G=K|X=x)} &= \beta_{10} + \beta_1^T x \\ log\frac{Pr(G=2|X=x)}{Pr(G=K|X=x)} &= \beta_{20} + \beta_2^T x \\ &\vdots \\ log\frac{Pr(G=K-1|X=x)}{Pr(G=K|X=x)} &= \beta_{(K-1)0} + \beta_{K-1}^T x \end{aligned} \tag{2.4}$$

where $Pr(G = k|X = x)$ and $Pr(G = K|X = x)$ are defined as follows:

$$Pr(G=k|X=x) = \frac{exp(\beta_{k0} + \beta_k^T x)}{1 + \sum_{i=1}^{K-1} exp(\beta_{i0} + \beta_i^T x)},\ k = 1, ..., K-1 \tag{2.5}$$

$$Pr(G=K|X=x) = \frac{1}{1 + \sum_{i=1}^{K-1} exp(\beta_{i0} + \beta_i^T x)}. \tag{2.6}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Logistic regression models are most of the time fit by maximum likelihood, utilizing the conditional likelihood of G given X [HTFF09]. The log-likelihood for N observations looks like the following [HTFF09]:

$$l(\theta) = \sum_{i=1}^{N} log\, p_{g_i}(x_i; \theta) \tag{2.7}$$

with $p_k(x_i; \theta) = Pr(G = k|X = x_i; \theta)$.

K-Nearest Neighbors The k-nearest neighbors (KNN) algorithm is a supervised learning method that can be used for regression and classification. For this thesis the focus will only be on its use for classification. An input is classified based on a majority vote among the classes of its k nearest neighbors [HTFF09], where k is an integer. The nearest neighbors are found based on a distance metric, a common choice is the Euclidean distance, but there are many other options like cosine distance and Manhattan distance.

In Hastie et al. [HTFF09] the k-nearest neighbor fit for $\hat{Y}$, with $\hat{Y}$ being the prediction of the output, is defined as follows:

$$\hat{Y} = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i \tag{2.8}$$

Linear Discriminant Analysis Linear Discriminant Analysis (LDA) is another approach which can be used if the different classes or categories are known beforehand. The goal is to find a linear combination, which is based on the input variables, that separates two or more categories from each other.

This explanation is taken from section 4.3 of Hastie et al. [HTFF09] where $f_k(x)$ is the class-conditional density of X in class $G = k$ and π_k is the prior probability of class k, with $\sum_{k=1}^{K} \pi_k = 1$. Applying Bayes theorem leads to

$$Pr(G = k|X = x) = \frac{f_k(x)\pi_k}{\sum_{i=1}^{K} f_i(x)\pi_i}. \tag{2.9}$$

Linear Discriminant Analysis uses Gaussian densities as class densities. In this case the class densities are all modeled as multivariate Gaussian, the assumption is made that all classes have a common covariance matrix $\mathbf{\Sigma}_k = \mathbf{\Sigma}\ \forall k$:

$$f_k(x) = \frac{1}{\sqrt{(2\pi)^p |\mathbf{\Sigma}_k|}} e^{-\frac{1}{2}(x-\mu_k)^T \mathbf{\Sigma}_k^{-1} (x-\mu_k)} \tag{2.10}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Support Vector Machine Support Vector Machines were developed by Cortes and Vapnik [CV95]. When used for classification it is an optimization algorithm for finding a hyperplane that separates data points based on the class they belong to.

In the case of two dimensional space the hyperplane would be a line. In this paragraph, which is based on the explanation of Support Vector Machines by Berwick [Ber03] and the initial paper on the subject by Cortes and Vapnik [CV95], the focus will mainly be on the two dimensional case.

The support vectors are the data points that are closest to the hyperplane, they are the data points influencing the optimal location of the hyperplane [Ber03].

The optimal hyperplane has the form

$$\boldsymbol{w}^T\boldsymbol{x} + b = 0 \tag{2.11}$$

where $\boldsymbol{w}$ is a weight vector, $\boldsymbol{x}$ is an input vector and b is a bias.

The method of optimal hyperplanes by Vapnik [Vap82], which is described in the paper by Cortes and Vapnik [CV95], shows that the problem of constructing an optimal hyperplane is a quadratic programming problem.

Decision Tree Tree-based methods can be applied to regression and classification tasks, in this case the focus lies on decision trees being applied to a classification task. In the context of classification, the features are distributed into different groups by applying a recursive algorithm, as stated by Strobl et al.[SMT09, p. 5], based on the different groups the class labels are added. The idea is to create a hierarchical structure which separates the different classes in the optimal way, based on decision rules.

The following explanation is based on the description by Filzmoser [Fil20]. In the case of classification one needs the proportion of class k observations in node m, which is defined as follows [Fil20]:

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I(y_i = k) \tag{2.12}$$

where m is the current node, representing a region R_m with N_m observations. The observations in node m are classified as the majority class in node m. There are different measures of node impurity $((Q_m(T))$ including the misclassification error and the Gini index, which are used to determine how the features of a dataset are split. They are defined as follows [Fil20]:

Misclassification Error:

$$\frac{1}{N_m} \sum_{i \in R_m} I(y_i \neq k(m)) = 1 - \hat{p}_{mk(m)} \tag{2.13}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Gini index:

$$\sum_{k\neq k'} \hat{p}_{mk}\hat{p}_{mk'} = \sum_{k=1}^{K} \hat{p}_{mk}(1-\hat{p}_{mk}) \tag{2.14}$$

Random Forest Bagging or bootstrap aggregation is a technique that was developed by Leo Breiman [Bre96] which is designed to improve the performance of machine learning algorithms. It is the foundation for the random forest approach.

This explanation is based on the chapter about random forests by Hastie et al. [HTFF09, p. 587]. The main idea is to average many models so that the variance can be reduced. In the case of classification a group of trees each propose a vote for a predicted class. The random forests approach, which was developed by Leo Breiman as well [Bre01], is a modification of bagging where a large collection of de-correlated trees is constructed and averaged, as described by Hastie et al. [HTFF09]. The prediction of a new point x is made based on a majority vote, which is defined as follows:

$$\hat{C}^{B}_{rf}(x) = majority\ vote\ \{\hat{C}_b(x)\}_1^B \tag{2.15}$$

where $\hat{C}_b(x)$ is the class prediction of the bth random-forest tree.

Neural Network The explanations in this section are based on the book "Neural Networks and Deep Learning" by Aggarwal [A$^+$18]. Neural networks are nonlinear statistical models which are based on biological principles. They are used to solve problems in different domains like image classification or weather forecasting and consist of neurons that are connected to other neurons. A neuron can have multiple inputs, each input has a weight that is associated to it.

Let $x_1, x_2, ..., x_n$ be the inputs, the corresponding weights for the input variables are $w_1, w_2, ..., w_n$, so the output can be defined as follows [A$^+$18]:

$$y = \sigma(\sum_{i=0}^{i=n} w_i x_i) \tag{2.16}$$

where σ is the activation function. The classical activation functions according to Aggarwal [A$^+$18] are listed below:

- $\sigma(x) = sign(x)$ - sign function
- $\sigma(x) = \frac{1}{1+e^{-v}}$ - sigmoid function
- $\sigma(x) = \frac{e^{2v}-1}{e^{2v+1}}$ - tanh function

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

The neurons are then organised in some type of network, one version is the feed-forward neural network, which was first published by Ivakhnenko and Lapa [IL+65] in 1965. The feed-forward neural network is based on the idea of the perceptron, which was invented by Warren McCulloch and Walter Pitts [MP43] in 1943, and the idea of a layered network of perceptrons which was introduced by Frank Rosenblatt [Ros58] in 1958. This type of network consists of multiple layers, one of them being the input layer, one output layer and potentially several layers in between that are called hidden layers. From the input layer each input is sent to every neuron in the first hidden layer, the outputs are then again sent to the neurons in the next hidden layer. This process continues until the last layer is reached.

One method that is often used to train a neural network is back-propagation which was proposed by Rumelhart et al. [RHW86]. The backpropagation algorithm consists of the following steps:

1. The pattern of the input is created and propagated forward through the network
2. The output of the network is compared to the actual expected output. The difference between the two values is considered as an error
3. The error is propagated back from the output layer to the input layer. The weights of the different connections between the neurons are changed based on their influence on the error.

2.1.2 Deep Learning

Deep learning is a form of machine learning which can employ either supervised, semi-supervised or unsupervised techniques. LeCun et al. [LBH15] formulated it as a method that allows computational models, which are composed of multiple processing layers, to learn representations of data with multiple levels of abstraction. Deep learning approaches were able to improve the state-of-the-art in many different fields like speech recognition [HDY+12], image recognition [KSH12] and natural language understanding tasks [CWB+11]. Most of the modern deep learning models are based on multi-layered neural networks.

2.1.2.1 Transformers

The transformer is a deep learning architecture which is based on the attention mechanism. It was proposed by Vaswani et al. [VSP+17] in their paper "Attention Is All You Need", this explanation is based on this paper. It formed the basis of current state-of-the-art models like BERT and GPT-4 and follows the encoder-decoder structure.

The encoder is built out of 6 identical layers, where each layer consists of two sub-layers. The first sub-layer implements a multi-head self-attention mechanism and the

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

second sub-layer is a position wise fully connected feed-forward network. The decoder is built out of 6 identical layers as well, each of them consisting of three sub-layers, being the two sub-layers in the encoder layer and an additional layer which performs multi-head attention on the output of the encoder stack. [VSP+17]

Figure 2.1, which is taken from Vaswani et al. [VSP+17], shows their proposed architecture for the transformer. The basic building blocks of the transformer are the attention blocks. The goal of an attention block is to add context related to all other tokens to each embedded token of an input sequence. This enables the network to learn which relationships between tokens are more relevant than others.

Figure 2.1 shows the encoder on the left side and the decoder on the right side. When considering machine translation from English to German as an example, the encoder receives the input, in this case the English sentence, and the decoder receives the desired output, which would be the same sentence but in the German language in this case.

Figure 2.2 which is also taken from Vaswani et al. [VSP+17] shows a multi-head attention block on the right side and a scaled dot-product attention block, which is a part of the multi-head attention block, on the left. The graph shows that the scaled dot-product attention block is utilized h times, each of the scaled dot-product attention blocks can fulfill a different task, based on the provided weight matrices. [VSP+17]

The input of the scaled dot-product attention block subsists of queries (Q), keys (K) and values (V). The queries and the keys are both ($n \times d_k$)-dimensional matrices and the values are a ($n \times d_v$)-dimensional matrix where n is the amount of elements in the input sequence ($x_1 ... x_n$). The attention is defined as follows:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{2.17}$$

Vaswani et al. [VSP+17] found it to be beneficial if they linearly project the queries, keys and values h times with different learned linear projections to d_k or d_v dimensions, depending on the case. They called this approach multi-head attention and defined it as follows:

$$\begin{aligned} MultiHeadAttention(Q, K, V) &= concat(head_1, ..., head_n)W^O \\ where\, head_i &= Attention(QW_i^Q, KW_i^K, VW_i^V) \end{aligned} \tag{2.18}$$

with the following definitions:

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

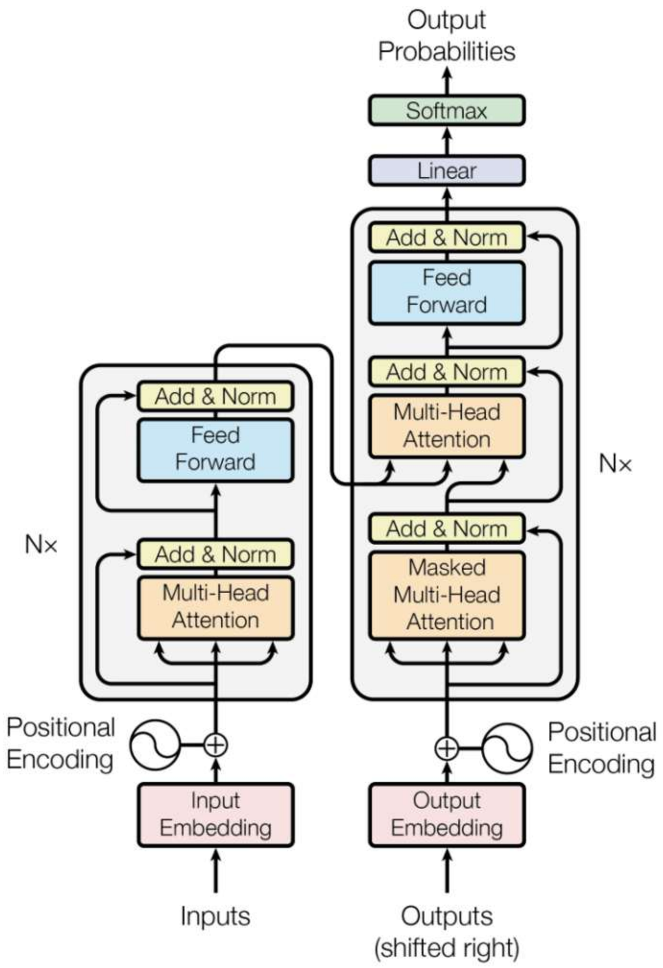


Figure 2.1: Transformer architecture by Vaswani et al. [VSP⁺17]

$$\boxed{\begin{aligned} W_i^Q &\in \mathbb{R}^{d_{model} \times d_k} \\ W_i^K &\in \mathbb{R}^{d_{model} \times d_k} \\ W_i^V &\in \mathbb{R}^{d_{model} \times d_v} \\ W^O &\in \mathbb{R}^{hd_v \times d_{model}} \end{aligned}}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

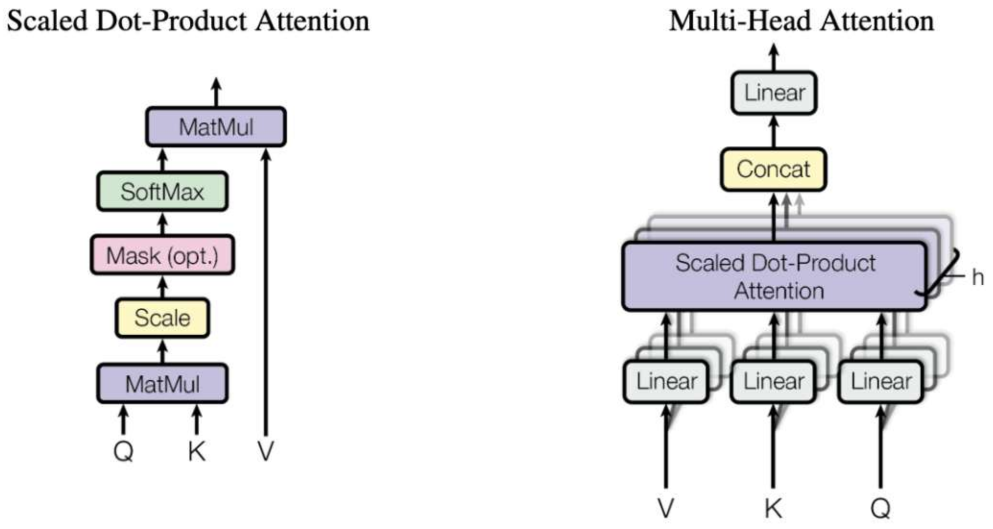


Figure 2.2: Scaled Dot-Product Attention and Multi-Head Attention by Vaswani et al. [VSP+17]

2.2 Natural Language Processing (NLP)

2.2.1 Tokenization

Tokenization is the process of separating a text into smaller chunks. In this subsection the concept of tokenization will be explained based on the explanation by Jurafsky and Martin in their book "Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition" [JM]. The two most common variations are sentence tokenization and word tokenization. The goal of sentence tokenization is the separation of a text with multiple sentences into separate sentences, the goal of word tokenization is the separation of a text into single words. One might start with looking at a very simple example, consider the following text:

This is a simple example text. It contains two sentences.

Now, when sentence tokenization and word tokenization is applied one after the other, the result might look like the following:

1. This is a simple example text. | It contains two sentences.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

2. `This` `is` `a` `simple` `example` `text` `·` `It` `contains` `two` `sentences` `·`

In the first step the two sentences are separated from each other, in the second step each word is an entity, and the dot at the end of each sentence as well. This might be different according to the implementation, this is just an example. To tokenize a text like this a few very simple rules are already sufficient, one only needs to be able to separate multiple sentences based on the ending dot and words based on white space. In addition the tokenizer has to be aware of the dot as a sentence ending token and therefore consider each dot as a separate token. This is a very trivial example, it can get way more complicated rather quickly. One might consider the following sentence:

> This sentence isn't as simple anymore, the rules that
> were applied so far would not be sufficient in this case.

The tokenizer needs to be able to consider additional special characters like comma and apostrophe and it needs to be able to deal with the "isn't" construct. Here it needs to be decided how the tokenizer should handle it, theoretically there are a lot of options like the following:

1. `isn` `'` `t`
2. `is` `n't`
3. `isn'` `t`
4. `is` `not`

There are a lot of further examples where a simple white space based separation is not sufficient, for example any sentence that contains the city "New York", here it might lead to bad search results if "York" is considered as a separate token, therefore it might be a good idea to consider "New York" as a single token.

One approach for handling names of all kinds, for example person names, cities and companies but also years and dates in general, is named entity recognition.

Named entity recognition has the goal to assign meaning to named entities in any kind of text. Different implementations exist, for example rule-based systems and machine learning based systems. The problem of named entity recognition is not solved yet,

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

the best systems achieve a F-Score of around 93% [MP98].

The following example shows a simple case where the named entity recognition worked as expected:

in: Frank moved to New York in 2006

out: [Frank]$_{\text{Person}}$ moved to [New York]$_{\text{City}}$ in [2006]$_{\text{Time}}$

In general one needs to consider that a lot of languages have a set of specific challenges. In Chinese for example each character has a meaning on its own, in combination with other characters the meaning can change. Chen et al. [CSQH17] used the following sentence as an example and showed, that it could be seen as consisting of three words, five words or even seven individual characters. In the following examples the language code "zh" is used for Chinese and the language code "en" is used for English.

zh: 姚明进入总决赛

en: Yao Ming reaches the finals (Initial sentence)

The first version splits the sentence into three parts:

zh: 姚明 进入 总决赛

en: YaoMing reaches finals (Tokenized using Chinese Treebank)

The second version splits the sentence into five different parts:

zh: 姚 明 进入 总 决赛

en: Yao Ming reaches overall finals (Tokenized using "Peking University" segmentation)

The last version splits the sentence into each of the individual characters:

zh: 姚 明 进 入 总 决 赛

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

en: Yao Ming enter enter overall decision game (Tokenized using character-wise segmentation)

It turns out that the character-wise segmentation is sufficient for most NLP tasks regarding the Chinese language [LMS+19].

2.2.2 Phonetic Algorithms

Phonetic algorithms are used to encode a word into a representation that contains some information regarding its phonetics. One phonetic algorithm for the English language is Soundex [Rob18] which was developed by Russell and Odell and patented in 1918. It is a phonetic algorithm for indexing names by sound and it is used in many popular database systems like Oracle [Soua] and PostgreSQL [Soub].

The basic idea is that a word is turned into a representation consisting of one letter at the beginning, followed by three numbers, the conversion table can be seen in Table 2.1.

The Soundex algorithm performs the following steps:

1. All letters are turned into upper case letters, punctuation marks are removed
2. The first letter of the word is kept and ignored in the following transformations, it is used in the end as a prefix.
3. Vowels (A, E, I, O, U), semivowels (W, Y) and the letter H are replaced with 0, as they should be ignored in the end
4. The remaining letters are transformed based on the rules in the conversion table which is shown in Table 2.1
5. All the occurrences of the same number next to each other, for example "11" are replaced with just one occurrence of the number (from "11" to "1" in this case). The number 0 is ignored, so it is still possible that there are two zeros next to each other
6. All the zeros are removed from the result
7. The final code should have a length of four, one letter at the beginning and three numbers following the letter. If there are not enough numbers, the remaining numbers are filled up with zeros. If there are more than three numbers all the numbers after the third number are removed

An example is presented in the following list:

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

1. Consider the surname Mueller-Luedenscheidt as an example
2. In step 1 the "-" is removed, all letters are turned into upper case: **MUELLER-LUEDENSCHEIDT**
3. In step 2 the first letter (M) is stored, it is not affected by the transformations in the next steps
4. In this case there are no semivowels the letter H and all the vowels are replaced with 0: M**00**LL**0**RL**00**D**0**NSC**000**DT
5. The remaining letters are replaced based on the conversion table (Table 2.1), the result is the following: M00**44**0**64**00**3**0**522**000**33**
6. Now all occurrences of the same number next to each other (except zeros) are reduced to just one occurrence of the number: M00**4**0640030 5**2**000**3**
7. All zeros are removed: **M4643523**
8. In the last step the code is shortened to length 4: **M464**

Letter	**Code**
B, F, P, V	1
C, G, J, K, Q, S, X, Z	2
D, T	3
L	4
M, N	5
R	6

Table 2.1: Conversion Table from the Soundex algorithm by Russell and Odell [Rob18]

For the German language, as well as for the English language, exist different kinds of phonetic algorithms, Wilz [Wil05] lists multiple different phonetic algorithms, some for the English language and some for the German language. Two of the phonetic algorithms for the German language are cologne phonetics (Kölner Phonetik) which was developed by Postel [Pos69] and Phonet, which has two different approaches which are both described in this article by Michael [Mic88].

The cologne phonetics algorithm transforms a word into a numeric representation. A central part of the algorithm is a transformation table, which is shown in Table 2.2. The following three steps are performed in the algorithm:

1. The letters are transformed into numbers based on the rules which are specified in the transformation table

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

2. For all the numbers that appear multiple times next to each other, all the appearances, besides the first one, are removed
3. All zeros, besides one being at the start, are removed

To show a more concrete example, one can look at the transformation of the name Müller-Lüdenscheidt. One has to consider that this name counts as one word in this algorithm, as both names are connected by hyphen. This example is taken from Wikipedia [wike]:

1. Based on the transformation table, the letters are turned into the following representation: 60550750206880022
2. Now all the numbers that are shown multiple times next to each other are reduced to the number just showing once: 6050750206802
3. In the last step all zeros that are not the first number in the representation are removed, in this case all zeros are removed: 65752682

Letter	Context	Code
A, E, I, J, O, U, Y		0
H		-
B		1
P	if not in front of H	1
D, T	if not in front of C, S or Z	2
F, V, W		3
P	if in front of H	3
G, K, Q		4
C	if in the initial sound before A, H, K, L, O, Q, R, U or X	
C	if in front of A, H, K, O, Q, U or X, if not in front of S or Z	
X	if not in front of C, K or Q	48
L		5
M, N		6
R		7
S, Z		8
C	if after S or Z	
	if initial sound and not in front of A, H, K, L, O, Q, R, U, X	
	if not in front of A, H, K, O, Q, U or X	
D, T	if in front of C, S or Z	
X	if after C, K or Q	

Table 2.2: The transformation table of the cologne phonetics algorithm by Postel [Pos69]

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

2.2.3 Part-of-speech Tagging

A part-of-speech tagger or a POS-tagger is a tool which is used to add certain tags to words in a text based on grammatical rules and context or in other terms determine the part of speech. The goal is to analyze and understand the syntactic structure of a sentence by categorizing each word into its appropriate grammatical class.

A few examples for parts of speech in the English language are noun, adjective, verb, pronoun, and adverb. There are different tag sets, a popular one for the English language is the Penn tag set, which was developed in the Penn Treebank project [MSM93, Pen]. For the German language the STTS [STT95, WSJB17] and the Tiger tag set [BDH$^+$02] are popular. Figure 2.3 shows a simple example where POS-tagging has been applied to a German sentence, the STTS tag set has been used. The tags that have been used in the example are listed and described in Table 2.3, the STTS tag set contains 54 tags.

STTS-Tag	**Part of Speech in German**	**Example**
NE	Eigenname (named entity)	Hans, Berlin
VVFIN	Infinitiv voll (full infinitive)	gehen, ankommen
ART	bestimmter/unbestimmter Artikel (definite/indefinite article)	der, die, das, ein, eine
ADJA	attributives Adjektiv (attributive adjective)	[den] **langen** [Roman]
NN	normale Nomina (common nouns)	Tisch, Herr, Buch

Table 2.3: POS-Tags from the STTS tagset [WSJB17] with explanation, related to the example

The RFTagger is a POS-tagger which was created by Schmid and Laws [SL08]. It supports multiple languages like German, Hungarian and Russian. It is a Hidden-Markov-Model (HMM) tagger which computes the POS tag sequence which is the most likely one for a given word sequence.

TU Bibliothek Your knowledge hub WIEN
Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

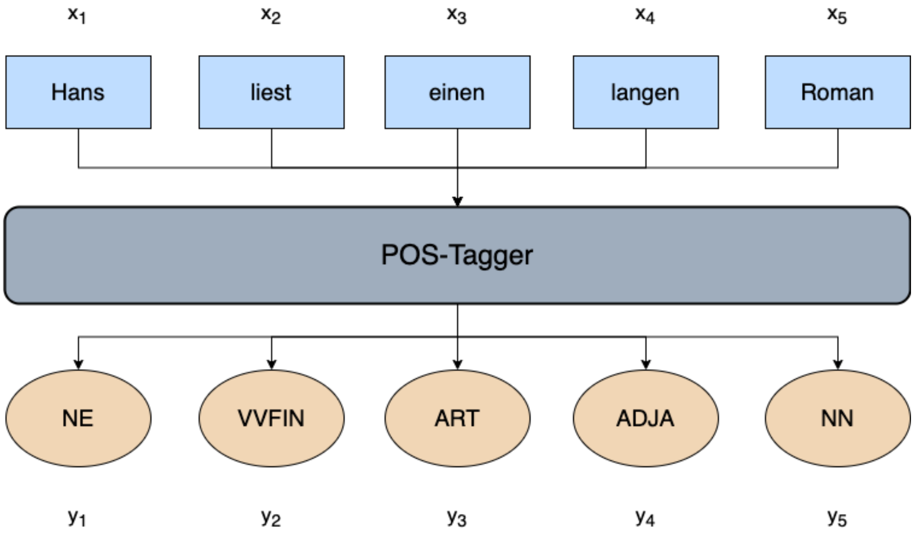


Figure 2.3: A simple POS-tagging example in German using the STTS tag set [WSJB17]

2.2.4 Sentiment Analysis

Sentiment analysis is a text categorization task which aims at detecting the sentiment which is expressed in a given text. In the simplest case the goal is to have a distinction between positive and negative sentiment, some sentiment analysis tools provide a neutral sentiment as a third option. Common scenarios where sentiment analysis is used are product review analysis or movie review analysis.

Wankhade et al. [WRK22] published a book called "A survey on sentiment analysis methods, applications, and challenges" in 2022, where they give an interesting overview of the field of sentiment analysis. Figure 2.4, which is taken from Wankhade et al. [WRK22] gives an overview of the different approaches, they separate them into lexicon based approaches, machine learning approaches, hybrid approaches and other approaches. The following explanations are based on the before mentioned book by Wankhade et al. [WRK22]

Lexicon based approaches are based on the idea of a collection of tokens where each token has a certain score which indicates whether the token is neutral, positive or negative. This collection of tokens is a lexicon. There are different approaches for assigning a score to a token, one might for example assign the values [-1, 0, 1] for [negative, neutral, positive] tokens respectively or use a value range, for example the interval [-1,1]. Once a score has been assigned to each token of a text the different scores, according to their class, are summed up separately. After that the overall sentiment is assigned based on the highest score among the scores. The advantage of a lexicon based approach is that there is no training data required, a lexicon needs to be created though. A disadvantage is that it does not work very well across different domains as the vocabulary or even the

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

sentiment of certain words might be different.

Machine learning approaches use different machine learning algorithms to assign a certain sentiment to a text. Supervised learning methods require training on a training set, common algorithms that are used in the domain of sentiment analysis are Naive Bayes, Support Vector Machine and Decision Tree.

A hybrid approach is a combination of machine learning approaches and lexicon based approaches, there have been some successful implementations where a hybrid approach was able to perform better than the separated models, but according to Wankhade et al. [WRK22] it is still a promising research field where a lot of additional research should be conducted.

Aspect based approaches aim at detecting different aspects of a text, so that the sentiment regarding each aspect can be assigned separately. In the final step the different sentiments across the aspects are aggregated. These approaches are popular when analyzing reviews, for example product reviews or hotel reviews. For a simple example consider different aspects of a smartphone $A = [camera, battery\ life, display, storage]$ and a review $R =$"the camera is amazing but the battery is draining fast". This reviews contains two different aspects, the camera and the battery life. The camera receives very positive feedback therefore the sentiment of the "camera" aspect is positive, the battery life receives negative feedback, therefore the "battery life" aspect has a negative sentiment.

Transfer learning is a technique where a pre-trained model is used to train a new model. This approach is popular as it turned out that features that have been trained for one domain are often useful for other domains as well. One model that is often used for transfer learning is BERT, which has been developed by Devlin et al. [DCLT18] at Google.

2.2.5 BERT

BERT (**B**idirectional **E**ncoder **R**epresentations from **T**ransformers) is a language model that was developed by Devlin et al. [DCLT18] The training of BERT consists of two steps, pre-training and fine-tuning. Both steps are shown in Figure 2.5, which is taken from Devlin et al. [DCLT18]

The architecture of the two steps is identical besides the output layer. It is a multi-layer bidirectional Transformer encoder based on the original Transformer that was proposed by Vaswani et al. [VSP$^+$17] A good example for the difference between a unidirectional and a bidirectional approach is shown in the BERT Github repository by Devlin [ber], where the sentence "I made a bank deposit" is discussed, with the focus being on the word "bank".

In the case of a unidirectional approach the model would either pay attention to the context on the left side ("I made a") or the context on the right side ("deposit"). Using the

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

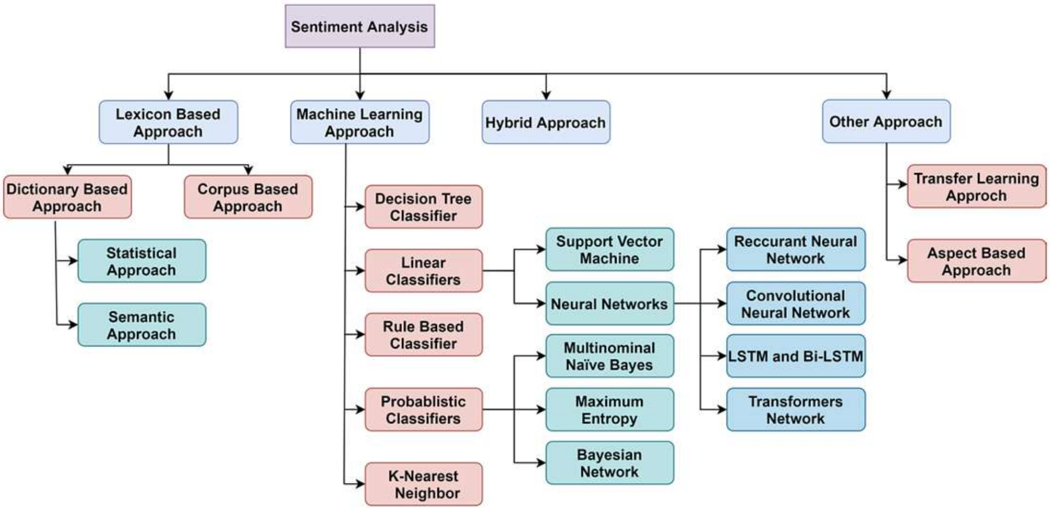


Figure 2.4: Different approaches for sentiment analysis, taken from Wankhade et al. [WRK22]

bidirectional approach both sides are considered, the word "bank" is therefore represented by the context "I made a ... deposit" [ber].

BERT was pre-trained on two unsupervised tasks, namely "Masked LM" and "Next Sentence Prediction". [DCLT18]. During the training for the "Masked LM" task some of the input tokens were randomly replaced with either a predefined token (80%), a random token (10%) or the unchanged token (10%). The model is trained with the target of predicting the tokens which have been replaced just based on the context [DCLT18].

The "Next Sentence Prediction" task is based on the connection of two sentences. Devlin et al. [DCLT18] model this connection with a simple data representation that contains three fields. Two of the fields are the sentences, the third field is the label that states whether the second sentence is the next sentence after the first sentence or not. A simple example for this is shown in the listing below:

- Case 1:
  - Sentence A: I was going for a walk with my dog.
  - Sentence B: It saw a bird and started running after it.
  - label: IsNext
- Case 2:
  - Sentence A: I was going for a walk with my dog.
  - Sentence B: Vienna is the capital of Austria

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

- label: NotNext

While the pre-training of BERT is expensive [ber] the fine-tuning is relatively inexpensive. The input can be either single text or pairs of text, as shown above. This property allows one to model many different tasks using the same pre-trained model.

BERT was able to outperform state-of-the-art models on eleven tasks at the time it was released, in October 2019 a blog post by Google was published which stated that BERT was incorporated into Google Search. [blo]

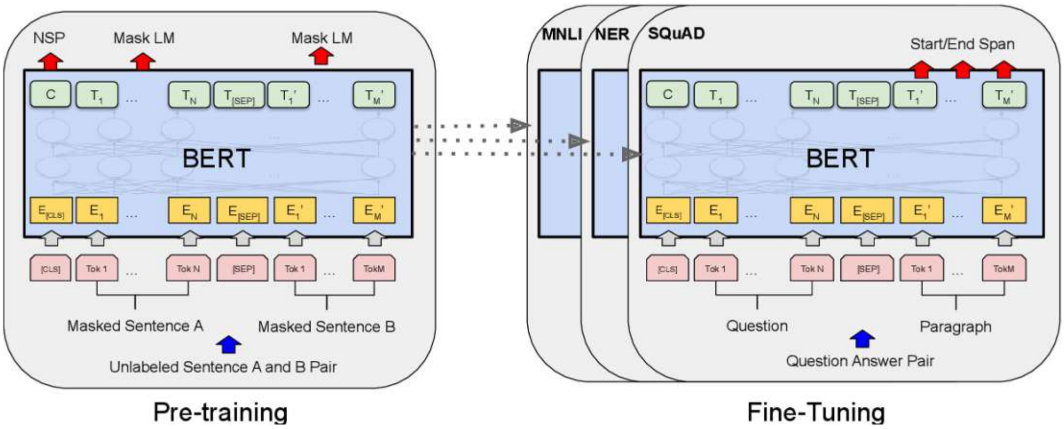


Figure 2.5: Overall pre-training and fine-tuning procedures for BERT by Devlin et al. [DCLT18]

2.3 Related Work

This section aims at giving a brief overview about a selection of related work.

2.3.1 Alliteration and hyperbole in marketing and politics

Alliteration and Hyperbole are two figures of speech that find use in marketing. Davis et al. [DBB16] show that alliterations might have a positive effect on the chance of a product being purchased. Fox et al. [FND19] measured the impact of an ad by tracking the eye movement of participants. Their results demonstrate the positive impact of the use of alliteration on consumer attention. These references are included to demonstrate that alliteration is researched in marketing and that there might be an impact on consumer behavior.

Stuckey [Stu17] cited Walter Dean Burnham, saying that critical elections are characterized by abnormally high intensity. This turns a hyperbole into a natural tactic according to Burnham's opinion. A hyperbole intentionally overstates the case. Even if the audience is not fully captured by it, it might be moved a little bit more by a hyperbolic statement in comparison to a more balanced and rational statement. They

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

describe hyperbolic arguments as excessive and overwhelming and claim that audiences persuaded by a hyperbole are carried along and do not inch carefully towards a conclusion.

Swartz [Swa76] conducted research regarding hyperbole in politics. He analyzed two Bena baraza cases, hyperbole was used in both. He concluded that the hyperbole did not work in the favor of its users. The examples used are very hyperbolic, one of two brothers stated that he doesn't know the man who accuses him of owing money, while the man actually was his brother. This was interpreted as the first man not knowing his brother anymore, as he is not behaving like a brother should, in his opinion. In the second case the damage done to someones property was highly over-exaggerated.

Considering this, Swartz [Swa76] made some conclusions about the use of hyperbole in politics, which will be stated in this paragraph. He suggested that the rare appearance of hyperbole is tied to it being best-suited for situations in which there are several aspects of reality bearing on the same value, yet affecting differently that value's implications in the situation. What this effectively implies, is that hyperbole is used in situations where its user feels the need to structure reality such that certain aspects overshadow others. In addition, it is argued that speakers might almost always feel that they must structure reality for their listeners, if they want to gain their support, and that exaggeration is not the only way to alter a statement for that purpose.

The discussion continues with the statement that individuals might exaggerate most in the areas of their greatest weaknesses, speaking of their greatest areas of weaknesses according to their own perception. Furthermore it is suggested that a hyperbole is more likely to occur if a speaker believes that the audience has certain ethical values, which might lead to the speaker trying to appeal to these ethical values using over exaggeration.

2.3.2 Hyperbole detection using Natural Language Processing

Troiano et al. [TS18] worked on the computational exploration of exaggeration. They built a corpus called HYPO, containing overstatements collected on the web, validated via crowd-sourcing. The chosen approach is classification, the set of used algorithms includes Logistic Regression, Naive Bayes, k-Nearest Neighbors, Decision Trees, Support Vector Machine and Linear Discriminant Analysis. Their experimental results lead to the conclusion, that automatic hyperbole detection could be successfully executed based on semantic features. The semantic features that they extracted are imageability, subjectivity, unexpectedness, polarity and emotional intensity.

Tian et al. [TkSP21] published a model called HypoGen for generating hyperboles at the clause or sentence level. The program expects an input prompt **A** and amis at generating a subject **B** and a predicate **C**. The example which is listed in their work is the following: Consider "*the party is lit*" as input prompt **A**, they would want to generate a sentence like "*the party is so lit that even the wardrobe is dancing*" where "*the wardrobe*" would be the subject **B** and "*is dancing*" would be the predicate **C**.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Zhang et al. [ZW21] propose an unsupervised approach for hyperbole generation, which uses a fine-tuned version of BART [LLG^{+}19] and a BERT-based ranker for selecting the best candidate. They created an open source dataset containing hyperboles in the English language called HYPO-L, this dataset was used in this thesis as well.

Vorakitphan et al. [Vor21] published a tool called PROTECT (**PRO**paganda **T**ext d**E**te**CT**ion) which focuses on propaganda technique classification. They used 14 different propaganda techniques like flag-waving, whataboutism and exaggeration. First they are classifying the text on the token level, where each token is either propagandist or not. After that the propagandist tokens are classified according to the 14 propaganda categories.

Schneidermann et al. [SHP23] performed edge and minimal description length probing experiments on pre-trained language models in the context of hyperbolic information being encoded in the pre-trained language models. Their results showed, that hyperbolic information is encoded in pre-trained language models to a limited extent.

2.3.3 Opinion Type Classification

Othman et al. [OHMI15] classified sentences based on whether they are opinionated statements, comparative opinionated statements, superlative opinionated statements or non-opinionated statements. The main method used for this task is Part-of-Speech (POS) Tagging using the Stanford Part-Of-Speech Tagger. The tag set used is the Penn Treebank P.O.S Tags tag set. They worked with text written in the English language.

Yu et al. [YKD08] explored the characteristics of political opinion expression and found that recognizing the sentiment alone is not sufficient for political opinion classification. They compared the average sentiment level of Congressional debates to the ones found in neutral news articles and movie reviews, where movie reviews showed higher sentiment levels and neutral news articles showed lower sentiment levels on average. Furthermore they analyzed how sentiment is expressed in different kinds of texts and found out, that the choice of topics plays a very important role in the case of political opinion expression, which is reflected in nouns, while the sentiment expression in movie reviews is more adjective-centered. They compared the results of manual sentiment annotation and found out, that a significant number of political opinions are expressed in a neutral tone.

Afzaal et al. [AUFF19] proposed a model called "enhanced multiaspect-based opinion classification" which extracts explicit and implicit aspects from tourist reviews and classifies the multiaspect opinions into different polarity classes. The model consists of three different methods, namely a probabilistic co-occurrence-based method, a hierarchy-based implicit aspect extraction method and a multiaspect opinion classification method. The probabilistic co-occurrence-based method captures the similarities between aspects using the co-occurrence of sentiment words with aspects, after that the similar aspects

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

are merged into a group. The hierarchy-based implicit aspect extraction method utilizes grammatical relationships between opinion words and aspects to create an aspect-specific lexicon, after that a hierarchy for extracting the implicit existence of aspects in reviews is built. An example can be seen in Figure 2.6. The multiaspect opinion classification approach is used to generate features from reviews. Once these three steps are completed some multilabel classification algorithms are applied to classify the multiaspect opinions into different polarity classes.

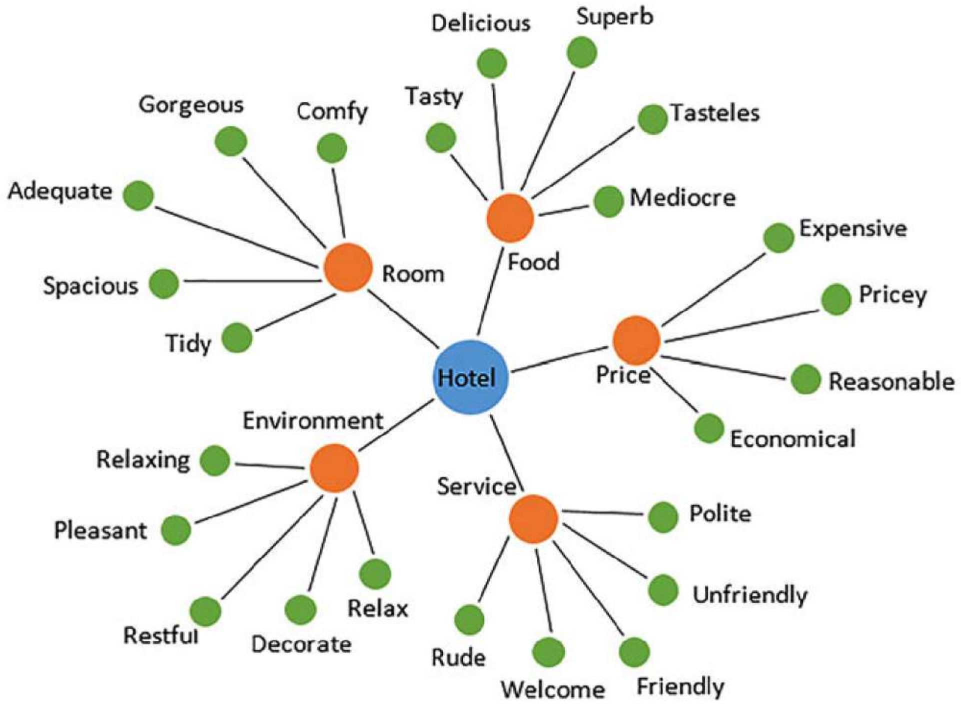


Figure 2.6: Example of an aspect-sentiment hierarchy as described by Afzaal et al. [AUFF19]

Liu [L+11] published a book on web data mining which includes a chapter on opinion mining. In this chapter they defined the three basic components of an opinion as follows:

- **Opinion holder:** The person who communicates an opinion on something
- **Entity:** The entity on which the opinion is expressed by the opinion holder
- **Opinion:** The view, sentiment or appraisal which is communicated by the opinion holder on the object

This chapter gave an overview over the algorithms, approaches and models that were used in the experiments that were conducted in the scope of this work. Furthermore it

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek WIEN Your knowledge hub

presented a selection of related work. The following chapters will focus on describing the experiments and the results that were achieved.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

CHAPTER 3

Design

3.1 Research Questions

The first section of this chapter focuses on describing the requirements that are derived from the research questions. The requirements need to be fulfilled so that the research question can be answered.

RQ1: *How high is the precision of a rule-based approach for topic classification in political statements using corpora extracted from Wikipedia?*

To successfully answer this research question a method for computing the precision needs to be defined. The statements which are retrieved using the implemented method are manually evaluated as either being relevant to the topic or not. Based on this binary classification the precision can be calculated.

RQ2: *How does an approach using a German tag set and a tagger for the German language, instead of an English tag set and a tagger for the English language, for opinion type classification in German sentences, hold up against the approach developed by Othman et al. regarding opinion type classification in English sentences? This question will be evaluated using precision*

This question is answered by comparing the achieved precision to the precision that was achieved in the experiments by Othman et al. [OHMI15]. The results of the experiments are manually evaluated, the precision is calculated based on the results of the manual evaluation.

RQ3: *The Cologne phonetics algorithm can be used to transform a word into a numerical representation depicting the underlying phonetics. Which additional constraints need to be added so that the numerical representation resulting from the Cologne phonetics algorithm is feasible for alliteration detection with 95 percent precision?*

The third question is answered by computing the precision on a dataset containing 605 alliterations, which was downloaded from the website of Ulrich Mehner [meh]. The results can be automatically evaluated, as it is known that all the entries are alliterations. The precision is calculated based on the amount of detected alliterations.

RQ4: *Troiano et al. developed an approach for classifying English sentences as either hyperbolic or not. Is this approach applicable to the German language as well? This question will be evaluated using accuracy, precision, recall and the F1-Score.*

This question is evaluated using a dataset containing hyperboles, therefore it can be evaluated automatically. The HYPO-L dataset by Zhang et al. [ZW21] will be used as a basis, as it is an openly available. The hyperboles need to be translated as they are currently only available in the English language. It contains around 3200 sentences where around 1000 are hyperboles, 200 hyperboles and 200 literal sentences are manually translated, in addition all the sentences are machine-translated using Google Translate. The results using the 400 manually translated sentences are compared to the results using the same 400 sentences in their machine-translated version and to the results which are achieved on the full machine-translated dataset. The accuracy, precision, recall and F1-Score can be computed as we know the class of each sentence.

3.2 Methodological Approach

This section describes the methodological approach that will be followed in this thesis. The before mentioned research questions are answered by applying one of two different approaches. RQ2 and RQ4 will be answered by implementing an approach that has been developed for the English language so that it can be applied to the German language. RQ1 and RQ3 will be answered by applying evolutionary prototyping, an algorithm will be outlined, implemented, tested and improved until it can be used to answer the research question.

The following listing contains the methodology that is followed throughout this work:

1. **Literature Research:**

   A literature research will be conducted which includes the following keywords among others: natural language processing, opinion mining, sentiment analysis, opinion type classification, topic classification, hyperbole detection, hyperbole, alliteration, alliteration detection. The master thesis by Stefan Zaruba [Zar21] will be the first entry point.

2. **Creating datasets and working with existing datasets:**

   Many existing datasets will be used during this work, if not available new datasets will be created. The first dataset that will be created is a dataset containing

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

political statements extracted from the Austrian national council protocols which are available on the website [aus]. Besides that a partly translated version of the HYPO-L datasets, which is a dataset containing English hyperboles, will be created. 400 sentences are manually translated and every sentence of the dataset (around 3200) will be machine-translated as well. Among the existing datasets that will be used are the DeWaC German corpus by Baroni et al. [BBFZ09, deWa, deWb, FE13], the German SentiWords dataset by Gatti et al. [RQH10, GGT15] and the alliteration dataset which is published on the website of Ulrich Mehner [meh].

3. **Using Natural Language Processing methods and Machine Learning models:**

   Various different natural language processing methods will be used to tackle the problem of hyperbole detection, alliteration detection, opinion type classification and topic classification in the context of political statements. In addition pre-trained machine learning models will be used if available and new models will be trained if required.

4. **Implementation:** The code for the experiments will be written in the Python 3 programming language [VRD09], Jupyter Notebooks [KRKP$^+$16] will be used for the experiments and the evaluation. Libraries and existing frameworks will be used if available. The implementation will be containerized using Docker [Mer14], so that it is easily possible to reproduce the results.

5. **Evaluation:** The evaluation of the experiments will be done automatically if possible and manually if required. The automatic evaluation will be done using existing Python libraries, common metrics like precision, recall, accuracy and F1-score will be used if possible. The manual evaluation will be done using doccano [NKK$^+$18], which is an open source data labeling tool developed by Nakayama et al. and freely available on Github.

6. **Data Analysis and Data Visualization:** The results of the experiments will be analyzed so that conclusions can be drawn and potential improvements on the implemented approaches can be derived. Insights gathered from the data analysis will be visualized using libraries for drawing plots, for example Matplotlib [Hun07].

3.3 Experiment Design

This section is used to describe the design of the experiments that were conducted to answer the research questions.

To answer the first research question it was necessary to find topics which might be discussed in the Austrian national council. Three topics were chosen, namely climate change, feminism and the European migrant crisis. The implementation of the topic classification approach could then be used to search for statements regarding these three

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

topics. The retrieved statements were then manually evaluated as being either relevant or irrelevant to the topic. Based on these two classes it was possible to compute the precision of the approach. The data analysis was conducted based on the topic-related terms that were extracted from the Wikipedia article, so that it is possible to see whether the retrieved terms are generally relevant to the topic or not.

The experiments for the second question were focused on rebuilding the approach by Othman et al. [OHMI15], they used an approach based on part-of-speech tagging. First it was necessary to research part-of-speech tagging for the German language, find an appropriate tag set and a tagger that supports it. After that the tags from the tag set were matched to the tags used by Othman et al. [OHMI15] and the political statements were tagged. The statements are then put into four different categories based on the part-of-speech tags defined earlier, three of them being opinion types (opinion, comparative opinion, superlative opinion) and a fourth category for statements that are not opinionated. A small subset of the statements that were classified as containing one of the three opinion types were manually evaluated based on whether they contain the opinion type or not. The binary results were used to calculate the precision. In addition two different approaches were implemented, one being rule-based and one being based on a dictionary. The dictionary contains positive, comparative and superlative forms of around 13.000 adjectives in the German language, it was downloaded from Wiktionary [Wikf]. The rules that were implemented are grammatical rules for comparative and superlative forms. The results of the two additional approaches were evaluated in the same way as the main approach, a subset was manually evaluated.

The third research question was answered by implementing an algorithm that is able to detect alliterations in the alliteration dataset by Ulrich Mehner [meh]. Different settings and approaches were tried until the algorithm was able to detect more than 95% of the alliterations in the dataset. The precison was computed automatically in this case as it was given that the dataset contains alliterations only. In addition to that an experiment was conducted on the political statements. A subset of the retrieved alliterations was then manually evaluated to check how well the current algorithm would perform on any free text. These results were used to determine problematic cases that can show up in free text and derive potential improvements to the algorithm.

The first step to answering the fourth research question was the computation of the semantic features which are used by Troiano et al. [TSÖT18] in their approach for hyperbole detection. As they used specific datasets and libraries for the English language it was necessary to research how the same semantic features can be computed for the German language. Once the semantic features could be computed they were used in supervised machine learning experiments using the same algorithms as Troiano et al. [TSÖT18] and the Random Forest algorithm in addition. It was a classification task with two classes, literal and hyperbolic. Three different dataset were used in the experiments, they are all based on the English HYPO-L dataset by Zhang et al. [ZW21], where one of

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

them is a full machine-translated version of the dataset, one a manually translated subset containing 400 sentences (200 for each of the two classes) and the same subset using the machine-translated version so that the potential impact of the machine translation could be seen as well. The machine-translated dataset was created using Google Translate. The experiments were conducted using 10-fold cross validation. As the dataset already contained the classes it was possible to evaluate the experiments automatically. Precision, recall, accuracy and the F1-score were computed and compared to the results achieved by Troiano et al. [TSÖT18] in their experiments. In addition an experiment on statements from a single protocol of the Austrian national council was conducted. The statements which were classified as being hyperbolic were manually evaluated, so that the effect on free text could be evaluated as well using precision.

3.4 Datasets

This section describes the datasets that were created in the scope of this master thesis.

The most essential dataset for this thesis contains around 65.000 political statements that were extracted from the protocols of the Austrian national council, which are available in the HTML format on the website [aus]. Next to the statements the dataset contains the speaker and the party of the speaker. In some cases a speaker has been written in different ways, therefore a dataset containing every form of every speaker was created, these forms were then all mapped to one unique form which was then used to map all speakers to their unique form. The party was added to each unique speaker as well.

This dataset is essential for this thesis as it provides the foundation for working on figure of speech detection, topic classification and opinion type classification in the context of political speech, which is the overall topic of this thesis. Some analysis was necessary to extract the data in a clean manner as the HTML follows a unique schema.

The columns of the political statement dataset are the following:

- **speaker:** The speaker of the political statement
- **speech:** The political statement
- **file:** The protocol name in the format NRSITZ_{*number*}_PARSED, for example NRSITZ_00087_PARSED for the parsed version of the 87. sitting.
- **party:** The party that the speaker belongs to
- **unique_speaker:** The unique speaker, this column was added for the cases where the same politician was mentioned in slightly different ways

The following listing presents one example of the political statements dataset:

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

- **speaker:** Abgeordneter Michael Bernhard
- **speech:** Natürlich können wir sagen, zwischen den Achtziger- und den 2020er-Jahren ist sehr viel passiert, aber wir dürfen keinen Moment darauf stolz sein, denn das ist in Wahrheit viel zu wenig für die vielen Jahrzehnte, die inzwischen verflossen sind.
- **file:** NRSITZ_00087_PARSED
- **party:** NEOS
- **unique_speaker:** Abgeordneter Michael Bernhard

As a result of the experiments of the first research question three subsets of the political statement dataset were created that contain the political statements that were retrieved based on the topic classification approach and a classification for each of the statements as either being relevant or irrelevant regarding one of the three topics. The topics are climate change, feminism and the European migrant crisis.

To prepare the political statements dataset for the experiments regarding the second research question, the RFTagger by Schmid and Laws [SL08] was used to annotate all the statements with part-of-speech tags. The political statement dataset was extended by one additional column called "tagged", this column contains the political statement in its POS-tagged form. In the case of the example that was listed above, the "tagged" column would look similar to the item presented below, the only difference being that each tag is preceded by a $\backslash t$ and no white space in between (for example Natürlich$\backslash t$ADV). In this listing a color and white space was added and the $\backslash t$ was removed to enhance readability:

- **tagged:** Natürlich ADV können VFIN.Mod.1.Pl.Pres.Ind wir PRO.Pers.Subst.1.Nom.Pl.* sagen VINF.Full.- , SYM.Pun.Comma zwischen APPR.Zwischen den ART.Def.Dat.Pl.Masc Achtziger- TRUNC.Noun und CONJ.Coord.- den ART.Def.Dat.Pl.Fem 2020er-Jahren N.Reg.Dat.Pl.Fem ist VFIN.Sein.3.Sg.Pres.Ind sehr ADV viel ADV passiert VPP.Full.Psp , SYM.Pun.Comma aber CONJ.Coord.Aber wir PRO.Pers.Subst.1.Nom.Pl.* dürfen VFIN.Mod.1.Pl.Pres.Ind keinen PRO.Indef.Attr.-.Acc.Sg.Masc Moment N.Reg.Acc.Sg.Masc darauf PROADV.Dem stolz ADJD.Pos sein VINF.Sein.- , SYM.Pun.Comma denn CONJ.Coord.Denn das PRO.Dem.Subst.-.Nom.Sg.Neut ist VFIN.Sein.3.Sg.Pres.Ind in APPR.In Wahrheit N.Reg.Dat.Sg.Fem viel ADV zu PART.Deg wenig PRO.Indef.Subst.-.*.*.* für APPR.Acc die ART.Def.Acc.Pl.Neut vielen PRO.Indef.Attr.-.Acc.Pl.Neut Jahrzehnte N.Reg.Acc.Pl.Neut , SYM.Pun.Comma die PRO.Rel.Subst.-.Nom.Pl.Neut inzwischen ADV verflossen ADJD.Pos sind VFIN.Sein.3.Pl.Pres.Ind . SYM.Pun.Sent

TU WIEN Bibliothek Your knowledge hub
Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

An additional dataset was created which contains 13.000 German adjectives in their positive, comparative and superlative forms if available. The data was extracted from Wiktionary [Wikf]. This dataset was used for an additional approach that was implemented in the scope of research question 2. It is a simple data-driven approach where the same opinion types (opinionated, comparative opinionated, superlative opinionated) as in the approach by Othman et al. [OHMI15] are computed based on the occurrence of an adjective in one of the three forms available in the dataset. If there was no comparative or superlative form available it was replaced by "-", two example entries of the dataset can be seen in Table 3.1:

positiv	**komparativ**	**superlativ**
hoch	höher	am höchsten
portofrei	-	-

Table 3.1: Two examples from the German adjectives dataset

As a result of the experiments for the second research question, there exist subsets of the statements that were annotated by the algorithm as containing one of the three opinion types and a binary manual evaluation.

For the fourth research question a machine-translated German version of the HYPO-L dataset by Zhang et al. [ZW21] was created using Google Translate. In addition 400 sentences (200 for each class), were manually translated, to increase the quality of the translation. The label 0 stands for the class "literal" and the label 1 is used for the class "hyperbolic". One problem is that the machine translation was lacking in quality in some cases, as the first example will show. A second problem is that the original dataset contains a lot of idioms which cannot be translated word by word, which also is a problem when using machine translation. There is still a need for a large manually created or translated German hyperbole dataset, the HYPO-L dataset by Zhang et al. [ZW21] could still be very useful as a basis. The list below will show four examples from the dataset:

- Example 1:
  - **label:** 0
  - **english:** We were pelted with rotten tomatoes
  - **manually translated:** Wir wurden mit faulen Tomaten beworfen
  - **machine translated:** Wir waren mit faulen Tomaten ausgestoßen
- Example 2:
  - **label:** 0
  - **english:** She bit the thread in two

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

  - **manually translated:** Sie hat den Faden durchgebissen
  - **machine translated:** Sie biss den Faden in zwei Teile

- Example 3:
  - **label:** 1
  - **english:** He was boiling over with indignation
  - **manually translated:** Er kochte über vor Empörung
  - **machine translated:** Er kochte mit Empörung
- Example 4:
  - **label:** 1
  - **english:** Her steadfast belief never left her for one moment
  - **manually translated:** Ihr festgefahrener Glaube verließ sie nicht mal für einen Moment
  - **machine translated:** Ihr unerschütterlicher Glaube hat sie nie für einen Moment verlassen

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

CHAPTER 4

Experiments

4.1 Extracting technical terms from Wikipedia

The idea, which led to the series of experiments described in this section, is the extraction of technical terms from a Wikipedia article about a certain topic. The technical terms are then used to search for political statements concerning a certain topic. The political statements are extracted from the protocols of the Austrian national council, which are available to the public. They are then put into an inverted index [ZM06] so that it is possible to search for statements containing a certain term.

The pseudo code below (algorithm 4.1) shows the algorithm that was implemented for creating the inverted index. Each document has two properties, id and text. An empty key/value store R is created, where each word of each document's text is inserted as a key. Dictionaries are the key/value stores implemented in Python.

The value which is mapped to each word is an empty list. If a word is part of a document's text, the id of the document is added to the list. This allows for a basic full text search, as one gets the id's of all the documents containing a certain word, if the word is used as the input key to the dictionary.

In this approach, the terms are extracted from the Wikipedia article based on a large frequency list extracted from the deWaC German corpus [FE13, deWa].

The frequency list from the source [deWb] was pre-processed and a few obvious outliers were excluded, for example words that contained only one letter, like aaaaa, or words that contained special characters, which lead to a frequency list containing 1.121.589 words with their respective amounts of appearance in the deWaC German corpus [FE13, deWa]. Based on these amounts and the sum of all the amounts a percentage of appearance was calculated.

Algorithm 4.1: Creation of a simple inverted index R

```
Input: Set of documents D with size(D) > 0
$i \leftarrow 0$;
$R \leftarrow \{\}$;
while $i < size(D)$ do
    $t \leftarrow tokenize(D[i].text)$ List of words in the text of document D[i] ;
    $k \leftarrow 0$;
    while $k < size(t)$ do
        $w \leftarrow t[k]$;
        if $w$ not in $R$ then
            $R[w] \leftarrow []$ Empty list;
        end
        $R[w] = R[w] + (D[i].id)$ Append tuple with document id of D[i] ;
        $k \leftarrow k + 1$;
    end
    $k \leftarrow 0$;
    $i \leftarrow i + 1$;
end
```

The percentage of appearance is then used to specify whether a word is appearing too frequently. The value specified for these experiments is the mean of all the calculated percentages. So if the value of a specific word is below the mean, it is considered as a technical term.

Based on this approach the technical terms are extracted from the Wikipedia article about a specific topic. The extracted technical terms are then used to search for relevant speeches.

4.1.1 Process Description

The following paragraphs are going to explain the whole process with the help of an activity diagram [JR99]Figure 4.1.

In the first step, which is called "Provide a topic as input", any topic can be entered as input to the program. The topic needs to be the exact name of an article on Wikipedia. If the topic exists on Wikipedia, the whole text from the Wikipedia article is downloaded using the Wikipedia API [wikd]. In case there is no article about the topic, having exactly the name provided as input, an error is displayed.

Before the technical terms are extracted from the article, one is able to define ad-

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

ditional terms that should definitely be considered, even if they would not be considered according to the frequency list. It is written in a way so that every other word starting with a word from the custom terms is also considered.

One example for this is the word "Klima" which would definitely not be considered per default as it is used too frequently. If this word is defined as a custom term, not only the word "Klima" is preserved but words like "Klimawandel" and "Klimakrise" as well, even if they would not be considered according to the frequency list. This can be of great help in cases where a lot of technical terms that are relevant to a topic would be ignored because they are used too frequently.

After that optional step the technical terms are extracted from the article. The words are put into a set, so that one is left with all the unique words of the article, and afterwards every word in the set is then used as input to the frequency list. If the frequency of the usage of a word is below the overall mean frequency value, so at least slightly below average, the word is considered as a technical term.

All of the words that were selected above are then used as input on the previously defined inverted index, so that all speeches containing a certain word are then retrieved. In the best case one would only find speeches that are relevant to the topic that was defined in the first step, but this is not realistic. The evaluation section will describe the results of the experiments in detail.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

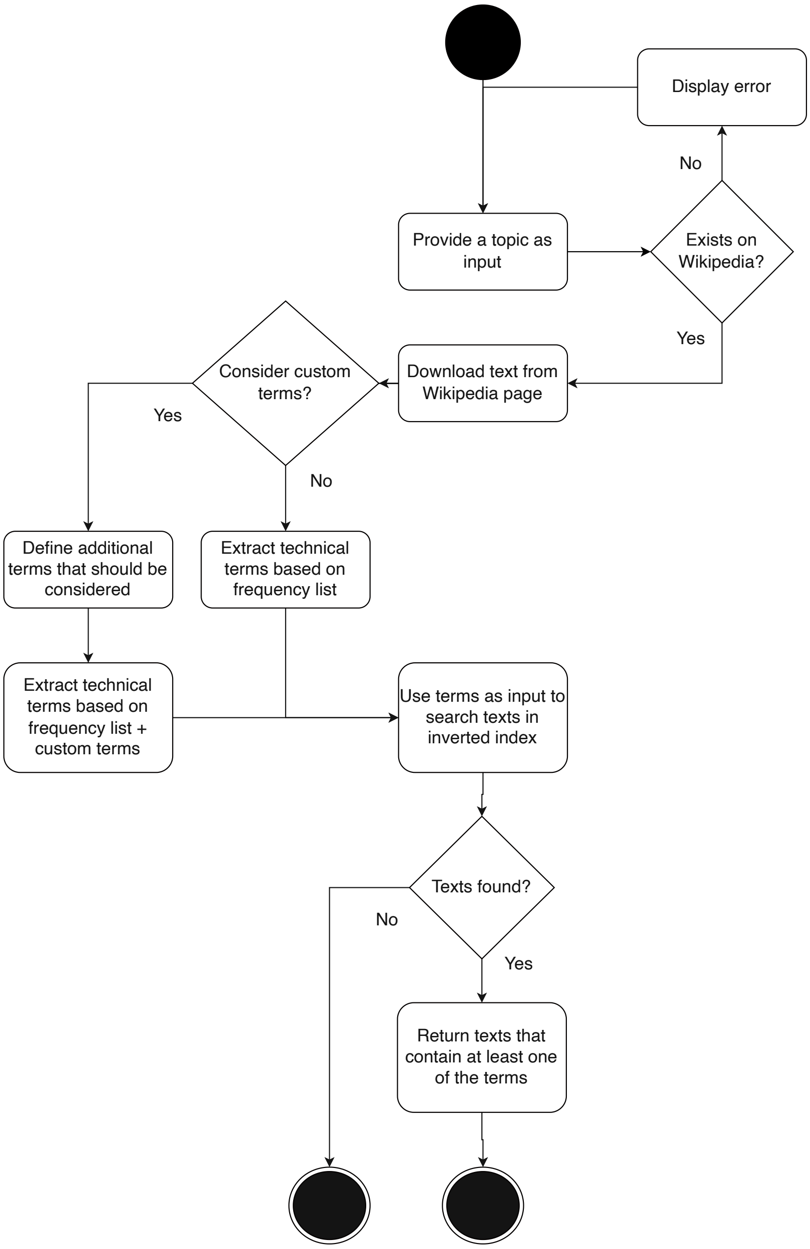

Display error
No
Provide a topic as input
Exists on Wikipedia?
Yes
Download text from Wikipedia page
Consider custom terms?
Yes
No
Define additional terms that should be considered
Extract technical terms based on frequency list
Extract technical terms based on frequency list + custom terms
Use terms as input to search texts in inverted index
Texts found?
No
Yes
Return texts that contain at least one of the terms

Figure 4.1: Activity diagram visualizing the process that was implemented to extract the technical terms from Wikipedia

4.1.2 Dataset Description

The dataset which was used to create the frequency list is the deWaC German corpus introduced by Baroni et al. in 2009 [BBFZ09, FE13, deWa]. In this case deWaC stands

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

for German web corpus. It is made up of texts which were extracted from the internet. The corpus contains more than 1.34 billion words and was created following the standards which were defined by Kilgarriff et al. in this work [KRPA10].

Some additional datasets were derived from this corpus, one being a frequency list. This frequency list [deWb] was the basis for the frequency list that was used to extract the technical terms. Only a few pre-processing steps were performed, so that a few obvious outliers were excluded from the frequency list. These outliers contain words that consist of one letter only, like "aaaaa", words containing special characters, like "ab&" and words containing more than 42 characters.

After all the pre-processing steps the frequency list contains 1.121.589 words with their respective amounts of appearance in the deWaC German corpus.

The paper by Baroni et al. [BBFZ09] introduces two additional corpora, namely ukWaC and itWaC, which stand for English web corpus and Italian web corpus respectively. The first two letters of all three corpora describe the domain. The text that were used for the deWaC were extracted from websites under the .de domain, texts for the ukWaC were extracted from websites under the .uk domain and texts for the itWaC were extracted from websites under the .it domain.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

4.2 Opinion Type Classification

This part of the thesis focuses on classifying the type of opinion communicated in a certain text. In this experiment the approach by Othman et al. [OHMI15] is followed. In their example, they focused on the English language and used the Stanford POS-Tagger [TM00, TKMS03, KT04]. A POS-Tagger, or Part-Of-Speech-Tagger, is a tool that is used to add certain tags to a part-of-speech in a text based on the definition and the context. A part-of-speech describes certain classes of words where each word in that class has similar grammatical properties. Common parts-of-speech are adjectives, adverbs and nouns.

The approach by Othman et al. uses these tags to classify texts as either non-opinionated, opinionated, comparative opinionated or superlative opinionated. Table 4.1 shows the POS-Tags of the Stanford POS-Tagger which are used in this approach to mark a text as one of the four classes. Table 4.2 contains a description of each tag. The Stanford POS-Tagger uses the Penn Treebank Tag Set [MSM93, Pen] for the English language [TKMS03, KT04]

Sentimental Category	POS-Tags (Stanford)
Non-Opinionated	-
Opinionated	JJ
Comparative Opinionated	JJR, RBR
Superlative Opinionated	JJS, RBS

Table 4.1: The four different sentimental categories defined by Othman et al. [OHMI15]

The goal of this experiment is to use the approach by Othman et al. but for the German language instead of the English language. According to Schmid and Laws [SL08], a more fine-grained tag set is often considered more appropriate for a language with a rich morphology, like German.

POS-Tag (Stanford)	Description
JJ	Adjective
JJR	Adjective Comparative
RBR	Adverb Comparative
JJS	Adjective Superlative
RBS	Adverb Superlative

Table 4.2: Used POS-Tags by Othman et al. [OHMI15] with description

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

For this experiment the RFTagger by Schmid and Laws [SL08] is used. It uses a tag set, which is similar to the STTS [STT95, WSJB17] or Tiger tag set [BDH$^+$02].

Table 4.3 shows the tags which are used in this experiment and Table 4.4 shows the description of the tags.

Sentimental Category	POS-Tags
Non-Opinionated	-
Opinionated	ADJA, ADJD
Comparative Opinionated	ADJA.Comp, ADJD.Comp
Superlative Opinionated	ADJA.Sup, ADJD.Sup

Table 4.3: Used POS-Tags in this experiment

There are different sub-forms of the tags mentioned in Table 4.4. In this experiment, they are all treated as the form depicted in the table.

The experiment will show, however, that this method does not produce the results we expect. Instead, a few adjustments have to be applied. This leads to two additional approaches for finding comparative and superlative forms.

POS-Tag	Description
ADJA	Attributives Adjektiv
ADJD	Adverbiales or prädikatives Adjektiv
ADJA.Comp	Comparative form of ADJA
ADJD.Comp	Comparative form of ADJD
ADJA.Sup	Superlative form of ADJA
ADJD.Sup	Superlative form of ADJD

Table 4.4: Description of the used POS-Tags in this experiment

The first additional approach is based on grammatical rules and regular expressions. In this method, the idea is to formulate the grammatical rules as regular expressions so that the regular expressions can be used to search for these patterns in the text. Table 4.5 shows the different rules which are implemented and used in the experiments.

The second additional approach is a data-driven approach. For this method, a dataset was created by processing data from Wiktionary [Wikf]. It contains around 13.000 adjectives,

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

along with their comparative and superlative forms, if available. The sets of comparative and superlative forms are then used as corpora; if one of the words appears in a text, the text is marked as a text containing a comparative or superlative opinion, respectively.

The major problem with the original POS-Tagging approach is a set of words which are superlative forms, but do not - usually - express an opinion. One example that showed up particularly often in this case is the word 'nächsten'. It is true, that this word is a superlative form, the three forms are 'nah', 'näher', 'am nächsten'. In our case it is misleading, as we are searching for opinionated statements, but the word 'nächsten' is used to refer to the next sitting, one example is 'In der nächsten Sitzung'.

To solve this problem, "trivial" superlatives to be ignored were gathered in a set. The dataset used for these experiments is a dataset that had been extracted from the Austrian parliament protocols (national council). It contains 63909 statements.

Rule	Example	Form
So ... wie	so groß wie	comparative
Nicht so ... wie	nicht so groß wie	comparative
Immer ...er	immer größer	comparative
...er als	größer als	comparative
Je ... desto	Je größer desto	comparative
Je ... umso	Je größer umso	comparative
am ...sten	am stärksten	superlative
am ...ßten	am größten	superlative

Table 4.5: Rules which are used in the rule-based approach

4.3 Alliteration Detection

This part of the thesis focuses on the detection of alliterations in texts. An alliteration is a figure of speech that describes the situation in which the first letter of multiple words that follow each other have the same sound. The words do not have to follow each other directly, steps in between them are allowed. Two examples for an alliteration in the German language are "Der frühe Vogel fängt den Wurm" and "Flora und Fauna".

The first example has three words with the same initial sound right after each other, being "frühe", "Vogel" and "fängt". This example shows that it is not enough to just focus on the first letter of a word, as different letters still can have the same sound in the German language.

The second example shows another valid example, where the first letters of the words "Flora" and "Fauna" have the same sound and the first letter of the word in between ("und") does not have the same sound. So if one wants to write an algorithm for detecting

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

alliterations, one might want to include a way to detect cases where different letters produce the same sound (as in the first example) and where there are steps of a certain size between two or more words, where the first letter has the same sound (as in the second example).

4.3.1 Phonetic Algorithms

To handle the first case, which is described above, an algorithm for encoding a word into some kind of representation that gives information about its phonetics might be helpful. One phonetic algorithm for the English language is Soundex [Rob18] which was developed by Russell and Odell and patented in 1918. It is a phonetic algorithm for indexing names by sound and it is used in many popular database systems like Oracle [Soua] and PostgreSQL [Soub].

The basic idea is that a word is turned into a representation consisting of one letter at the beginning, followed by three numbers. Letters are turned into numbers based on certain rules, which are not explained in detail in this section. To show one concrete example, the name "Britney" would be transformed into "BRTN" (as the letters a, e, i, o, u, y, h, and w are removed if they are not the first letter) and then the remaining letters, after the first letter, will be transformed into a numeric representation, which would lead to "B635" in this case. This is a simplification as there are additional rules to follow in the algorithm.

For the German language there are different kinds of phonetic algorithms, Wilz [Wil05] lists multiple different phonetic algorithms, some for the English language and some for the German language. Two phonetic algorithms for the German language are cologne phonetics (Kölner Phonetik) which was developed by Postel [Pos69] and Phonet, which has two different approaches which are both described in this article by Michael [Mic88].

In the algorithm that was developed to execute the experiments for this thesis, the cologne phonetics algorithm by Postel was used [Pos69]. For Python 3 there is an implementation of the algorithm by Nouvertné [nou], this one was used in these experiments.

The cologne phonetics algorithm transforms a word into a numeric representation. A central part of the algorithm is a transformation table, which is shown in Table 4.6. The following three steps are performed in the algorithm:

1. The letters are transformed into numbers based on the rules which are specified in the transformation table
2. For all the numbers that appear multiple times next to each other, all the appearances, besides the first one, are removed
3. All zeros, besides one being at the start, are removed

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

To show a more concrete example, one can look at the transformation of the name Müller-Lüdenscheidt. One has to consider that this name counts as one word in this algorithm, as both names are connected by hyphen. This example is taken from Wikipedia [wike]:

1. Based on the transformation table, the letters are turned into the following representation: 60550750206880022
2. Now all the numbers that are shown multiple times next to each other are reduced to the number just showing once: 6050750206802
3. In the last step all zeros that are not the first number in the representation are removed, in this case all zeros are removed: 65752682

4.3.2 Algorithm which is used in the experiments

This subsection will describe the algorithm that is used in the experiments. In general there are three parameters that the algorithm can be configured with.

The first parameter gives information on whether the cologne phonetics algorithm should be used to find the alliteration or not. If the cologne phonetics algorithm is not used, the alliterations are just searched based on their initial letter.

The second parameters name is "size" and specifies the size of the sublists of a text that are considered. When setting the size to three, having the sentence "Das hier ist ein kurzer Beispielsatz" the sublists would look like the following:

(Das hier ist), (hier ist ein), (ist ein kurzer), (ein kurzer Beispielsatz)

The procedure for generating the sublists is defined in a way where each sublist has to have exactly the same size. The sublists are used to find an alliteration of a certain size. If all the words in a sublist would have the same letter, or the same number if encoded using cologne phonetics, it would be considered as an alliteration. This is restrictive, as it would not find alliterations like "Flora und Fauna", which is a valid example. To make the algorithm less restrictive a third parameter is added.

The third parameters name is "steps", it is a list that could theoretically contain integers from 1 to the value of size - 1. In this example the maximum would be two, so the list cannot have more than two elements. It defines the amount of steps that is allowed between the first word of an alliteration and the second word of an alliteration. An example can be shown with the following sentence: "Die Flora und Fauna beschreiben zusammen die Natur". If the size is set to three the sublists would look like this:

(Die Flora und), (Flora und Fauna), (und Fauna beschreiben), (Fauna beschreiben zusammen) ...

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

If there wouldn't be any steps specified, the algorithm would not find an alliteration, as it is only looking for sublists where each word starts with the same letter, or in the case of cologne phonetics with the same number. But if the steps are set to a list containing two, like this [2], a step of two is allowed between the words as well. In that case the sublist "(Flora und Fauna)" would contain an alliteration.

If one would set the size of the sublists to four, the list which is provided as input to steps could contain one, two, three or any combination of the three values. If the list is empty the stricter approach is taken, where each word needs to have the first letter or number.

The pseudocode in algorithm 4.2 shows the procedure which is used to create the sublists of a text. It is a simple approach which expects a list of words as its input. The list of words is then splitted into multiple sublists containing exactly the amount of words that was specified using the the size parameter.

Algorithm 4.2: Returns a list of sublists, each sublist has size s

```
Input: List of words $w$ with $size(w) > 0$
Input: Integer $s$ where $s \leq size(w)$
Function getSublists($w$, $s$):
    $i \leftarrow 0$;
    $res \leftarrow []$ Empty list ;
    while $i < (size(w) - s + 1)$ do
        $res \leftarrow w[i : i + s]$ Words from index i until index i + s are put in a sublist
          and added to the result ;
        $i \leftarrow i + 1$;
    end
    return $res$ List of sublists of size s ;
```

algorithm 4.3 shows the short preprocessing routine that is applied to each sentence or text. The text is transformed into its lowercase version, afterwards it is tokenized so that all the words are a single entry in a list and at the end all the numbers are replaced with strings. That is interesting for sentences that contain constructs like "Er hatte **v**orher 40 **v**olle **F**ässer" which would be converted to "er hatte **v**orher **v**ierzig **v**olle **f**ässer" (it would be separated into a list, here it is just displayed as a single string for visual purposes), which would be detected as an alliteration using the rule which is defined for code 3 in the cologne phonetics algorithm as seen in Table 4.6.

To convert numbers into strings the python package num2words [num] is used, which is maintained by Virgil Dupras. Its predecessor pynum2word was developed in 2003 by Taro Ogawa, this new package is a continuation of pynum2word. It supports a lot of

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Algorithm 4.3: Returns preprocessed words from sentence

```
Input: String $sent$ with $size(sent) > 0$
Function getPreprocessedWords($sent$):
    $lowercase \leftarrow lower(sent)$;
    $words \leftarrow tokenize(lowercase)$;
    $nonnumeric \leftarrow replace\_numbers\_with\_strings(words)$;
    return $nonnumeric$;
```

different languages, for example, English, German, Spanish and Arabic.

algorithm 4.4 shows the whole alliteration detection algorithm. It takes four input parameters:

- sent: A sentence or text (String)
- s: The minimum size of the detected alliterations (Integer)
- st: A list of integers describing the steps that are allowed between words in the alliteration
- useCP: A boolean that is used for controlling whether cologne phonetics should be used or not

In the first step the input text is preprocessed using the logic that has been defined earlier (algorithm 4.3), all the words are then stored in the list with the name "*words*". If the "*useCP*" boolean has been set to true, the cologne phonetics algorithm is applied to each word in the list and the first number is taken and put into a list, which is called "*letters*" in this example. If the "*useCP*" boolean is set to false a similar approach is taken, but the cologne phonetics part is left out and just the first letter of each word is stored in the list "*letters*".

The resulting list ("*letters*") and the minimum size of the detected alliteration (Integer "*s*") are then used as an input to the getSublists(words, size) function (algorithm 4.2), the results are stored in a list called "*sublists*". In the next step the results from the previous step are used as input to the getFilteredSublists(sublists, useCP) function (algorithm 4.5). This step is just used if cologne phonetics was used before. It removes cases where each number is equal to zero, or in other words, each of the words of a part of a sentence started with the letters a, e, i, j, o, u or y, as these letters are handled by the rule which is defined for code 0 (Table 4.6).

To show one example illustrating this process, consider the sentence at the beginning as the following: "Ein Ai ist eine Art von Faultier". After preprocessing the sentence

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Algorithm 4.4: Returns true if a sentence contains an alliteration, false otherwise

```
Input: String $sent$ with $size(sent) > 0$
Input: Integer $s$
Input: List of Integers $st$
Input: Boolean $useCP$
Function containsAlliteration($sent, s, st, useCP$):
    $result \leftarrow []$
    $words \leftarrow getPreprocessedWords(sent)$
    if $useCP == True$ then
        $letters$ = firstColognePhoneticsNumberOfEach($words$)
    else
        $letters$ = firstLetterOfEach($words$)
    if $size(letters) < 2$ then
        return False
    end
    $sublists$ = getSublists(words, s)
    sublists = getFilteredSublists(sublists, useCP)
    foreach $sublist \in sublists$ do
        if the first character of each string in sublist is equal then
            result.add(True)
        end
    end
    foreach $step \in st$ do
        if hasAlliterationAtStep(sublists, step) then
            result.add(True)
        end
    end
    return True in result
```

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Algorithm 4.5: Returns filtered list of sublists

```
    Input: List of lists sublists with $size(sublists) > 0$
    Input: Boolean useCP
1   Function getFilteredSublists($sublists, useCP$):
2       if $useCP == True$ then
3           $cologne\_sublists \leftarrow []$;
4           foreach $sublist \in sublists$ do
5               if all cologne phonetics encoded words ∈ sublist start with 0 then
6                   continue;
7               else
8                   cologne_sublists.add(sublist)
9               end
10          end
11          return cologne_sublists
12      else
13          return sublists
14      end
```

with cologne phonetics and taking the first number of each word the "*letters*" list would look like this: [0,0,0,0,0,3,3]. The getSublists(words, size) method would be called with the "*letters*" list as input and a size, for this example the size is defined as three. The resulting list of sublists is the following: [(0,0,0),(0,0,0),(0,0,0),(0,0,3),(0,3,3)]. If the getFilteredSublists(sublists, useCP) function is now applied to this list, with the "useCP" parameter being set correctly to true, the first three sublists would be removed and the list would look like the following: [(0,0,3),(0,3,3)]. This step is included as the letters a, e, i, j, o, u or y could all be the reason for the value being 0, therefore one cannot decide whether it is an alliteration or not based only on the number 0.

In the next step the first for-each loop is used (line 13) to handle each sublist in the resulting list from the previous step. If all the elements in the sublist are equal, true is added to the list of results which is defined in line 2 above. This is important for the final step (line 23) which only checks whether true is included in the list of results and returns true or false based on the result of this check.

The second for-each loop which starts in line 18 goes over the steps that are defined in the input list "*st*". In the default case the list is empty. Generally it is describing whether there are steps allowed between the same sounds in an alliteration or not. If yes, the size of the steps can be defined, so if the list is equal to [1,2] steps of the size one and two are allowed.

For each of the steps the hasAlliterationAtStep(sublists,step) function is called (algorithm 4.6). This algorithm checks whether there is the same character after the step, so it is possible to skip characters. The sublists still have the specified size, but the

TU Bibliothek Your knowledge hub WIEN
Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Letter	Context	Code
A, E, I, J, O, U, Y		0
H		-
B		1
P	if not in front of H	1
D, T	if not in front of C, S or Z	2
F, V, W		3
P	if in front of H	3
G, K, Q		4
C	if in the initial sound before A, H, K, L, O, Q, R, U or X	
C	if in front of A, H, K, O, Q, U or X, if not in front of S or Z	
X	if not in front of C, K or Q	48
L		5
M, N		6
R		7
S, Z		8
C	if after S or Z	
	if initial sound and not in front of A, H, K, L, O, Q, R, U, X	
	if not in front of A, H, K, O, Q, U or X	
D, T	if in front of C, S or Z	
X	if after C, K or Q	

Table 4.6: The transformation table of the cologne phonetics algorithm by Postel [Pos69]

alliteration could be shorter, with *size* - max(st) being the minimum size. True is added to the list of results which is defined in line 2, if the criterion is fulfilled.

At the end (line 23) a check is performed on whether there is at least one occurrence of true in the list of results. If there is at least one occurrence, the algorithm returns true, which would mean that there is an alliteration in the text. In any other case false would be returned.

4.3.3 Experiment setup

In this subsection the experiments that were conducted are described in more detail. In total two different datasets were used. The first dataset consists of 605 alliterations which were extracted from the website of Ulrich Mehner [meh]. If the website is visited it automatically redirects to another website after a few seconds, but it is still possible to look at the alliterations for a short period of time. At the time of this writing there might be even more alliterations than 605, as the dataset was created in August 2022.

This dataset is helpful for evaluating whether the algorithm is detecting correct alliterations, but it does not help with evaluating the performance of the algorithm on any

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Algorithm 4.6: Returns true if one sublist has an alliteration with step st, false otherwise

```
Input: List of sublists $ls$ with $size(ls) > 0$ all sublists have same size
Input: Integer $st$ where $st \leq size(ls[0]) - 2$
Function hasAlliterationAtStep($ls$, $st$):
    $ind\_list \leftarrow 0$;
    $res \leftarrow []$ Empty list;
    while $ind\_list < (size(ls)$ do
        $ind\_sublist \leftarrow 0$;
        $sublist \leftarrow ls[ind\_list]$;
        $size\_sublist \leftarrow size(sublist)$;
        while $ind\_sublist < size\_sublist$ do
            if $ind\_sublist + st \geq size\_sublist$ then
                $ind\_sublist \leftarrow ind\_sublist + 1$;
            else
                if $sublist[ind\_sublist] == sublist[ind\_sublist + st]$ then
                    $res$.add(True);
                end
                $ind\_sublist \leftarrow ind\_sublist + 1$;
        end
        $ind\_list \leftarrow ind\_list + 1$;
    end
    return True in $res$ ;
    True if there was an alliteration considering step st
```

text input. For this case a dataset which contains more than 65000 statements from the Austrian national council was used, as in other experiments in this thesis. The dataset was created based on html protocols which are available on the website of the national council [aus].

To get the best possible results, the results of different setups of algorithm 4.4 were combined, so that multiple cases can be handled.

The setup for the best result on the alliteration dataset combined the results of the four different setups shown in Table 4.7. As some of the alliterations in the dataset are not continuous it is important to consider different step sizes as well as different sizes.

For the dataset which is consisting of political statements it was important to define stricter rules, as the rules for the alliteration dataset were way to permissive, which was shown in the amount of statements that were marked as containing an alliteration,

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

which was way above 90%. The rules can be found in Table 4.8, in this case no steps in between are allowed and the minimum size is set to four, which means that an alliteration needs to consist of at least four consecutive words with the same first letter, or the same first number in the case of cologne phonetics, to be counted as a valid alliteration.

Setup	Uses Cologne Phonetics	Size	Steps
Setup 1	No	2	[1]
Setup 2	No	3	[1,2]
Setup 3	Yes	2	[1]
Setup 4	Yes	3	[1,2]

Table 4.7: Different setups of the algorithm which were combined in the best attempt on the alliteration dataset

Setup	Uses Cologne Phonetics	Size	Steps
Setup 1	No	4	-
Setup 2	Yes	4	-

Table 4.8: Different setups of the algorithm which were combined to search for alliterations in the dataset of political statements

4.4 Hyperbole Detection

This part of the thesis focuses on the detection of hyperboles in texts. More precisely it aims at reproducing part of the results that were achieved by Troiano et al. [TSÖT18] when implementing an approach for hyperbole detection for the English language. In these experiments the same approach is implemented for the German language.

Troiano et al. [TSÖT18] worked on the computational exploration of exaggeration. They built a corpus called HYPO, containing overstatements collected on the web, validated via crowd-sourcing. The chosen approach is classification, the set of used algorithms includes Logistic Regression, Naive Bayes, k-Nearest Neighbors, Decision Trees, Support Vector Machine and Linear Discriminant Analysis. Their experimental results lead to the conclusion, that automatic hyperbole detection could be successfully executed based on semantic features.

4.4.1 Dataset

The dataset that was created by Troiano et al. [TSÖT18] is called HYPO and contains three different types of sentences, being hyperboles, paraphrases and minimal units.

The previously collected hyperboles were evaluated by five annotators, if three out of the five annotators decided that the sentence contains an exaggeration, the sentence was kept as a hyperbole in the final dataset. In total 854 sentences were judged, 709 of them were hyperboles in the final dataset. The annotators also collected information

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

about the words that are hyperbolic in their opinion.

A paraphrase in this context is a sentence that conveys the same message as a hyperbolic sentence but without the exaggeration. It is practically very similar to a hyperbolic sentence but with a few slight changes in the syntax and semantic so that it does not contain an exaggeration anymore. The final dataset contained 709 paraphrases, one for each hyperbole.

A minimal unit is a sentence that is not hyperbolic but contains hyperbolic words, which were collected during the annotating process. For each hyperbole the tokens which were selected by the majority of the annotators were chosen. These tokens were then used by Troiano et al. to extract sentences containing these tokens from corpora like the WaCKy corpus, which is a dump of the English Wikipedia. They selected the sentences from the WaCKy corpus [BBFZ09] based on the editorial criteria of Wikipedia which states that entries need to be neutral and verifiable, which leads to the conclusion that these sentences should not be hyperbolic. If they were not successful using the WaCKy corpus a Google search was performed. They collected 698 sentences of this category.

In general the HYPO dataset is an English dataset, so it would not fit for the purpose of the experiments described in this section, as they are focusing on the German language. The first idea was to translate the HYPO dataset to German, but the dataset is not publicly available. Zhang et al. [ZW21] worked on similar datasets, namely HYPO-L and HYPO-XL, while developing MOVER. HYPO-L is a manually annotated dataset containing English sentences which are either hyperbolic or literal. The dataset is not as complex as HYPO, as it does not contain paraphrases and minimal units. The sentences in HYPO-L were annotated by students with proficiency in English. Two students annotated each sentence, only if both students chose the same class, either hyperbolic or literal, the sentence was classified as the chosen class and kept in the dataset.

In this experiment a German version of the HYPO-L dataset is used. 400 of the sentences, 200 of each class, were manually translated. In addition all the sentences were automatically translated using the Google Translate function in a Google Sheet document. In the end the experiments were conducted with three different versions of the dataset: The whole machine translated dataset, the 400 manually translated sentences and the 400 sentences that were manually translated before but with their machine translated version.

To validate the performance on a dataset that is not related to HYPO-L at all, a dataset containing sentences from the 58. sitting of the national council of Austria (legislature XXVII) was used. The model was trained on the full machine translated version of HYPO-L. The sentences that were classified as hyperbole were manually evaluated, so that the precision can be calculated.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

4.4.2 Semantic Features

The set of semantic features that were computed by Troiano et al. consists of imageability, unexpectedness, polarity, subjectivity and emotional intensity.

According to the definition in the paper of Troiano et al. [TSÖT18] imageability refers to the extent to which a word has the capacity to conjure a mental picture. Their assumption is that speakers who hyperbolize might use words with a high value in imageability. To compute the imageability they used a resource by Tsvetkov et al. [TBG+14] which contains imageability ratings for 150.114 terms. For each sentence the imageability values of all the words were averaged.

In this experiment the dataset which was developed by Köper et al. [KIW16] was used to compute the imageability for German sentences. This dataset contains 350.000 German lemmatised words which are rated on four psycholinguistic attributes, one of them being imageability. All ratings were calculated using a supervised learning algorithm . During the feature engineering procedure the imageability values of each word in a sentence were averaged.

Unexpectedness is used under the assumption that hyperboles are less predictable expressions than literals. The idea is that words which are unexpected in a certain context might give a hint towards them being used in a hyperbolic way. Troiano et al. used pre-trained vectors by Mikolov et al. [MCCD13](from the Skip-gram model) and GloVe vectors by Pennington et al [PSM14].

The idea is that the words in a hyperbolic sentence have less similar meanings, which would mean that their vectorial representations are more distant from each other. The words are mapped to the vector representation of the two pre-trained vector sets mentioned above and then the cosine distance between all possible word pairs in a sentence are computed. The resulting features are then found in two ways, one being the average similarity among all of the word pairs and the second one being the lowest value of the pair similarities. This results in four values in total as they are computed for both vector sets.

In this experiment the GloVe embeddings [gita] and the Word2Vec embeddings [gitb] published on deepset.ai [emb] were used to compute the four unexpectedness values for the German language.

Polarity refers to the sentiment of a statement. Troiano et al. [TSÖT18] computed it using TextBlob [LKH+14] by Loria et al. and SentiWords [GGT15] by Gatti et al. TextBlob is a library that can be used for multiple NLP tasks, like the computation of sentiment. SentiWords is a dataset which contains the polarity values of 155.000 POS-tagged lemmas. The SentiWords dataset is used to compute the polarity of each

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

word, afterwards they are averaged. In total there are two polarity scores: the TextBlob score and the average of the polarity values which were gathered from the SentiWords.

As TextBlob and SentiWords are resources for the English language they needed to be substituted. There is a library called textblob-de [kil] for the German language which is developed by Markus Killer and there is a German version of the SentiWords dataset [RQH10] which was published by Remus et al.

These two resource where used in the same way as the English resources were used by Troiano et al.

Subjectivity is a value that is used to describe whether a statement expresses an opinion or objective information. Troiano et al. used TextBlob to calculate the subjectivity. As textblob-de does not support subjectivity yet, an alternative approach using the mdebertav3-subjectivity-german model [hug] developed by a team of the University of Groningen was chosen to calculate subjectivity. It was published by Folkert Leistra and Wietse de Vries on huggingface.co and it is a fine-tuned mDeBERTa V3 model [HGC21] which was developed for the second task of the CLEF 2023 CheckThat! Lab [cle] which was focused on subjectivity in news articles.

Emotional intensity gives information about the strength of the sentiment. Troiano et al. compute the sentiment using VADER, which was developed by Hutto and Gilbert [HG14]. VADER returns four values regarding the sentiment, being pos, neu, neg and compound, where pos stands for positive, neu for neutral and neg for negative. The compound score gives information about the overall sentiment, and the other three values show the level of the respective sentiment in the sentence. Unfortunately it was not clear which of the four values was used by Troiano et al., an assumption is made that all four of them are used.

As VADER is only applicable to sentences written in the English language, an alternative approach needs to be chosen. There is a library called GerVADER, which was developed by Tymann et al. [TLPG19] that aims at implementing VADER for the German language. GerVADER returns the same values as VADER, all four of them were used as a features when training the model.

4.4.3 Experiment Setup

Troiano et al. [TSÖT18] defined the task of hyperbole detection as a supervised learning problem, more specifically as a classification task which is based on sentences. It is a binary classification task with the classes hyperbolic and literal.

Various algorithms were used by Troiano et al. to conduct experiments using the semantic features described above, namely Logistic Regression (LR), Naive Bayes (NB), Linear Discriminate Analysis (LDA), k-Nearest Neighbors (KNN), Decision Trees (DT)

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

and Support Vector Machine (SVM). In the experiments for this thesis Random Forest (RF) was used in addition. The implementations of all the mentioned machine learning algorithms in the scikit-learn package , which was developed by Pedregosa et al. [PVG$^+$11], were used to train, test and evaluate the models.

A 10-fold cross validation was performed on the experiments which were conducted with the translated HYPO-L dataset, the results were evaluated using accuracy, precision, recall and the F1-score.

The experiment with the dataset containing the political statements was manually evaluated and the precision was computed based on the results of the manual evaluation. For this experiment random forest was used as a classifier.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

CHAPTER 5

Evaluation

5.1 Extracting technical terms from Wikipedia

The following section presents the results that were achieved by using the frequency-based approach for finding technical terms in a Wikipedia article. Three different topics were chosen, namely the European migrant crisis, climate change and feminism. The target was finding as many relevant speeches as possible, by searching the collection using the technical terms extracted from the Wikipedia article.

The results are visualized using bar charts. The bar charts are used to visualize the amount of relevant and irrelevant speeches that were returned and to show the amount of texts containing a certain technical term. The latter are ordered by amount and the ten terms with the highest amount of texts are shown. The plots visualize the data for three different sets of texts, namely for all texts, for all relevant texts and for all irrelevant texts.

5.1.1 Topic: European migrant crisis

The following subsection gives an evaluation of the results which were achieved using the frequency-based approach when trying to find technical terms which are then used to find speeches that are relevant to the European migrant crisis. The Wikipedia article used can be found here [wikc].

The first plot Figure 5.1 shows the amount of relevant and irrelevant speeches retrieved from the corpus. The speeches were manually evaluated on whether they are relevant to the topic or not.

In this case it is notable, that a lot of speeches were found, in total more than 3000 speeches. Only around one sixth of the speeches were actually found to be relevant to the

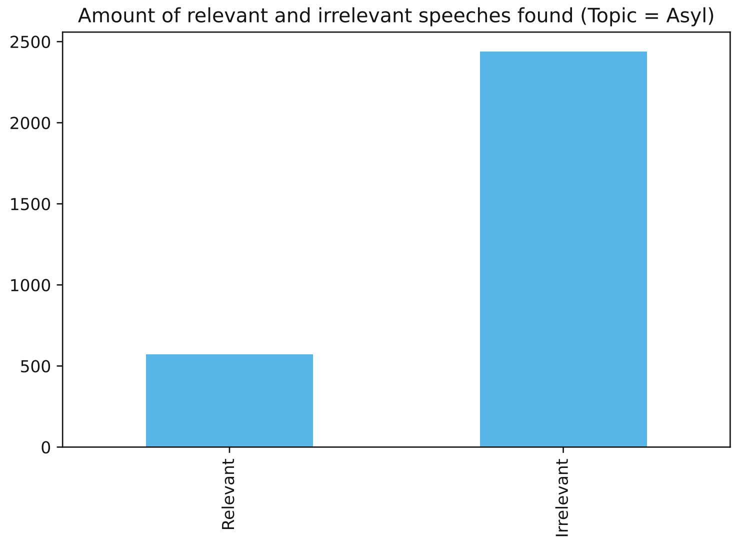


Figure 5.1: Value counts of relevant and irrelevant speeches (Topic = European Migrant Crisis)

topic, which is a bad result for this approach, as only every sixth speech that is presented to the end user would actually be relevant.

To get a better picture of why the irrelevant speeches were found, a data analysis on the technical terms retrieved from the Wikipedia article was conducted.

The Wikipedia article that had been used was not read before, such that the results are not based on a particularly well-suited article. In this case the article about the European migration crisis includes some information about the pandemic caused by COVID-19, which leads to a problem in that specific case, as many speeches in the corpus revolve around this topic.

The second bar plot Figure 5.2 shows the amount of speeches containing a specific technical term, ordered by amount. The first ten entries are displayed in the plot. It becomes evident here that "Pandemie" - a word which is mostly included in speeches that are irrelevant to the topic - is included in over 1000 speeches. In other words, the technical term is present in around every third speech which was retrieved. This is already more than five times as frequent as the second technical term, which is also mostly included in irrelevant speeches.

This strengthens the conclusion, that this word has a big impact on the result on this specific corpus, as it includes a lot of speeches revolving around this topic.

The third bar plot Figure 5.3 shows the amount of relevant speeches containing a specific technical term, ordered by amount. The first ten entries are displayed in the plot. Here

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

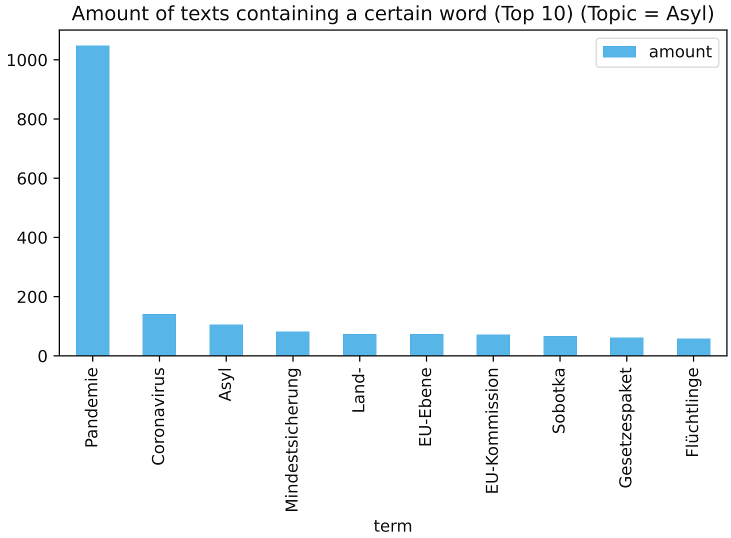


Figure 5.2: Amount of texts containing a specific word (Top 10) (Topic = European Migrant Crisis)

one can see, that they are all relevant to the topic and that the frequency is generally very low when compared to the the frequency of the word "Pandemie".

The fourth bar plot Figure 5.4 shows the amount of irrelevant speeches containing a specific technical term, ordered by amount. The first ten entries are displayed in the plot. Here one can again see the strong impact of the word "Pandemie".

5.1.2 Topic: Climate Change

The following subsection gives an evaluation of the results which were achieved using the frequency-based approach when trying to find technical terms which are then used to find speeches that are relevant to climate change. The Wikipedia article used can be found here [wika].

The first plot Figure 5.5 shows the amount of relevant and irrelevant speeches retrieved from the corpus. The speeches were manually evaluated on whether or not they are relevant to the topic.

In this case the method performed very well, as 89,03 percent of all retrieved speeches are relevant to the chosen topic. The following plots will give some insight to the technical terms that were extracted from the Wikipedia article.

The second bar plot Figure 5.6 shows the amount of speeches containing a specific technical term, ordered by amount. The first ten entries are displayed in the plot. Here one can see, that the first eight technical terms are relevant to climate change in many

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

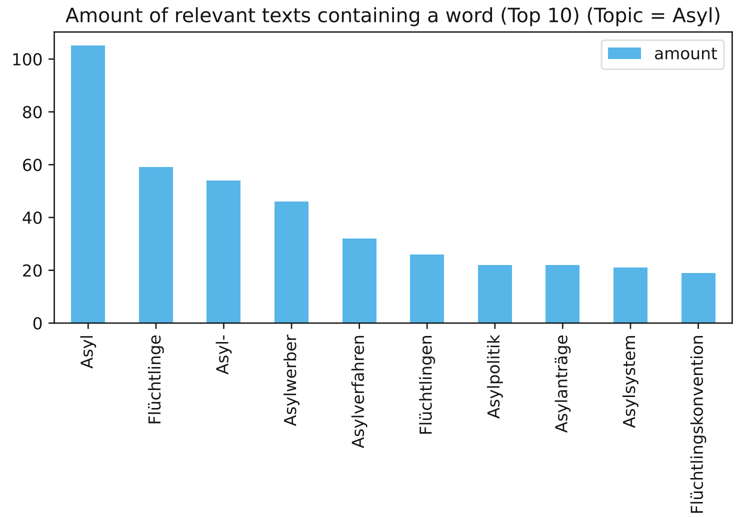


Figure 5.3: Amount of relevant texts containing a specific word (Top 10) (Topic = European Migrant Crisis)

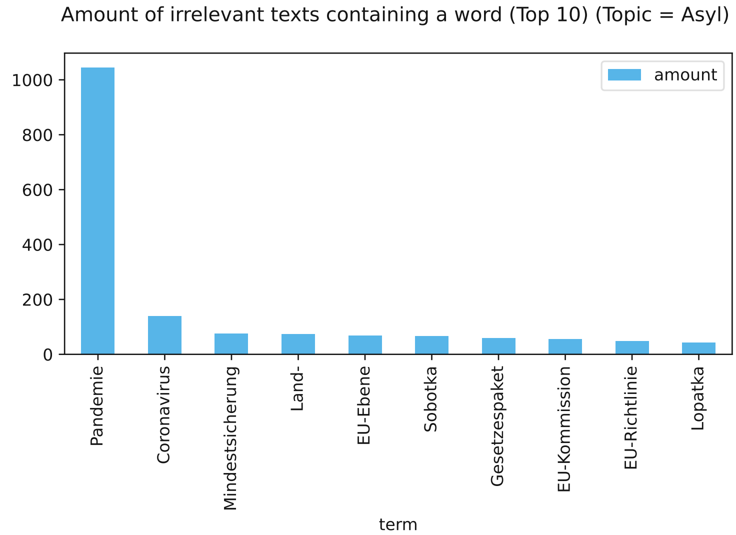


Figure 5.4: Amount of irrelevant texts containing a specific word (Top 10) (Topic = European Migrant Crisis)

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

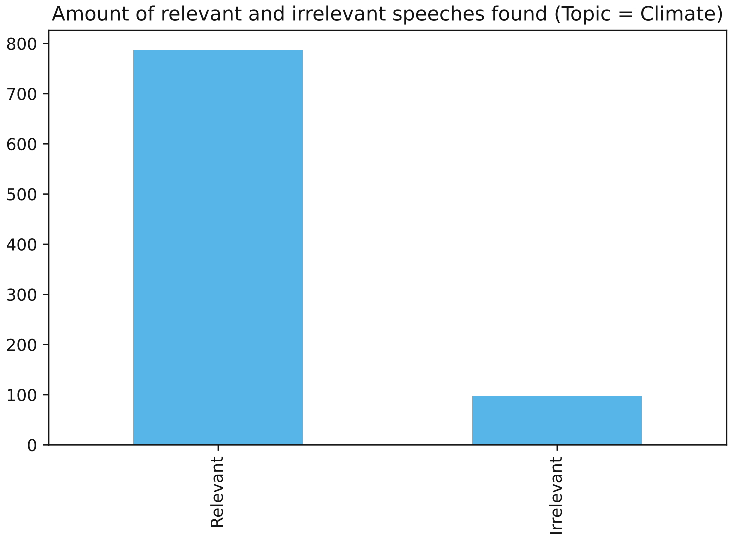


Figure 5.5: Value counts of relevant and irrelevant speeches (Topic = Climate Change)

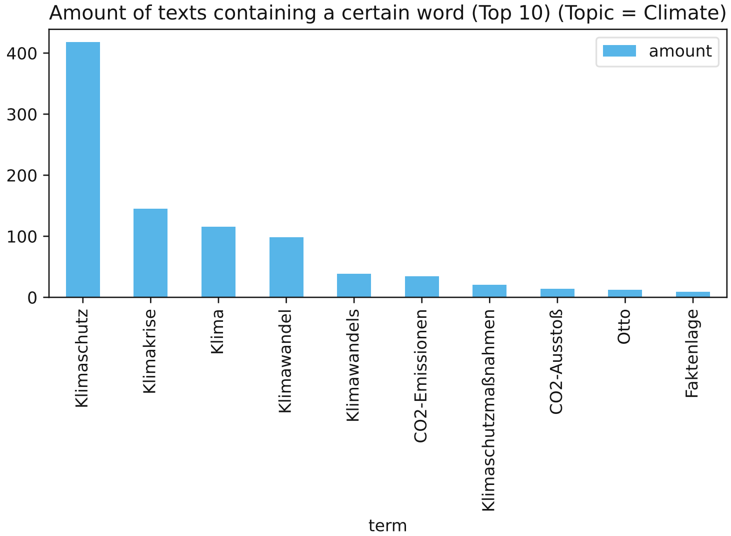


Figure 5.6: Amount of texts containing a specific word (Top 10) (Topic = Climate Change)

cases, "Klimakrise", "Klima" and "Klimawandel" were used in the social climate context as well.

The third bar plot Figure 5.7 shows the amount of relevant speeches containing a specific technical term, ordered by amount. The first ten entries are displayed in the plot. The technical term "Klimaschutz" performed best, as the amount of relevant speeches

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

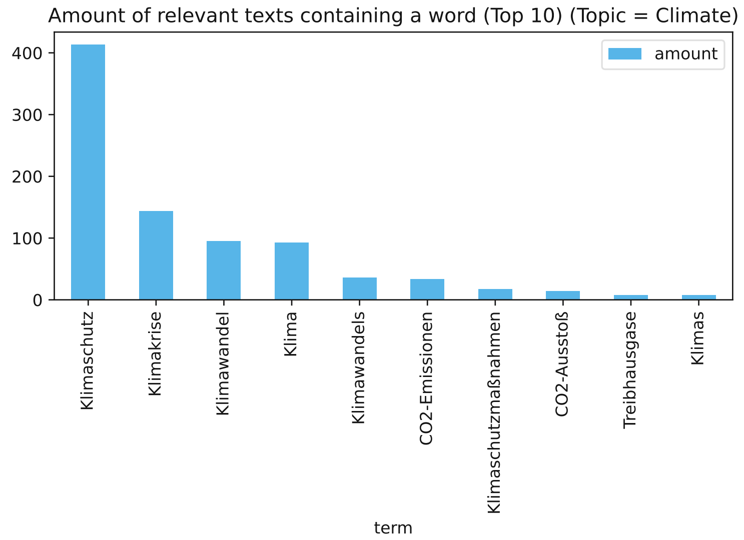


Figure 5.7: Amount of relevant texts containing a specific word (Top 10) (Topic = Climate Change)

containing it is higher as for any other technical term and because the total amount of speeches containing the term is nearly equal to the amount of relevant speeches containing it.

The third bar plot Figure 5.8 shows the amount of irrelevant speeches containing a specific technical term, ordered by amount. The first ten entries are displayed in the plot. This plot shows the note-worthy result, that even words like "Klimaschutzmaßnahmen" and "Klimaschutz" are used in a social climate context, as the speeches containing them were irrelevant to climate change. In this experiment, it was possible to get good results by focusing on single words only, but even in such a case, it still shows that context plays an important role.

5.1.3 Topic: Feminism

The following subsection gives an evaluation of the results which were obtained using the frequency-based approach when applied to feminism. The Wikipedia article used can be found here [wikb].

The first plot Figure 5.9 shows the amount of relevant and irrelevant speeches retrieved from the corpus. The speeches were manually evaluated on whether they are relevant to the topic or not.

This plot shows that the total amount of retrieved speeches is way lower than the amount of speeches retrieved for the two other topics. A possible reason is the amount of words related to feminism that are very commonly found in the speeches of the national

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

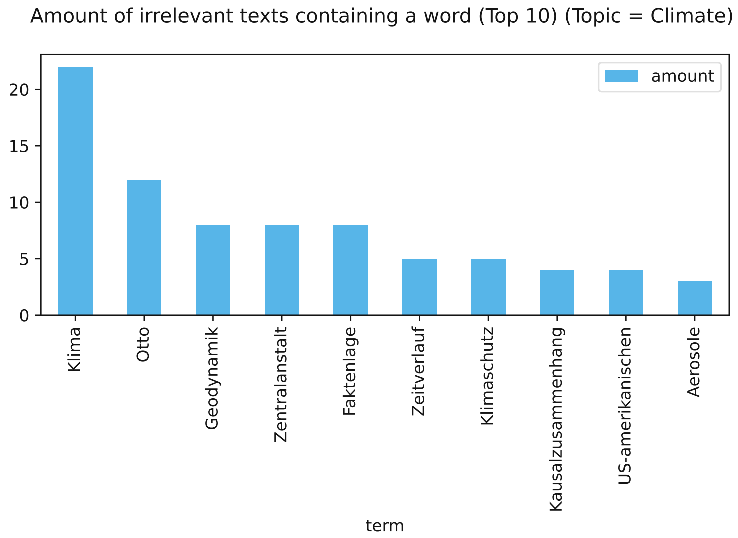


Figure 5.8: Amount of irrelevant texts containing a specific word (Top 10) (Topic = Climate Change)

council, such as "Frau", which appears in sentences like "Wie Frau Muster bereits erwähnt hat" that may appear in any context. These words are excluded from the technical terms as they are above the cut-off frequency threshold. Still, they could be included in highly-relevant speeches as well, if they contain passages like "Rechte der Frau". The general result is not as bad as for the topic "European migrant crisis" but there are still close to twice as many speeches that are irrelevant. The following plots show which technical terms were used to retrieve the speeches.

Figure 5.10 shows the amount of speeches containing a specific technical term, Figure 5.11 shows the amount of relevant speeches containing a specific technical term and Figure 5.12 shows the amount of irrelevant speeches containing a specific technical term. The results are always ordered descending by amount and the first ten entries are displayed in each plot.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

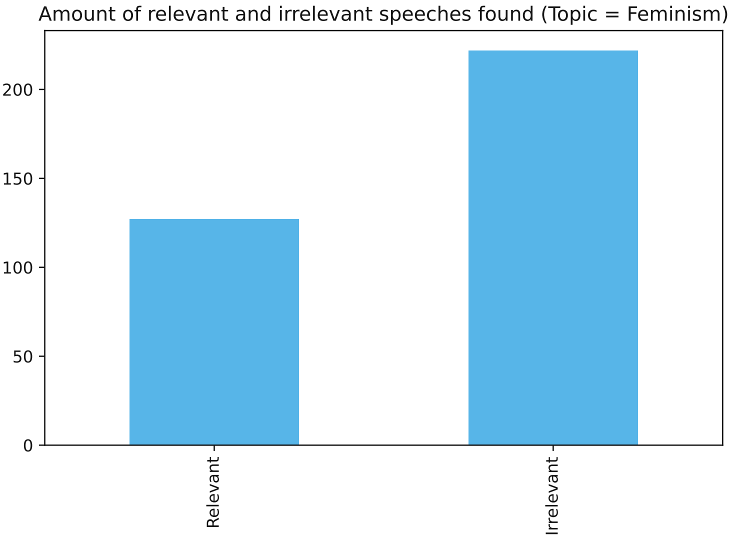


Figure 5.9: Value counts of relevant and irrelevant speeches (Topic = Feminism)

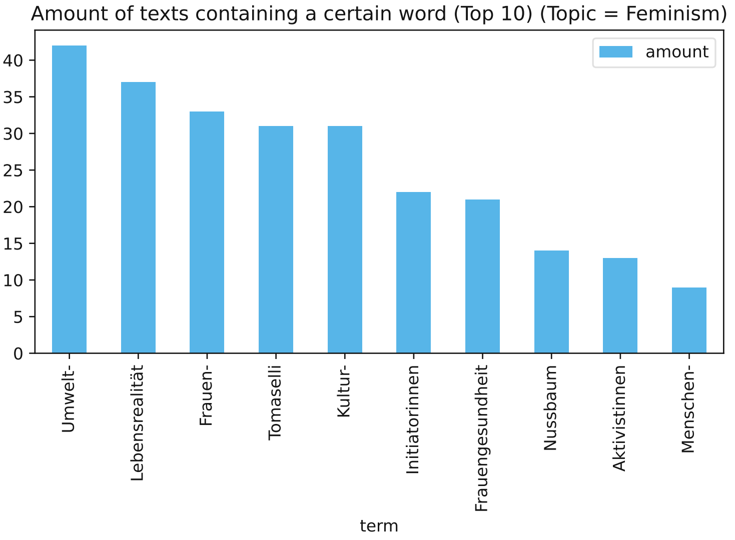


Figure 5.10: Amount of texts containing a specific word (Top 10) (Topic = Feminism)

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

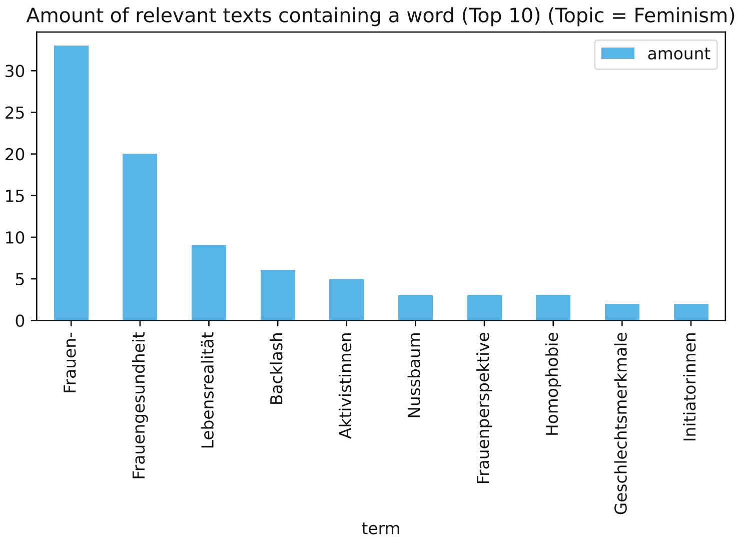


Figure 5.11: Amount of relevant texts containing a specific word (Top 10) (Topic = Feminism)

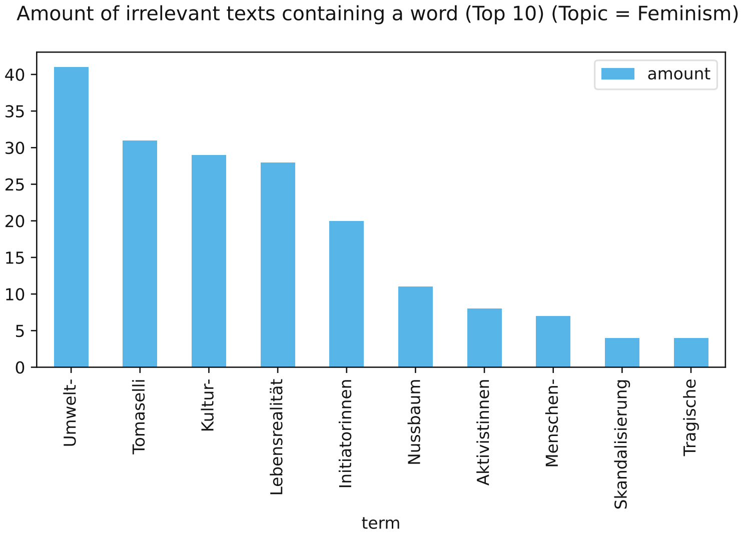


Figure 5.12: Amount of irrelevant texts containing a specific word (Top 10) (Topic = Feminism)

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

5.2 Opinion Type Classification

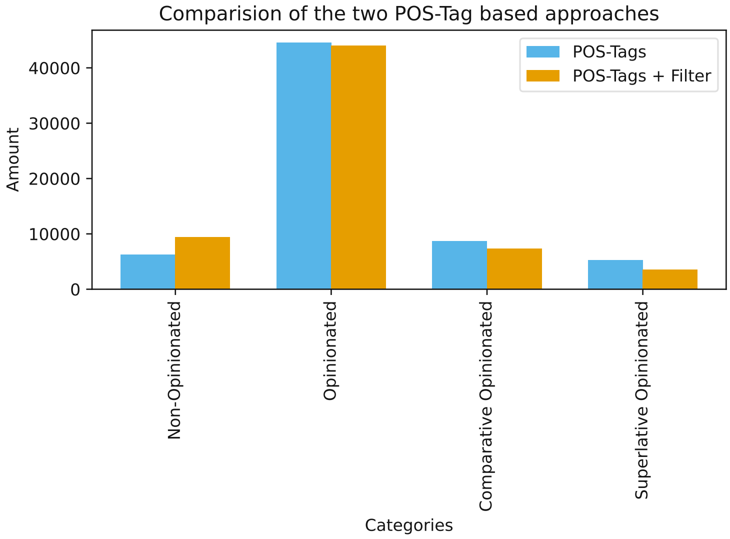


Figure 5.13: Comparison of the results of the two POS-Tag based approaches

This section presents the results that were gathered while implementing the approaches for opinion type classification, which are described in the experiment section above. Two approaches were implemented for detecting opinionated texts and three different approaches were implemented for the detection of comparative and superlative opinionated texts.

To be able to compare the results of the different approaches a manual evaluation was conducted. The goal was to classify the statements that were tagged as opinionated, as either opinionated or not. As there were way to many statements, which were classified as opinionated, only a subset of 200 statements per method was evaluated. The 200 statements were selected randomly. It needs to be said that the manual evaluation introduces a bias as it was only conducted by one person.

Table 5.1 gives a short overview of the precision of the different methods, the results are explained in more detail in the following subsections.

5.2.1 Approach 1: POS-Tagging using the RFTagger

The POS-Tagging approach, which was implemented following the approach by Othman et al. [OHMI15], was implemented for all three opinion types.

Figure 5.13 shows the results (category = "POS-Tags") that were achieved by naively searching for the respective tags. After analyzing the words that were tagged the most often, some words showed up, that would give the impression, that they are only used in greetings. As all of the texts are extracted from parliament protocols, this often is the

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Approach	Precision
POS-Tagging Opinionated	**71%**
Data-Driven Opinionated	67%
POS-Tagging Comparative Opinionated	**44%**
Data-Driven Comparative Opinionated	36%
Rule-Based Comparative Opinionated	33.5%
POS-Tagging Superlative Opinionated	44%
Data-Driven Superlative Opinionated	65%
Rule-Based Superlative Opinionated	**78%**

Table 5.1: Results of the approaches that were manually evaluated

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

case, as many of the texts start with a greeting. To handle this case, another set of words was created. This time, the words were the ones that showed up most often and gave the impression that they are part of a greeting. After that, a filter was implemented with the task of filtering out statements were each tagged word belongs to the aforementioned set. This step was performed for each tag and led to a significant reduction of superlative opinionated statements, which is shown in Figure 5.13 (category = "POS-Tags + Filter").

POS-Tagging: Manual Data Evaluation Figure 5.14 shows the amount of opinionated and non-opinionated statements in the sample containing 200 statements that were classified as opinionated based on the POS-Tagging approach. Around 140 out of 200 statements are truly opinionated, which leads to a precision of around 0.7 - One has to consider that 44578 out of 63909 statements were classified as opinionated by this approach, so 200 statements is a tiny sample in this case.

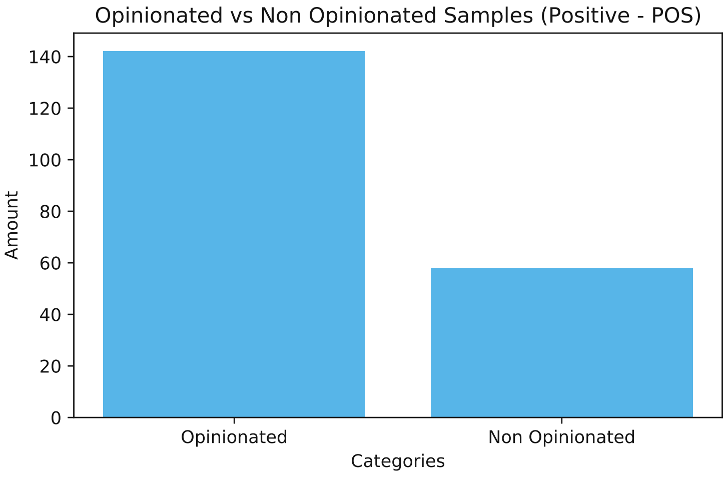


Figure 5.14: Opinionated and Non-Opinionated Statements (POS-Tagging)

Figure 5.15 shows the amount of comparative opinionated statements and non-opinionated statements based on the manual evaluation. Here one can see that there are around 55% non-opinionated statements.

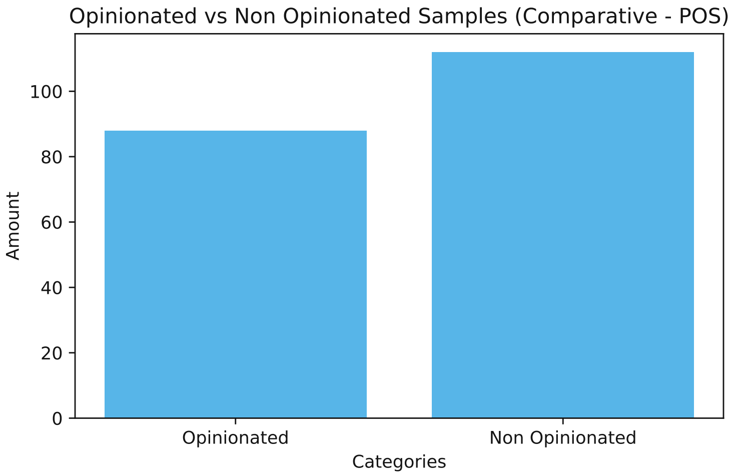


Figure 5.15: Comparative Opinionated and Non-Opinionated Statements (POS-Tagging)

Figure 5.16 which shows the results of the evaluation of the POS-Tagging approach when applied to finding superlative opinionated statements, surprisingly shows exactly the same result as Figure 5.15, where around 55% of the manually evaluated statements are non-opinionated.

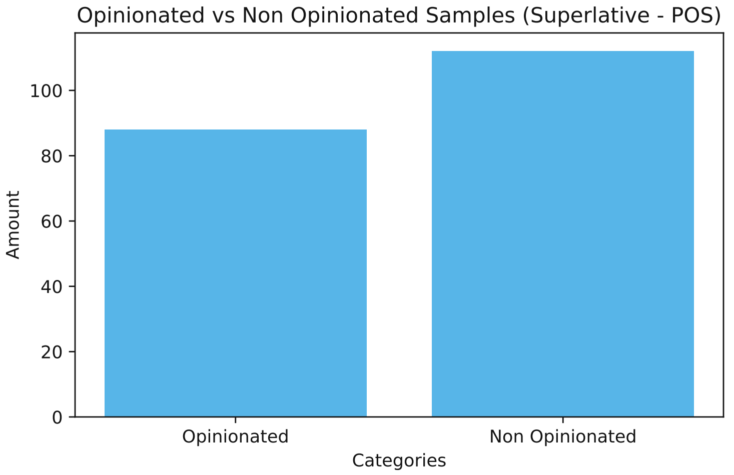


Figure 5.16: Superlative Opinionated and Non-Opinionated Statements (POS-Tagging)

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

5.2.2 Approach 2: Data-Driven Approach

For this approach, a dataset containing adjectives in three forms (positives, comparatives, superlatives) ,if available, was created using the data from Wiktionary [Wikf].

After that, five sets were created. Set P contains all the positive forms from the dataset, Set PD contains all the positive forms and their declensions. Set C contains all comparative forms from the dataset and Set CD contains the comparative forms and their declensions. Set S contains all superlative forms from the dataset. All the sets and their respective amount of words are shown in Table 5.2. The sets of words to ignore, which were defined in approach 1, were considered and the words were removed from the respective sets. Figure 5.17 shows the amount of statements that were marked as opinionated statements using different setups. All of the setups return a list of booleans, therefore it is possible to combine them using logical operators. If only one setup is mentioned (for example Set P) only the amount of statements which were marked as opinionated using Set P are considered. The logical operators used are the *AND operator* $\wedge$ and the *OR operator* $\vee$.

Set Name	Amount of Words
Set P (Positives)	13702
Set S (Superlatives)	5717
Set C (Comparatives)	5705
Set PD (Positives and Declensions)	82170
Set CD (Comparatives and Declensions)	34229

Table 5.2: Sizes of the sets which are used in the data-driven approach

The first observation one can make is that the set-based methods find way more sentences containing an adjective in positive form than the tag-based methods. When combining one of the sets with the results from the tag-based method (without filter) using the logical AND operator the amount is only slightly less than when using the tag-based method (without filter) alone. On the other hand, when combined using a logical OR operator, the amount of statement increases significantly.

Figure 5.18 shows the amount of statements that were marked as comparative opinionated statements using the same methods as described above but with the tags ADJD.Comp and ADJA.Comp and using the two sets Set C and Set CD which were generated using the Wiktionary dataset as well.

In this case the set-based methods find way more statements than the tag-based methods, but once they are combined with an AND operator they return less statements although both sets contain a significant amount of comparative forms. This is a similar result to the one that was observed in Figure 5.17.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

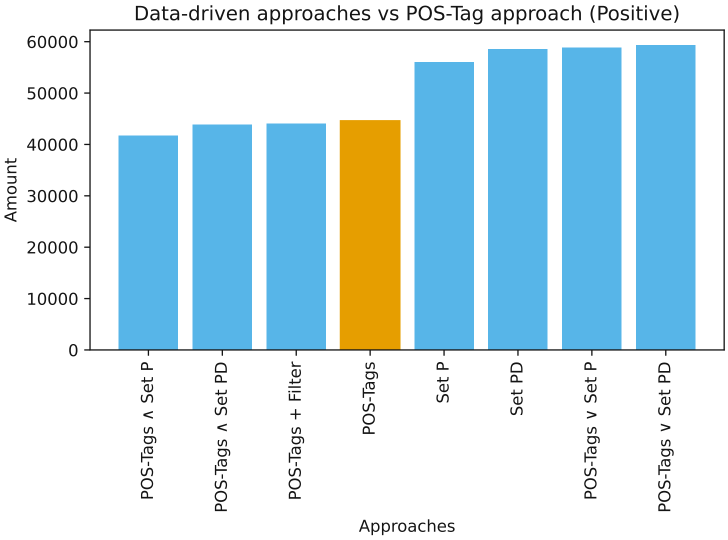


Figure 5.17: Results of the data-driven approach (positives)

Figure 5.19 shows the results that were achieved when applying the same approach to find superlative words. Here, the set-based method finds way less statements than the tag-based method. This may be the case because the tag-based method marked a lot of words as superlatives where the context reveals that it is not used to express an opinion. Examples are 'nächster', 'nächstes' and 'Hochgeschätzter'. All of these words are excluded from Set S.

Data-Driven: Manual Data Evaluation The results shown in Figure 5.20 are very similar to the ones shown in Figure 5.14, so in this regard the POS-Tagging approach and the data-driven approach seem to perform quite similar.

Figure 5.21 shows similar results to Figure 5.15 with a slight increment in non-opinionated statements.

One can see in Figure 5.22 that this approach performs better than the POS-Tagging approach (Figure 5.16). This might be related to the pre-filtering that has already been done, where some obvious words like "nächsten" were removed from the set before classifying the statements.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

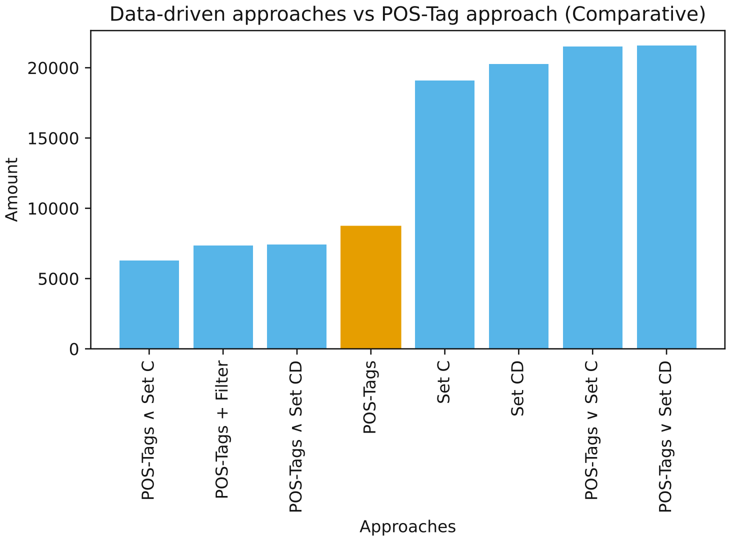


Figure 5.18: Results of the data-driven approach (comparative)

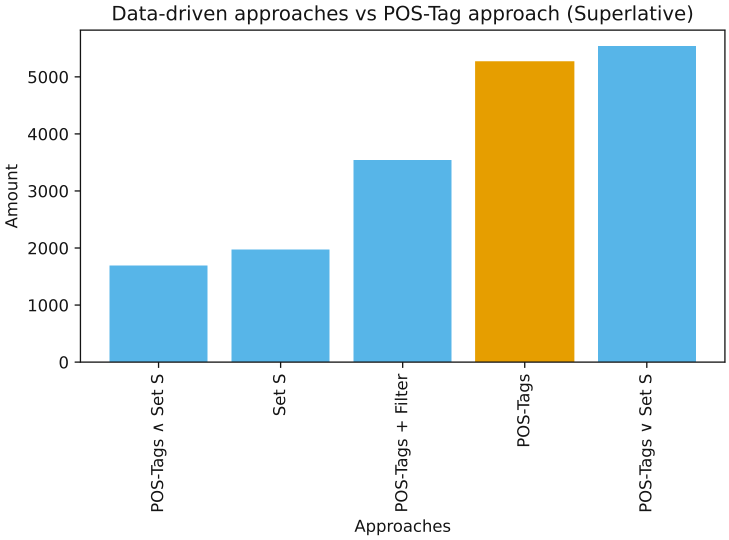


Figure 5.19: Results of the data-driven approach (superlative)

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

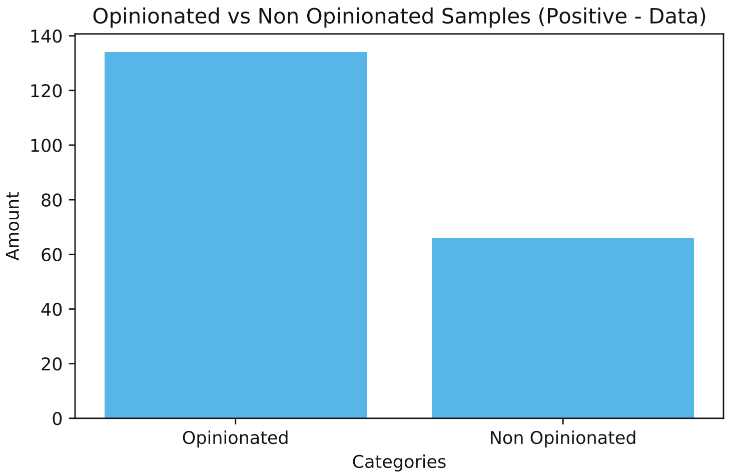


Figure 5.20: Opinionated and Non-Opinionated Statements (Data-Driven)

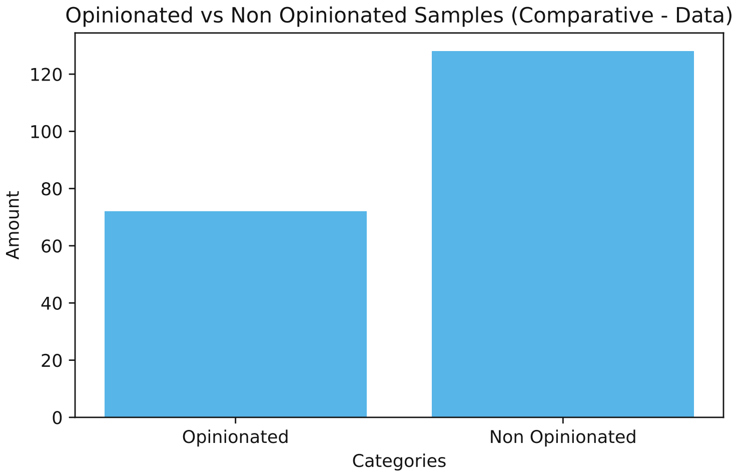


Figure 5.21: Comparative Opinionated and Non-Opinionated Statements (Data-Driven)

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

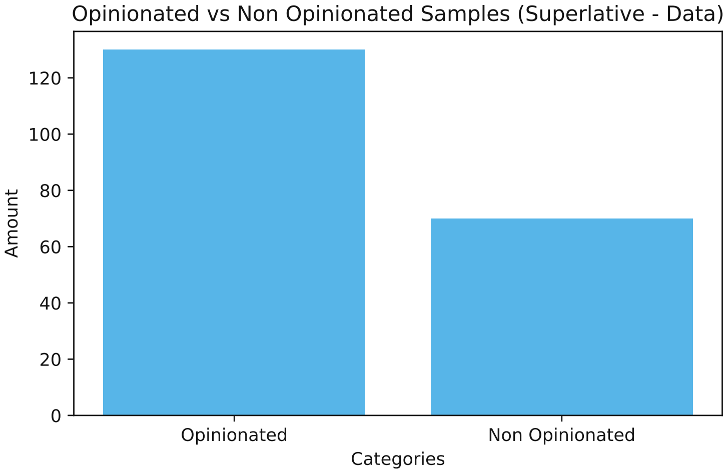


Figure 5.22: Superlative Opinionated and Non-Opinionated Statements (Data-Driven)

5.2.3 Approach 3: Rule-Based Approach with regular expressions

The rule-based approach was only applied for finding comparative or superlative statements. All the rules, which can be found in Table 4.5, were implemented using regular expressions.

One important constraint was added while experimenting with the rule-based approach. There was a problem with the rules considering "immer ...er" and "...er als", as this rule also gets triggered in cases like "immer wieder" and "wieder als" where it would not count as comparative. To prevent this from happening, the procedure was extended so that one of the sets containing comparatives (Set C or Set CD) can be added. The String which is found using the regular expression is then separated and a lookup is performed to find out whether the word is part of the set. If that is the case it is marked as a comparative, else it is ignored.

They are again compared to the tag-based method. Figure 5.23 shows the results of the different rule-based approaches that were executed to find comparative statements. The discrepancy between the different approaches is noteworthy; the tag-based approach finds around 8600 statements, while the rule-based approach with the set-based constraints finds below 2000 statements in both cases. When both approaches are combined with an AND operator, below 800 statements are found in both cases.

Figure 5.24 shows the results of the rule-based approach applied to finding superlative statements. The tag-based approach again finds significantly more superlative statements

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

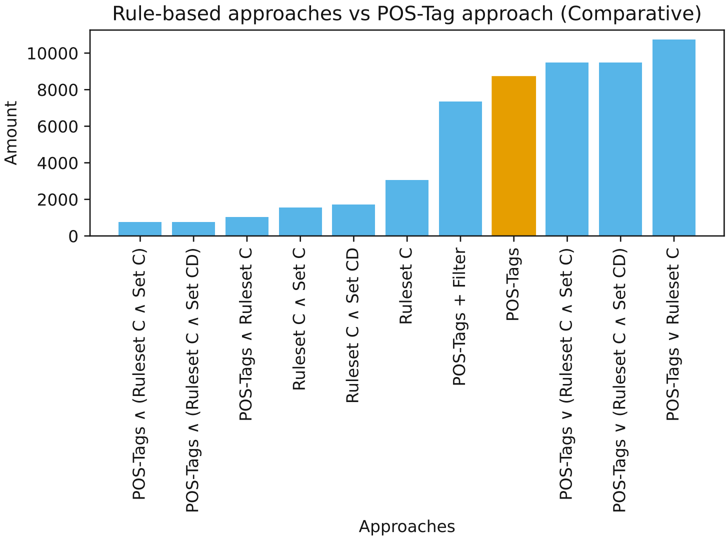


Figure 5.23: Results of the rule-based approach (comparative)

than the rule-based approach. Curiously, the rule-based approach finds superlative statements that were not found using the tag-based approach, as there are 477 in total but only 383 when they are combined with the results of the tag-based approach using the AND operator.

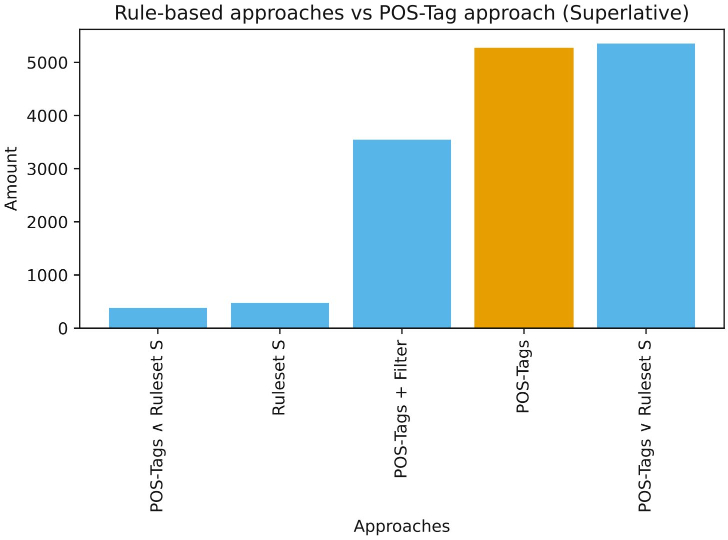


Figure 5.24: Results of the rule-based approach (superlative)

Rule-Based: Manual Data Evaluation Figure 5.25 shows that this rule-based approach performed slightly worse than the other two approaches. Here one needs to

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

consider that this is the approach without additional filters.

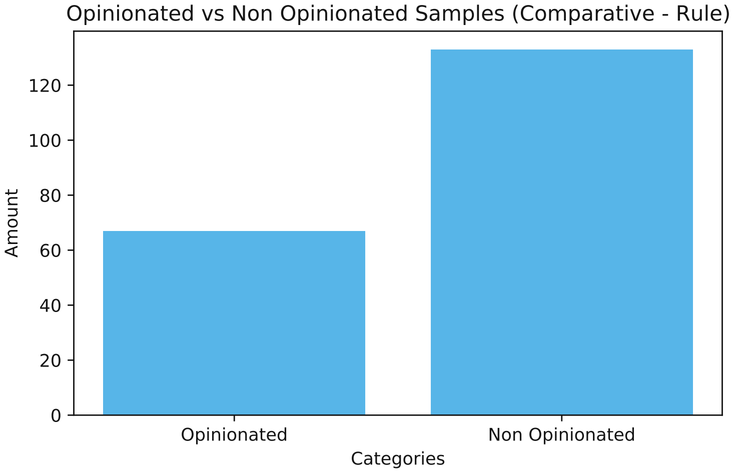


Figure 5.25: Comparative Opinionated and Non-Opinionated Statements (Rule-Based)

Figure 5.26 shows that the performance using this approach is better than the performance achieved by the POS-Tagging approach and the data-driven approach as shown in Figure 5.16 and Figure 5.22.

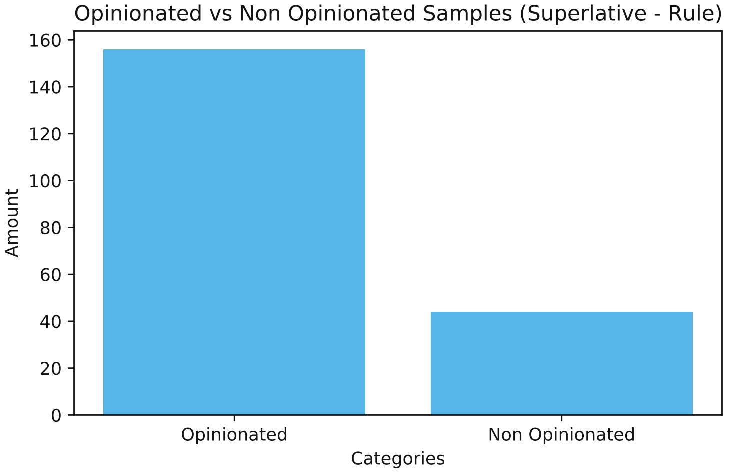


Figure 5.26: Superlative Opinionated and Non-Opinionated Statements (Rule-Based)

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

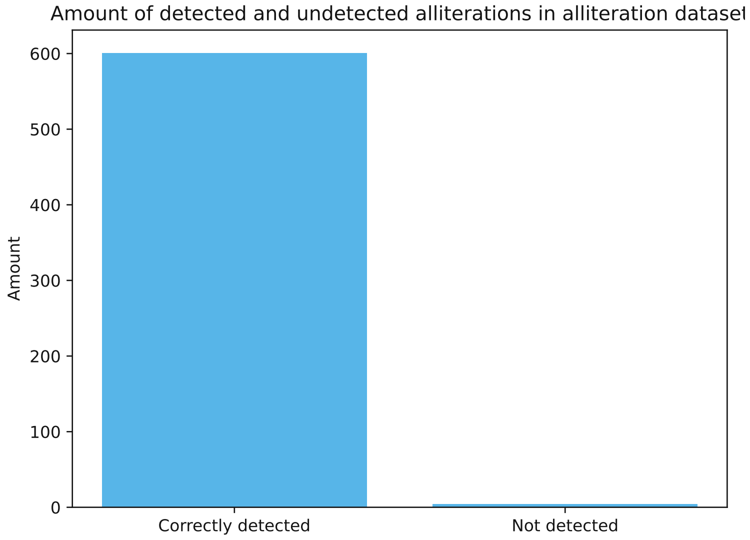


Figure 5.27: Amount of detected and undetected alliterations in the alliteration dataset

5.3 Alliteration Detection

This section presents the results that were gathered while conducting the experiments regarding the detection of alliterations, which are described in the experiments section above.

Two different datasets were used to evaluate the algorithm, one of them consists of alliterations and one consists of political statements. The first one was used to evaluate whether the algorithm is able to detect correct alliterations.

The second dataset was used to see whether the algorithm can detect alliterations in plain text. A subset of the results were manually annotated as there is no information about whether there are alliterations in the dataset or not. It was not possible to manually annotate all the results as the algorithm detected way to many cases, although the configuration Table 4.8 was strict compared to the configuration defined for the alliteration dataset Table 4.7

5.3.1 Alliteration Dataset

Figure 5.27 shows the results that were achieved when using the detection algorithm on the alliterations dataset. In this case the performance was very good as 601 out of 605 alliterations were detected. The only alliterations that were not detected are four examples of alliteration by word combination, as the algorithm does not consider the possibility that there is an alliteration in a single word. The four examples are "habhaft", "Linkliste", "Wendewinkel" and "Diridari".

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

5.3.2 Political Statement Dataset

The following plots show the results of the experiments using the dataset consisting of political statements.

The results are separated into four different sets:

- standard
- cologne phonetics
- both
- none

The algorithm was configured so that it would return one of four results per statement. The four different options are standard, cologne phonetics, both or none. If the algorithm returns "standard" it means that the alliteration was detected based on the first letter only. The option "cologne phonetics" stands for alliterations that were only detected because of the numeric representation of the letters. The other two options are either both of the previous ones or none of them. 200 statements were manually evaluated for the sets "standard", "cologne phonetics" and "both".

Figure 5.28 shows the overall results. With the hard restriction that an alliteration must contain at least four words without any steps in between the amount of detected alliterations went down drastically, without the restriction regarding the steps in between around 22.000 statements were marked as containing an alliteration.

During the manual evaluation of the first of the three sets a few edge cases showed up, they were invalid alliterations but the algorithm would detect them as alliterations. After that few rules were defined, which were considered when deciding whether an alliteration is valid or not, they are listed below. The algorithm could be improved by implementing logic which could deal with these cases. The examples are all taken from the political statements.

- An alliteration should only be considered if it is inside of a single sentence
    - Example: **e**inen **E**ntschließungsantrag **e**in**.** **E**r lautet
- If there are words which are separated by hyphen they should be considered separately
    - Example: aller **AMA-G**ütesiegelprodukte ausschließlich auf
- The initial sound of the words must not be different, even if the first letter is the same

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

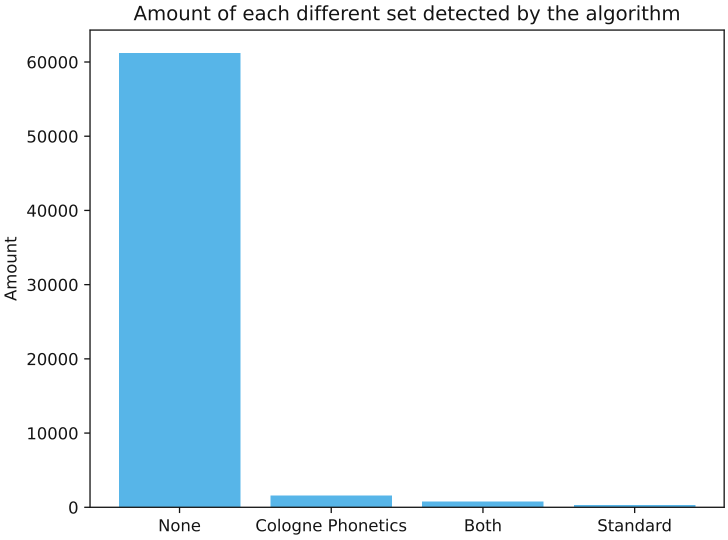


Figure 5.28: Amount of each of the sets detected by the algorithm in the political statements dataset

  - Example: nicht nur **s**ich **s**elbst **sch**ützt, **s**ondern
  - Example: **es** **ei**n **e**xtrem **e**rfreuliches
- Repetitions should not be considered
  - Example: in Österreich zugelassen **sind**, **sind** **s**icher, **sind**
  - Example: **v**ollkommen **v**erständlich, **v**ollkommen **v**erständlich
- Alliterations consisting of one unique word only should not be considered
  - Example: impfen, impfen, impfen, impfen
  - Example: Ja, ja, ja, ja

5.3.2.1 Standard

Figure 5.29 shows the results of the manual evaluation of the statements where an alliteration was found based on the standard method. In this case around 130 out of 200 cases were valid alliterations. The rules defined above were considered, they were responsible for the invalid cases.

The most common sets of letters that occurred in the standard set are shown in Figure 5.30. The set of letters for the detected alliteration was computed in the algorithm and appended to the dataset so that it can be used for the evaluation. Here it is interesting to see, that there is only a tiny amount of sets, containing the letters a, e, i, u, h and j. The reason for that is that code 0 and code - (from the cologne phonetics transformation table

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

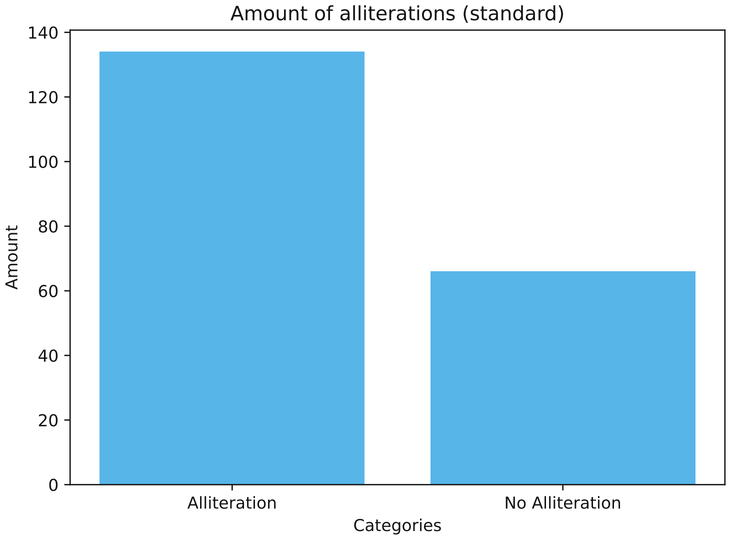


Figure 5.29: Amount of detected and undetected alliterations in the political statement dataset (standard)

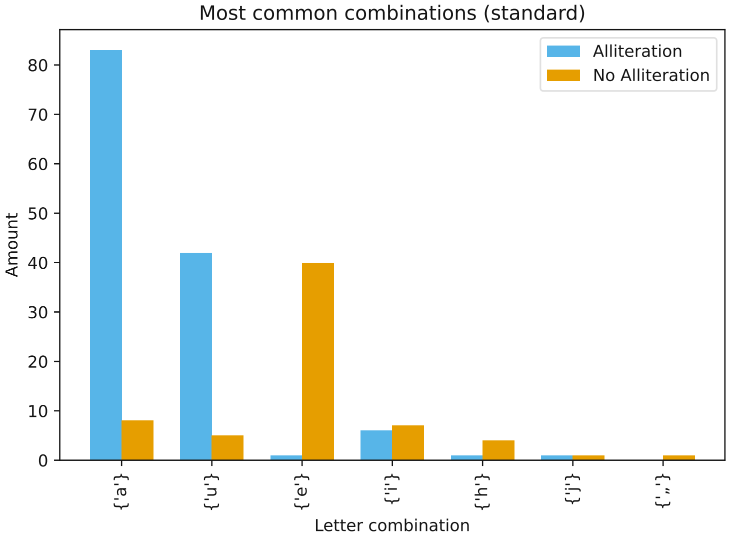


Figure 5.30: Most common sets of letters (standard)

Table 4.6) were ignored as they cannot be used to detect alliterations. The alliterations based on words starting with one of these letters were the only ones that only showed up using the standard method, where only the initial letters were combined. All the others either showed up with cologne phonetics only or with both ways.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

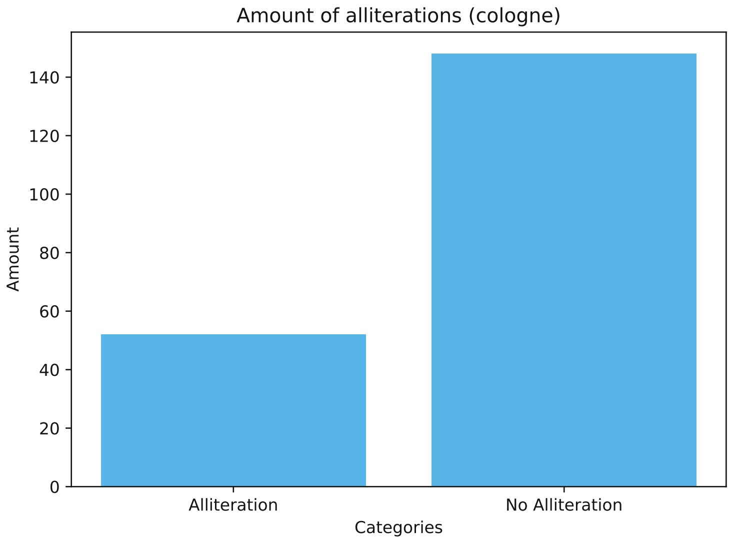


Figure 5.31: Amount of detected and undetected alliterations in the political statement dataset (cologne)

5.3.2.2 Cologne Phonetics

Figure 5.31 shows the results of the manual evaluation of the statements where an alliteration was found based on the cologne phonetics method. In this case more than 140 out of 200 cases were invalid alliterations, which is more than two thirds and a bad result compared to the results of the standard method. The following list presents negative and positive examples for a few of the transformation rules which are defined in Table 4.6, these examples might lead to the conclusion that a similar algorithm like cologne phonetics could be developed, which focuses on alliterations only, but it might turn out to be very difficult to find the full set of rules that needs to be considered. The examples are taken from the political statements.

- Negative examples for rule number 3 (f, v, w and p if before h)
  - Example: fangen wir wieder von vorne an
  - Example: vor Wahlen wieder Wahlzuckerl verteilen wollen
- Positive examples for rule number 3 (f, v, w and p if before h)
  - Example: für viele Frauen, für viele Familien
  - Example: Fremdübernahmen von Firmen vollzogen
- Negative example for rule number 8 (Table 4.6)
  - sich zum Ziel setzt

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

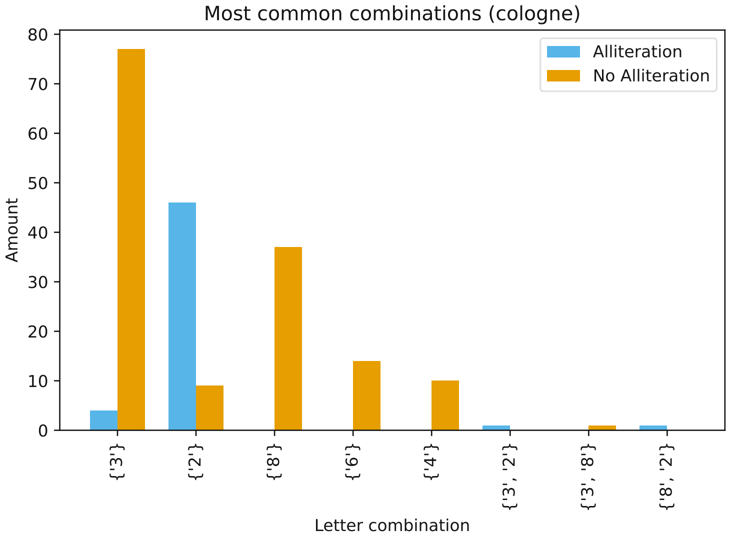


Figure 5.32: Most common sets of numbers (cologne)

- dass der Thinktank Think

Figure 5.32 shows the most common sets of numbers that showed up in the statements. It is interesting to see that only rule 2 led to valid alliterations in most of the cases, while all of the other significant ones led to invalid alliterations in most of the cases.

5.3.2.3 Both

The results of the manual evaluation of the statements where an alliteration was found based on the cologne phonetics method and the standard method are shown in Figure 5.33. The performance in this case is comparable to the performance of the standard method.

The plot in Figure 5.34 shows the most common letter + number sets and the amount of valid and invalid alliterations. Here it is interesting to see that there are so many invalid alliterations based on the letter "s", they are mostly related to the rule "The initial sound of the words must not be different, even if the first letter is the same", as the first number of the cologne phonetics encoding does not change if the letter "s" is in front of "p", "t" or "ch". One solution could be the consideration of multiple numbers of the cologne phonetics encoding, not just the first one.

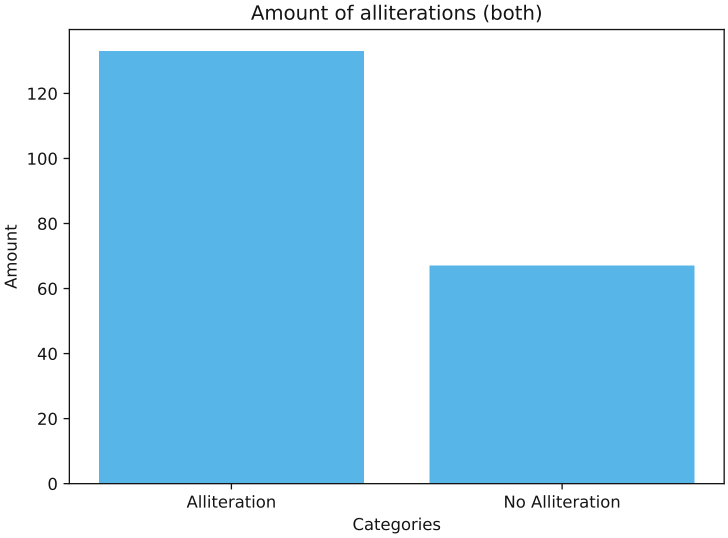


Figure 5.33: Amount of detected and undetected alliterations in the political statement dataset (both)

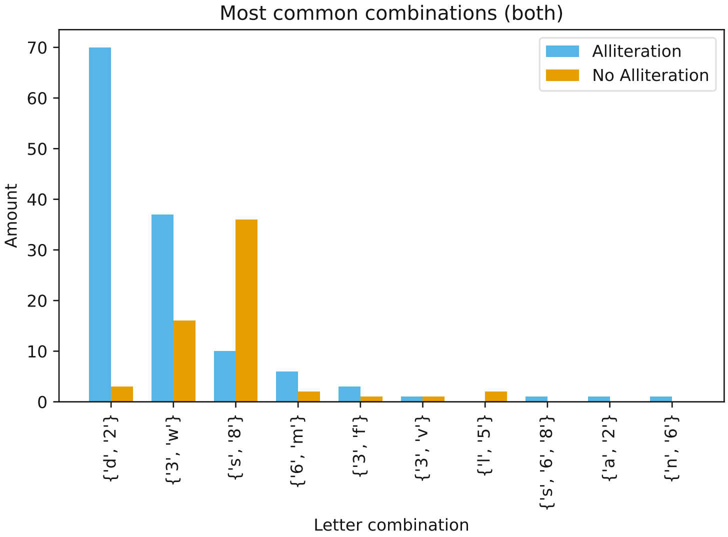


Figure 5.34: Most common sets of numbers (both)

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

	machine all	**machine 400**	**manually 400**
LR	0.689092	0.6350	0.6700
KNN	0.644471	0.5875	0.6375
NB	0.667395	0.6425	0.6400
DT	0.600736	0.6300	0.5975
SVM	0.686295	**0.6675**	**0.6875**
LDA	**0.689097**	0.6375	0.6675
RF	0.668928	**0.6675**	0.6525

Table 5.3: Mean accuracy which was achieved with 10-fold cross validation

5.4 Hyperbole Detection

This section presents the results that were gathered while conduction the experiments regarding hyperbole detection, which are described in the experiments section above.

Two different kinds of datasets were used to evaluate the approach. The first kind of dataset was a translated version of the HYPO-L dataset. The HYPO-L dataset was created by Zhang et al. [ZW21]. The whole dataset was translated using the Google Translate function provided in Google Sheets. 400 sentences of the HYPO-L dataset were manually translated as well.

The experiments were conducted using three different datasets: The whole machine translated HYPO-L dataset, the subset with the 400 manually translated sentences and the same subset but with the respective machine translation, so that the performance can be compared.

The second kind of dataset contained political statements which were extracted from a protocol of the Austrian national council.

5.4.1 Translated HYPO-L dataset

This subsection describes the results that were achieved when using the whole machine translated HYPO-L dataset and the two subsets (machine translated and manually translated).

Table 5.3 shows the mean accuracy that was achieved by the different algorithms on the three different datasets when applying 10-fold cross validation. It is interesting to see that Support Vector Machine (SVM) had the best performance in two cases and was also quite close to the best performance in the third case. Generally all of the algorithms had a mean accuracy above 0.58 which is slightly better than random. The best mean accuracy that was achieved by Troiano et al. [TSÖT18] is 0.72, so the results are not as good but still in a 3% range in the best case (0.689 LDA).

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

	machine all	**machine 400**	**manually 400**
LR	**0.522323**	0.632866	0.659049
KNN	0.395776	0.582831	0.626145
NB	0.464290	0.657486	0.621646
DT	0.366243	0.635871	0.603195
SVM	0.496337	**0.692868**	**0.672423**
LDA	0.520421	0.641219	0.651457
RF	0.443829	0.664046	0.657090

Table 5.4: Mean precision which was achieved with 10-fold cross validation

	machine all	**machine 400**	**manually 400**
LR	0.099307	0.635	0.725
KNN	0.263139	0.630	0.670
NB	0.370327	0.595	0.715
DT	**0.385228**	0.610	0.610
SVM	0.087416	0.605	**0.740**
LDA	0.123168	0.615	**0.740**
RF	0.242198	**0.685**	0.655

Table 5.5: Mean recall which was achieved with 10-fold cross validation

The data in Table 5.4 shows the mean precision that was achieved when applying 10-fold cross validation. One can see that the performance on the full machine translated HYPO-L dataset was rather low with 0.52 being the best value. The mean precision was way higher in the other two cases, with 0.69 and 0.67 as the highest values, but this could also be due to the smaller size of the dataset. It is interesting to see, that SVM, which had the highest mean precision, performs better on the machine translated subset then on the manually translated subset. The best mean precision that was achieved by Troiano et al. was 0.76, so in that case the results are not comparable.

The next table, Table 5.5, shows the mean recall that was achieved when applying 10-fold cross validation. The performance on the full machine translated HYPO-L dataset was quite bad, with 0.385 being the highest mean recall. SVM and LDA both had a quite good mean recall on the manually translated subset though (0.74). When comparing the results to the best mean recall that was achieved by Troiano et al. (0.76) they are only comparable when considering the performance when using the manually translated dataset, in the two other cases, especially when using the full machine translated dataset, they are not comparable.

Table 5.6 shows the mean f1-score that was achieved when applying 10-fold cross validation. The performance on the full machine translated dataset is low with 0.41 being the highest mean f1-score. When using the two subsets the highest mean f1-score was moderately high, with 0.67 and 0.70 being the best values. The best mean f1-score that was achieved by Troiano et al. was 0.76 - when looking at the performance on the full machine

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

	machine all	**machine 400**	**manually 400**
LR	0.166267	0.632089	0.688706
KNN	0.315328	0.603822	0.645403
NB	**0.411136**	0.621979	0.661793
DT	0.375204	0.619045	0.604268
SVM	0.147001	0.643456	**0.703524**
LDA	0.198310	0.625849	0.689583
RF	0.312876	**0.673094**	0.653529

Table 5.6: Mean F1-score which was achieved with 10-fold cross validation

translated dataset the performance is not comparable.

5.4.2 Political Statements Dataset

This subsection describes the results that were achieved when using the political statements dataset. Random Forest (RF) was used as an algorithm. The model got trained on the whole machine translated HYPO-L dataset using the semantic features described above. The feature engineered version of the political statements dataset was then used as a test dataset.

The approach was to manually annotate all the statements that were classified as being hyperbolic, so that the precision could be calculated.

Figure 5.35 shows the amount of statements that were classified as either literal or hyperbolic when using the model that was trained on the full machine translated HYPO-L dataset. 98 Statements were classified as hyperbolic, 1229 were classified as literal. The hyperbolic statements were manually evaluated so that the precision of the algorithm could be calculated.

The plot in Figure 5.36 shows the amount of hyperboles that were found based on the manual evaluation. As only one person was evaluating the statements the result contains a bias. 22 Statements were classified as being hyperbolic, the other 76 statements were classified as literal. This leads to a precision of 22%. The following paragraphs will show a few interesting examples for hyperboles and literal statements that were classified as being hyperbolic.

The following list will show a few examples that are hyperboles according to the annotator, the reasoning behind the decisions will be explained below.

1. Da hat anscheinend alleine das Einbringen der Petition beim Gesundheitsminister Wunder bewirkt.

2. Wer will, dass die Welt so bleibt, wie sie jetzt ist, der will nicht, dass sie bleibt.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

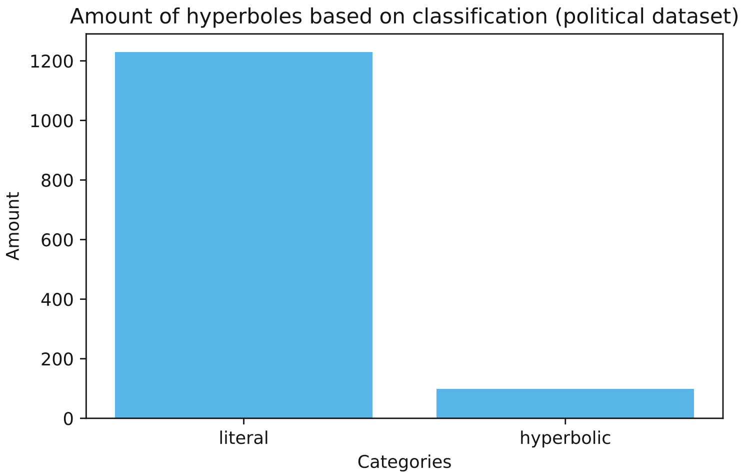


Figure 5.35: Amount of hyperboles based on the classification (political dataset)

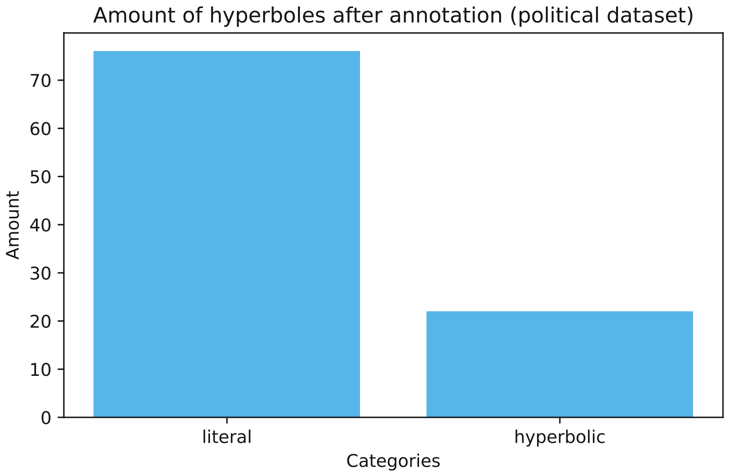


Figure 5.36: Amount of hyperboles based on the manual evaluation (political dataset)

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

3. Zu Tode gefürchtet ist auch gestorben!
4. Den Menschen habt ihr die Zuversicht genommen, die sind alle depressiv, und schuld daran sind unter anderem diese Masken, eure Verordnungen, bei denen sich keiner auskennt.
5. Die Menschen laufen alle nur mehr apathisch herum.
6. Statt Mut impft ihr den Leuten Angst ein.
7. Wer heute einmal keine Maske trägt dafür kann er eine medizinische Begründung haben oder nicht , ist sowieso schon böse.
8. Das ist eine schreiende Ungerechtigkeit.

Statement 1 claims that handing in a petition worked wonders, with is interpreted as being hyperbolic as it did not really cause any wonders. It is an idiom that might have been used to put more weight on the act of handing in a petition.

Statement 2 seems to be a bit of an exaggeration as it expresses the thought that people who want the world to stay the way it is now would not mind if it would not exist at all anymore.

Statement 3 is a quote by Johann Nestroy that was used in one of the speeches. The literal meaning is that being feared to death is the same as being dead, the interpretation in the context of this work is that having a lot of fear in the daily life could be seen as being dead as well. Using this interpretation the statements could be seen as hyperbolic.

Statement 4 claims that the people which are being talked to by the speaker took all the hope from the people, potentially by implementing some new guidelines like the lockdown during the COVID-19 pandemic, and that all the people are depressive. This is interpreted as an exaggeration as there are no statistics or studies claiming that 100% of people in Austria were depressive.

Statement 5 is similar to statement 4 as it claims that all people are running around apathetic, which is interpreted as an exaggeration as well.

Statement 6 claims that someone is injecting fear into people's minds. The word "impfen" is most commonly used as a medical term (vaccinate), therefore this is interpreted as being metaphorical and exaggerated.

The speaker who included statement 7 in their speech claims that there are people who think that if someone who does not wear a surgical mask, most likely during the COVID-19 pandemic, is automatically evil, does not matter whether they have medical reasons or not. The part "ist sowieso schon böse" is interpreted as a slight exaggeration,

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

as it might convey the idea that this is the general opinion, which is not only expressed by some people but actually the majority of people.

Statement 8 is interpreted as an exaggeration as "schreiende Ungerechtigkeit" stands for "screaming injustice", which puts stronger emphasis on the injustice.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

CHAPTER 6

Conclusion

During the experiments that were conducted in the scope of this master thesis multiple approaches for analyzing political statements were explored, one being relevant to the task of topic classification, one that focuses on opinion type classification and two different types of figure of speech detection, the detection of alliterations and hyperboles. All of these approaches were implemented for the German language. The insights that were gathered as a result of the experiments allow the following research questions to be answered:

RQ1: *How high is the precision of a rule-based approach for topic classification in political statements using corpora extracted from Wikipedia?*

The results of the experiments using three different topics (climate change, feminism, European migrant crisis) show that the precision can vary in a strong way between the different topics. In the case of climate change the approach showed promising results with a precision of 89,03%, in the case of feminism the precision was 36,39% and in the case of the European migrant crisis the precision was 19,04%. Following the experiments a data analysis was conducted which showed some of the underlying problems.

RQ2: *How does an approach using a German tag set and a tagger for the German language, instead of an English tag set and a tagger for the English language, for opinion type classification in German sentences, hold up against the approach developed by Othman et al. regarding opinion type classification in English sentences? This question will be evaluated using precision*

This question was evaluated based on the achieved precision by Othman et al. [OHMI15] regarding the classification of the three opinion types they used (opinion, comparative opinion, superlative opinion). In the case of the standard opinion the precision was comparable, 76,6% precision were achieved by Othman et al. and

71% precision were achieved by this implementation. In the case of the comparative and superlative opinion the results were not comparable. Othman et al. achieved 78,3% precision regarding comparative opinions, this implementation achieved only 44% precision. Superlative opinions were classified with 82,1% precision by Othman et al. while this implementation only achieved a precision of 44%. The two additional approaches that were implemented and the data analysis provided interesting insights regarding the improvement of this implementation.

RQ3: *The Cologne phonetics algorithm can be used to transform a word into a numerical representation depicting the underlying phonetics. Which additional constraints need to be added so that the numerical representation resulting from the Cologne phonetics algorithm is feasible for alliteration detection with 95 percent precision?*

The algorithm which was implemented for detecting alliterations achieved a very high precision (99,33%) on the alliteration dataset by Ulrich Mehner [meh] which contains 605 alliterations. The successful setup combined a phonetic approach using the Cologne Phonetics algorithm with a simple approach which is based on the words starting with the same letter. In addition the algorithm was written in a way so that it is flexible regarding gaps between words and the length of an alliteration. These constraints allow the detection of alliterations like "**B**rot und **B**utter" and "Der **f**rühe **V**ogel gibt **f**rohe Töne **v**on sich".

Additional experiments were performed on the political statements dataset, the three different setups that were tested and manually evaluated achieved lower precision (65%, 30% and 66,5%) and provided interesting insights into additional constraints that can be added and tested in future work.

RQ4: *Troiano et al. developed an approach for classifying English sentences as either hyperbolic or not. Is this approach applicable to the German language as well? This question will be evaluated using accuracy, precision, recall and the F1-Score.*

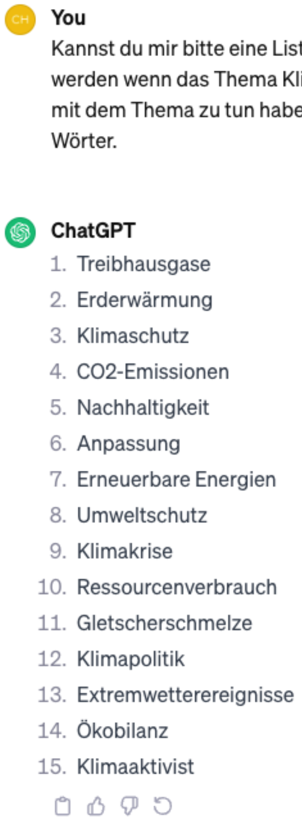
To answer this question, the same approach that Troiano et al. [TSÖT18] implemented for the English language was implemented for the German language. Precision, recall, accuracy and F1-Score were calculated based on the achieved results and compared to the best results achieved by Troiano et al. [TSÖT18] across their experiments. The accuracy is comparable, the best results are between 66,75% and 68,90%, while the best result by Troiano et al. is 72%. In the three other cases the best results are not comparable, especially when using the large machine translated version of the HYPO-L dataset (the original dataset was published by Zhang et al. [ZW21]), in this case the best results for precision (52,23%), recall (38,52%) and F1-Score (41,11%) were far below the best results that were achieved by Troiano et al. [TSÖT18] (precision = 76%, recall = 76%, F1-Score = 76%).

Future Work As a result of the data analysis that was performed during the evaluation of the experiments multiple interesting factors were found that could be made use of in

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

future work.

Regarding topic classification it would be very interesting to see the effect that a large language model like GPT-4 [AAA$^+$23] would have when being used instead of the method that was implemented for extracting topic-related words from a Wikipedia article. The general approach with the inverted index could still be used but the topic-related words could be generated using a tool like the ChatGPT API. A tiny example which is displayed in Figure 6.1 shows the potential of this approach, this could lead to a lot of time being saved when searching for political statements regarding a certain topic.



You

Kannst du mir bitte eine Liste von Wörtern (15 Stück) generieren die oft verwendet werden wenn das Thema Klimawandel diskutiert wird? Es sollen Wörter sein, die direkt mit dem Thema zu tun haben. Bitte generiere keine Beschreibung sondern nur die Wörter.

ChatGPT

1. Treibhausgase
2. Erderwärmung
3. Klimaschutz
4. CO2-Emissionen
5. Nachhaltigkeit
6. Anpassung
7. Erneuerbare Energien
8. Umweltschutz
9. Klimakrise
10. Ressourcenverbrauch
11. Gletscherschmelze
12. Klimapolitik
13. Extremwetterereignisse
14. Ökobilanz
15. Klimaaktivist

Figure 6.1: A simple example for a prompt that could be used to generate topic-related words using ChatGPT

The evaluation of the experiments that were conducted regarding opinion type classification showed that it might make sense to create a separate list of comparative and superlative forms that are not used for expressing an opinion in many cases, for example the superlative form of "nah" (am nächsten), which was used to speak about a future sitting in most of the political statements that were analyzed. This dataset in combination with a rule-based approach which covers the grammatical rules for comparatives and superlatives and the part-of-speech based approach could lead to a better performance.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

In the case of alliteration detection a few insights were gained during the data analysis of the results that were achieved on free text. One important improvement could be the implementation of a few additional rules like a restriction for the amount of stop words per alliteration and the exclusion of alliterations that contain the same word multiple times. These simple rules could already increase the precision on free text but it would still not be enough to solve the problem related to the phonetic aspect of an alliteration. In this regard it would be interesting to implement a simpler form of the Cologne phonetics algorithm by Postel [Pos69] which focuses on alliterations in the German language.

The results of the hyperbole detection approach could be improved by creating a specific hyperbole dataset for the German language. In addition it might be interesting to distinguish between hyperboles that were spoken, for example in public debates, and hyperboles that were extracted from literature, as they might have different characteristics. ChatGPT could be useful in the regard of the dataset creation as well, as it could be used to generate hyperboles which could later be crowd evaluated as either being a realistic hyperbole example or not.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

List of Figures

2.1 Transformer architecture by Vaswani et al. [VSP+17] 15
2.2 Scaled Dot-Product Attention and Multi-Head Attention by Vaswani et al. [VSP+17] 16
2.3 A simple POS-tagging example in German using the STTS tag set [WSJB17] 23
2.4 Different approaches for sentiment analysis, taken from Wankhade et al. [WRK22] 25
2.5 Overall pre-training and fine-tuning procedures for BERT by Devlin et al. [DCLT18] 26
2.6 Example of an aspect-sentiment hierarchy as described by Afzaal et al. [AUFF19] 29

4.1 Activity diagram visualizing the process that was implemented to extract the technical terms from Wikipedia 42

5.1 Value counts of relevant and irrelevant speeches (Topic = European Migrant Crisis) 62
5.2 Amount of texts containing a specific word (Top 10) (Topic = European Migrant Crisis) 63
5.3 Amount of relevant texts containing a specific word (Top 10) (Topic = European Migrant Crisis) 64
5.4 Amount of irrelevant texts containing a specific word (Top 10) (Topic = European Migrant Crisis) 64
5.5 Value counts of relevant and irrelevant speeches (Topic = Climate Change) 65
5.6 Amount of texts containing a specific word (Top 10) (Topic = Climate Change) 65
5.7 Amount of relevant texts containing a specific word (Top 10) (Topic = Climate Change) 66
5.8 Amount of irrelevant texts containing a specific word (Top 10) (Topic = Climate Change) 67
5.9 Value counts of relevant and irrelevant speeches (Topic = Feminism) . . . 68
5.10 Amount of texts containing a specific word (Top 10) (Topic = Feminism) 68
5.11 Amount of relevant texts containing a specific word (Top 10) (Topic = Feminism) 69
5.12 Amount of irrelevant texts containing a specific word (Top 10) (Topic = Feminism) 69

5.13 Comparison of the results of the two POS-Tag based approaches 70
5.14 Opinionated and Non-Opinionated Statements (POS-Tagging) 71
5.15 Comparative Opinionated and Non-Opinionated Statements (POS-Tagging) 72
5.16 Superlative Opinionated and Non-Opinionated Statements (POS-Tagging) 72
5.17 Results of the data-driven approach (positives) 74
5.18 Results of the data-driven approach (comparative) 75
5.19 Results of the data-driven approach (superlative) 75
5.20 Opinionated and Non-Opinionated Statements (Data-Driven) 76
5.21 Comparative Opinionated and Non-Opinionated Statements (Data-Driven) 76
5.22 Superlative Opinionated and Non-Opinionated Statements (Data-Driven) 77
5.23 Results of the rule-based approach (comparative) 78
5.24 Results of the rule-based approach (superlative) 78
5.25 Comparative Opinionated and Non-Opinionated Statements (Rule-Based) 79
5.26 Superlative Opinionated and Non-Opinionated Statements (Rule-Based) . 79
5.27 Amount of detected and undetected alliterations in the alliteration dataset 80
5.28 Amount of each of the sets detected by the algorithm in the political statements dataset 82
5.29 Amount of detected and undetected alliterations in the political statement dataset (standard) 83
5.30 Most common sets of letters (standard) 83
5.31 Amount of detected and undetected alliterations in the political statement dataset (cologne) 84
5.32 Most common sets of numbers (cologne) 85
5.33 Amount of detected and undetected alliterations in the political statement dataset (both) 86
5.34 Most common sets of numbers (both) 86
5.35 Amount of hyperboles based on the classification (political dataset) 90
5.36 Amount of hyperboles based on the manual evaluation (political dataset) 90

6.1 A simple example for a prompt that could be used to generate topic-related words using ChatGPT 95

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

List of Tables

2.1 Conversion Table from the Soundex algorithm by Russell and Odell [Rob18] 20
2.2 The transformation table of the cologne phonetics algorithm by Postel [Pos69] 21
2.3 POS-Tags from the STTS tagset [WSJB17] with explanation, related to the example 22

3.1 Two examples from the German adjectives dataset 37

4.1 The four different sentimental categories defined by Othman et al. [OHMI15] 44
4.2 Used POS-Tags by Othman et al. [OHMI15] with description 44
4.3 Used POS-Tags in this experiment 45
4.4 Description of the used POS-Tags in this experiment 45
4.5 Rules which are used in the rule-based approach 46
4.6 The transformation table of the cologne phonetics algorithm by Postel [Pos69] 53
4.7 Different setups of the algorithm which were combined in the best attempt on the alliteration dataset 55
4.8 Different setups of the algorithm which were combined to search for alliterations in the dataset of political statements 55

5.1 Results of the approaches that were manually evaluated 71
5.2 Sizes of the sets which are used in the data-driven approach 73
5.3 Mean accuracy which was achieved with 10-fold cross validation 87
5.4 Mean precision which was achieved with 10-fold cross validation 88
5.5 Mean recall which was achieved with 10-fold cross validation 88
5.6 Mean F1-score which was achieved with 10-fold cross validation 89

TU WIEN Bibliothek Your knowledge hub
Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

List of Algorithms

4.1 Creation of a simple inverted index R 40
4.2 Returns a list of sublists, each sublist has size s 49
4.3 Returns preprocessed words from sentence 50
4.4 Returns true if a sentence contains an alliteration, false otherwise . . . 51
4.5 Returns filtered list of sublists 52
4.6 Returns true if one sublist has an alliteration with step st, false otherwise 54

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

TU Bibliothek Your knowledge hub WIEN

Bibliography

[A+18] Charu C Aggarwal et al. Neural networks and deep learning. *Springer*, 10(978):3, 2018.

[AAA+23] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[AUFF19] Muhammad Afzaal, Muhammad Usman, Alvis CM Fong, and Simon Fong. Multiaspect-based opinion classification model for tourist reviews. *Expert Systems*, 36(2):e12371, 2019.

[aus] Austria national council: Protocols. `https://www.parlament.gv.at/recherchieren/protokolle/`. Accessed: 2023-10-05.

[BBFZ09] Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. The wacky wide web: a collection of very large linguistically processed web-crawled corpora. *Language resources and evaluation*, 43:209–226, 2009.

[BDH+02] Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. The tiger treebank. In *Proceedings of the workshop on treebanks and linguistic theories*, volume 168, pages 24–41, 2002.

[ber] Jacob devlin: Bert github repository. `https://github.com/google-research/bert`. Accessed: 2024-02-21.

[Ber03] Robert Berwick. An idiot's guide to support vector machines (svms). *Retrieved on October*, 21:2011, 2003.

[blo] Google blog article: Understanding searches better than ever before. `https://blog.google/products/search/search-language-understanding-bert/`. Accessed: 2024-02-21.

[Bre96] Leo Breiman. Bagging predictors. *Machine learning*, 24:123–140, 1996.

[Bre01] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[cle] Checkthat! lab at clef 2023. `https://checkthat.gitlab.io/clef2023/task2/`. Accessed: 2023-10-30.

[CSQH17] Xinchi Chen, Zhan Shi, Xipeng Qiu, and Xuanjing Huang. Adversarial multi-criteria learning for chinese word segmentation. *arXiv preprint arXiv:1704.07556*, 2017.

[CV95] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20:273–297, 1995.

[CWB$^+$11] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(ARTICLE):2493–2537, 2011.

[DBB16] Derick F Davis, Rajesh Bagchi, and Lauren G Block. Alliteration alters: Phonetic overlap in promotional messages influences evaluations and choice. *Journal of Retailing*, 92:1–12, 2016.

[DCLT18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[deWa] Website that describes the dewac corpus. `https://www.sketchengine.eu/dewac-german-corpus/`. Accessed: 2023-03-23.

[deWb] Website that provides the frequency list. `https://wacky.sslmit.unibo.it/doku.php?id=frequency_lists`. Accessed: 2023-03-23.

[emb] Embeddings described on deepset.ai. `https://www.deepset.ai/german-word-embeddings`. Accessed: 2023-10-27.

[FE13] Gertrud Faaß and Kerstin Eckart. Sdewac–a corpus of parsable sentences from the web. In *Language Processing and Knowledge in the Web: 25th International Conference, GSCL 2013, Darmstadt, Germany, September 25-27, 2013. Proceedings*, pages 61–68. Springer, 2013.

[Fil20] Peter Filzmoser. Advanced methods for regression and classification - lecture notes, 2020.

[FND19] Alexa K Fox, Chinintorn Nakhata, and George D Deitz. Eat, drink, and create content: a multi-method exploration of visual social media marketing content. *International Journal of Advertising*, 38:450–470, 2019.

[GGT15] Lorenzo Gatti, Marco Guerini, and Marco Turchi. Sentiwords: Deriving a high precision and high coverage lexicon for sentiment analysis. *IEEE Transactions on Affective Computing*, 7(4):409–421, 2015.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

[gita] Gitlab: Glove embeddings. `https://gitlab.com/deepset-ai/open-source/glove-embeddings-de`. Accessed: 2023-10-27.

[gitb] Gitlab: Word2vec embeddings. `https://gitlab.com/deepset-ai/open-source/word2vec-embeddings-de`. Accessed: 2023-10-27.

[HDY+12] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6):82–97, 2012.

[HG14] Clayton Hutto and Eric Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the international AAAI conference on web and social media*, volume 8, pages 216–225, 2014.

[HGC21] Pengcheng He, Jianfeng Gao, and Weizhu Chen. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*, 2021.

[HTFF09] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

[hug] Huggingface: mdebertav3-subjectivity-german model. `https://huggingface.co/GroNLP/mdebertav3-subjectivity-german`. Accessed: 2023-10-30.

[Hun07] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[IL+65] Alekseĭ Grigorevich Ivakhnenko, Valentin Grigorévich Lapa, et al. Cybernetic predicting devices. *(No Title)*, 1965.

[JM] Daniel Jurafsky and James H Martin. Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition.

[JR99] G. Booch J. Rumbaugh, I. Jacobson. *The Unified Modeling Language Reference Manual.* Addison-Wesley, 1999.

[kil] Markus killer: textblob-de. `https://github.com/markuskiller/textblob-de`. Accessed: 2023-10-27.

[KIW16] Maximilian Köper and Sabine Schulte Im Walde. Automatically generated affective norms of abstractness, arousal, imageability and valence for 350 000 german lemmas. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2595–2598, 2016.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

[KRKP+16] Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica Hamrick, Jason Grout, Sylvain Corlay, Paul Ivanov, Damián Avila, Safia Abdalla, Carol Willing, and Jupyter development team. Jupyter notebooks - a publishing format for reproducible computational workflows. In Fernando Loizides and Birgit Scmidt, editors, *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, pages 87–90, Netherlands, 2016. IOS Press.

[KRPA10] Adam Kilgarriff, Siva Reddy, Jan Pomikálek, and PVS Avinesh. A corpus factory for many languages. In *LREC*, 2010.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[KT04] Christopher D. Manning Kristina Toutanova. Stanford pos-tagger description. `https://nlp.stanford.edu/software/tagger.html`, 2004. Accessed: 2023-01-07.

[L+11] Bing Liu et al. *Web data mining: exploring hyperlinks, contents, and usage data*, volume 1. Springer, 2011.

[LBH15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[LJ98] Yong H Li and Anil K Jain. Classification of text documents. *The Computer Journal*, 41(8):537–546, 1998.

[LKH+14] Steven Loria, Pete Keen, Matthew Honnibal, Roman Yankovsky, David Karesh, Evan Dempsey, et al. Textblob: simplified text processing. *Secondary TextBlob: simplified text processing*, 3:2014, 2014.

[LLG+19] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

[LMS+19] Xiaoya Li, Yuxian Meng, Xiaofei Sun, Qinghong Han, Arianna Yuan, and Jiwei Li. Is word segmentation necessary for deep learning of Chinese representations? In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3242–3252, Florence, Italy, July 2019. Association for Computational Linguistics.

[MCCD13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

[meh] Ulrich mehner: Collection of alliterations. `https://www.mehner.info/html/alliteration.html`. Accessed: 2023-10-01.

[Mer14] Dirk Merkel. Docker: lightweight linux containers for consistent development and deployment. *Linux journal*, 2014(239):2, 2014.

[Mic88] Jörg Michael. Nicht wörtlich genommen–schreibweisentolerante suchroutinen in dbase implementiert. *c't Magazin für Computer und Technik*, 10:126–131, 1988.

[MP43] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5:115–133, 1943.

[MP98] Elaine Marsh and Dennis Perzanowski. Muc-7 evaluation of ie technology: Overview of results. In *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29-May 1, 1998*, 1998.

[MSM93] Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. 1993. *Computational linguistics*, 1993.

[nat] 58. protocol of the austrian national council. `https://www.parlament.gv.at/dokument/XXVII/NRSITZ/58/fnameorig_878722.html`. Accessed: 2024-01-24.

[NKK+18] Hiroki Nakayama, Takahiro Kubo, Junya Kamura, Yasufumi Taniguchi, and Xu Liang. doccano: Text annotation tool for human, 2018. Software available from https://github.com/doccano/doccano.

[nou] Nouvertne: Cologne phonetics implementation. `https://pypi.org/project/cologne-phonetics/`. Accessed: 2023-09-21.

[num] pypi: num2words library. `https://pypi.org/project/num2words/`. Accessed: 2023-09-26.

[OEC] OECD trust survey. `https://www.oecd.org/governance/trust-in-government/`. Accessed: 2023-12-08.

[OHMI15] Mahmoud Othman, Hesham Hassan, Ramadan Moawad, and Amira M Idrees. Using nlp approach for opinion types classifier. 2015.

[Pen] Penn treebank tagset. `https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html`. Accessed: 2023-01-07.

[Pos69] Hans Joachim Postel. Die kölner phonetik. ein verfahren zur identifizierung von personennamen auf der grundlage der gestaltanalyse. *IBM-Nachrichten*, 19:925–931, 1969.

[PSM14] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[PVG+11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[RHW86] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[Ris95] Eric Sven Ristad. A natural law of succession. *arXiv preprint cmp-lg/9508012*, 1995.

[Rob18] C Russell Robert. The soundex coding system. *Patent No. US1261167*, 1918.

[Ros58] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[RQH10] Robert Remus, Uwe Quasthoff, and Gerhard Heyer. Sentiws-a publicly available german-language resource for sentiment analysis. In *LREC*, 2010.

[SHP23] Nina Schneidermann, Daniel Hershcovich, and Bolette Pedersen. Probing for hyperbole in pre-trained language models. In Vishakh Padmakumar, Gisela Vallejo, and Yao Fu, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 200–211, Toronto, Canada, July 2023. Association for Computational Linguistics.

[SL08] Helmut Schmid and Florian Laws. Estimation of conditional probabilities with decision trees and an application to fine-grained pos tagging. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 777–784, 2008.

[SMT09] Carolin Strobl, James Malley, and Gerhard Tutz. An introduction to recursive partitioning: rationale, application, and characteristics of classification and regression trees, bagging, and random forests. *Psychological methods*, 14(4):323, 2009.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

[Soua] Oracle: Soundex algorithm. `https://docs.oracle.com/cd/B19306_01/server.102/b14200/functions148.htm`. Accessed: 2023-09-21.

[Soub] Postgres: Soundex algorithm. `https://www.postgresql.org/docs/9.1/fuzzystrmatch.html`. Accessed: 2023-09-21.

[STT95] Anne Schiller, Simone Teufel, and Christine Thielen. Guidelines f ur das tagging deutscher textcorpora mit stts. *Universität Stuttgart, Universität Tübingen, Germany*, 1995.

[Stu17] Mary E Stuckey. American elections and the rhetoric of political change: Hyperbole, anger, and hope in us politics, 2017.

[Swa76] Marc J Swartz. Hyperbole, politics, and potent specification: the political uses of a figure of speech. *Language and Politics*, pages 100–116, 1976.

[TBG+14] Yulia Tsvetkov, Leonid Boytsov, Anatole Gershman, Eric Nyberg, and Chris Dyer. Metaphor detection with cross-lingual model transfer. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 248–258, 2014.

[TKMS03] Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 252–259, 2003.

[TkSP21] Yufei Tian, Arvind krishna Sridhar, and Nanyun Peng. Hypogen: Hyperbole generation with commonsense and counterfactual knowledge, 2021.

[TLPG19] Karsten Tymann, Matthias Lutz, Patrick Palsbröker, and Carsten Gips. Gervader-a german adaptation of the vader sentiment analysis tool for social media texts. In *LWDA*, pages 178–189, 2019.

[TM00] Kristina Toutanvoa and Christopher D Manning. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora*, pages 63–70, 2000.

[TSÖT18] Enrica Troiano, Carlo Strapparava, Gözde Özbal, and Serra Sinem Tekiroğlu. A computational exploration of exaggeration. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3296–3304, 2018.

[TS18] Enrica Troiano, Carlo Strapparava, Gözde Özbal, and Serra Sinem Tekiroğlu. A computational exploration of exaggeration. pages 3296–3304, 2018.

[Vap82] V Vapnik. Estimation of dependences based on empirical data berlin, 1982.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

[Vor21] Vorakit Vorakitphan. *Fine grained classification of polarized and propagandist text in news articles and political debates.* PhD thesis, Université Côte d'Azur, 2021.

[VRD09] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual.* CreateSpace, Scotts Valley, CA, 2009.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[Wic19] Hadley Wickham. Dataset: Diamonds from hadley's ggplot2, October 2019.

[wika] German wikipedia article about climate change. `https://de.wikipedia.org/wiki/Klimawandel`. Accessed: 2023-02-05.

[wikb] German wikipedia article about feminism. `https://de.wikipedia.org/wiki/Feminismus`. Accessed: 2023-02-05.

[wikc] German wikipedia article about the european migrant crisis 2015/2016. `https://de.wikipedia.org/wiki/FlÃijchtlingskrise_in_Europa_2015/2016`. Accessed: 2023-02-05.

[wikd] German wikipedia article about the wikipedia api. `https://de.wikipedia.org/wiki/Wikipedia:Technik/Datenbank/API`. Accessed: 2023-04-14.

[wike] Wikipedia: Cologne phonetics. `https://de.wikipedia.org/wiki/KÃűlner_Phonetik`. Accessed: 2023-09-21.

[Wikf] Wiktionary: Collection of german adjectives. `https://de.wiktionary.org/wiki/Kategorie:Adjektiv_(Deutsch)`. Accessed: 2023-01-07.

[Wil05] Martin Wilz. Aspekte der kodierung phonetischer ähnlichkeiten in deutschen eigennamen. *Master's thesis, Universität zu Köln, Köln*, 2005.

[WRK22] Mayur Wankhade, Annavarapu Chandra Sekhara Rao, and Chaitanya Kulkarni. A survey on sentiment analysis methods, applications, and challenges. *Artificial Intelligence Review*, 55(7):5731–5780, 2022.

[WSJB17] Swantje Westpfahl, Thomas Schmidt, Jasmin Jonietz, and Anton Borlinghaus. Stts 2.0. guidelines für die annotation von pos-tags für transkripte gesprochener sprache in anlehnung an das stuttgart tübingen tagset (stts). 2017.

[WW16] Hadley Wickham and Hadley Wickham. *Data analysis.* Springer, 2016.

[YKD08] Bei Yu, Stefan Kaufmann, and Daniel Diermeier. Exploring the characteristics of opinion expressions for political opinion classification. 2008.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU WIEN Bibliothek Your knowledge hub

[Zar21] Stefan Zaruba. *Using Natural Language Processing to Measure the Consistency of Opinions Expressed by Politicians*. PhD thesis, Wien, 2021.

[ZM06] Justin Zobel and Alistair Moffat. Inverted files for text search engines. *ACM computing surveys (CSUR)*, 38(2):6–es, 2006.

[ZW21] Yunxiang Zhang and Xiaojun Wan. Mover: Mask, over-generate and rank for hyperbole generation. *arXiv preprint arXiv:2109.07726*, 2021.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.