TU WIEN Informatics

Explainable Prediction of User Post Popularity: An Analysis of the One Million Posts Corpus

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Data Science

eingereicht von

Dario Bogenreiter, MSc
Matrikelnummer 11702132

an der Fakultät für Informatik
der Technischen Universität Wien
Betreuung: Univ.Ass. Gábor Recski, PhD

Wien, 27. Jänner 2025

Dario Bogenreiter	Gábor Recski

TU WIEN Informatics

Explainable Prediction of User Post Popularity: An Analysis of the One Million Posts Corpus

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Data Science

by

Dario Bogenreiter, MSc
Registration Number 11702132

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Ass. Gábor Recski, PhD

Vienna, January 27, 2025

Dario Bogenreiter	Gábor Recski

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

Erklärung zur Verfassung der Arbeit

Dario Bogenreiter, MSc

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, 27. Jänner 2025

Dario Bogenreiter

Danksagung

Zuerst möchte ich dem österreichischen Staat und der TU Wien danken, dass sie mir die Möglichkeit gegeben haben, mein Studium zu verfolgen. Ich bin fest davon überzeugt, dass zugängliche und leistbare Bildung ein Grundpfeiler unserer Gesellschaft ist – ein unschätzbarer Wert, der es verdient, verteidigt zu werden.

Als Nächstes gilt mein Dank meinen Betreuer, Gábor Recski – nicht nur für seine Unterstützung und sein wertvolles Feedback während der gesamten Arbeit, sondern auch für die Organisation von Initiativen wie dem Seminar *194.135 on research topics in Natural Language Processing (NLP)*. Solche Formate bieten eine großartige Plattform, um Forschungsthemen aus dem NLP-Bereich gemeinsam in einer Gruppe zu diskutieren, neue Perspektiven zu entdecken und von anderen zu lernen.

Schließlich möchte ich meiner Familie und meinen Freunden danken, die immer für mich da waren und mich auf meinem Weg unterstützt haben.

Acknowledgements

First, I want to thank the Austrian state and TU Wien for providing me with the opportunity to pursue my studies. I firmly believe that accessible and affordable education is a cornerstone of our society — a priceless achievement that should not be taken for granted.

Next, I would like to express my gratitude to my supervisor, Gábor Recski, not only for his guidance and support throughout the thesis but also for organizing initiatives like the *Seminar 194.135 on research topics in NLP* which offer an invaluable platform for discussing and exploring NLP in a collaborative environment.

Finally, I want to thank my family and friends for always being there for me and supporting me along the way.

Kurzfassung

Diskussionen in den Kommentarbereichen von Zeitungen beeinflussen die öffentliche Meinung erheblich. Die Methoden, die zum Sortieren und Anzeigen von Nutzerbeiträgen verwendet werden, spielen eine entscheidende Rolle bei der Steuerung und Beeinflussung dieser Diskussionen. Die Sortierung wird jedoch oft teilweise von Forenmoderatoren vorgenommen, was zeitaufwändig ist und oftmals ungewünscht von der persönlichen Meinung der Moderatoren beeinflusst wird. Machine-Learning-Modelle, die auf der Grundlage von Nutzerabstimmungsstatistiken trainiert werden, bieten eine potenzielle Lösung für dieses Problem, indem sie ansprechende Beiträge automatisiert auf Basis der dokumentierten Nutzermeinungen identifizieren. Frühere Forschungen in diesem Bereich haben sich in erster Linie auf die Verbesserung der Vorhersagegenauigkeit mithilfe von Deep-Learning-Ansätzen konzentriert, wobei der kritische Aspekt der Erklärbarkeit oft vernachlässigt wurde. Diese Studie untersucht eine Reihe von erklärbaren Methoden zur algorithmischen Identifizierung von wertvollen Benutzerbeiträgen im *One Million Posts*-Korpus, der von der Website der österreichischen Tageszeitung *DerStandard* stammt. Es werden erklärbare Features vorgestellt und erklärbare und interpretierbare Modelle evaluiert, wobei ihre Leistung mit Deep-Learning-Ansätzen verglichen wird. Die Ergebnisse zeigen, dass interpretierbare Modelle, die auf erklärbaren Merkmalen trainiert wurden, die gängigen Baselines in diesem Bereich übertreffen. Sie bleiben jedoch hinter der Vorhersagekraft von Deep-Learning-Ansätzen zurück. Trotz ihrer geringeren Vorhersagekraft bieten diese interpretierbaren Ansätze wertvolle Einblicke in die algorithmische Entscheidungsfindung und ihre potenziellen Fallstricke. Zusätzlich zu den Modellergebnissen wird in dieser Arbeit eine umfassende Analyse der Bedeutung der erklärbaren Merkmale vorgestellt und ein neuartiger Labeling-Ansatz für engagierte Beiträge vorgeschlagen der sowohl für große als auch kleine Datensätze geeignet ist.

Abstract

Discussions in newspaper comment sections significantly influence public opinion. The methods used to sort and display user posts impact these discussions and can propagate certain opinions. However, sorting is often partially done by forum moderators, which is time-consuming and prone to bias. Machine learning models trained on user voting statistics offer a potential solution by automatically identifying engaging posts based on the documented opinion of the community. Prior research in this domain has primarily focused on improving predictive accuracy using deep learning approaches, often neglecting the critical aspect of explainability. This study explores a range of explainable methods to algorithmically identify valuable user posts in the *One Million Posts* corpus, sourced from Austrian newspaper forum *DerStandard*. It introduces explainable features and evaluates explainable and interpretable models, comparing their performance against deep learning approaches. Results show that interpretable models trained on explainable features outperform popular baselines in this domain. However, they fall short of the predictive power of deep learning approaches. Despite their lower predictive power, these interpretable approaches provide valuable insights into algorithmic decision-making and its potential pitfalls. In addition to the model results, this work presents a comprehensive analysis of the importance of the explainable features and proposes a novel labeling approach for engaging posts designed to accommodate both small and large datasets.

Contents

CHAPTER 1

Introduction

1.1 Motivation & Problem Statement

Even before the 2016 U.S. presidential election between Donald Trump and Hilary Clinton, it was evident that what happens on the Internet, whether on social media or news websites, has a significant impact on public opinion. Discussions on the web, expressed through user posts, can shape opinions, drive agendas, and potentially even sway elections. The visibility of these posts often depends on the number of likes and dislikes they receive, with highly liked posts being prioritized. Understanding the reasons why certain posts trend and attract varying numbers of likes is crucial.

Social Media Popularity Prediction (SMPD) is the endeavor to predict the popularity of content posted on social networks (Ding, Wang & Wang, 2019). In the past, research within this area focused mainly on data from Twitter, (as for example in a paper from Daga, Gupta, Vardhan und Mukherjee (2020)) and other standard social media (Lai, Zhang & Zhang, 2020; Liu et al., 2022; Trujillo & Cresci, 2022). However, when discussing and forming their political opinions, online communities of newspaper websites play a major role (Boczkowski & Mitchelstein, 2012). Such websites are not classic social media websites. Despite their significant influence, research on these websites remains limited compared to studies on traditional social media. Furthermore, only a small number of researchers, such as Risch und Krestel (2020a), have explicitly explored predictive models for user post popularity prediction on these types of platforms. Additionally, most studies have focused on English texts and news, often neglecting German use cases, which may exhibit distinct characteristics.

The Problem - On platforms like news websites, users can only process a small portion of available content, making it crucial to prioritize and present the most relevant content. The manual ranking done by moderators comes with the disadvantage of eventually propagating their personal opinion, leading to unwanted outcomes, such as polarisation (Shmargad & Klar, 2020). Furthermore, it naturally is time- and resource-intensive. Automated ranking systems offer a cost-effective alternative to manual ranking, which is often biased and labor-intensive (Risch & Krestel, 2020a). However, state-of-the-art deep learning models lack transparency, making it unclear if predictions are based on valid factors. While prior research (Park, Sachar, Diakopoulos & Elmqvist, 2016; Risch & Krestel, 2020a) has explored some explainable Artificial Intelligence (AI) approaches, it primarily focuses on improving prediction accuracy, leaving a gap in explainable prediction of post popularity.

Another challenge lies in the uniqueness of online communities, as factors influencing post popularity can vary significantly across platforms due to differences in user bases, features, and dynamics. To address this, this research focuses on text from the 2010s on *DerStandard*, one of Austria's leading platforms for political discussions. The 'One Million Posts Corpus' (Schabus, Skowron & Trapp, 2017), which contains over one million user posts from *DerStandard*, is particularly well-suited for this study, as it represents the largest collection of data from this platform during that time and offers sufficient variety to capture the diverse discussions happening within this community.

1.2 Research Questions

This study focuses on identifying and analyzing 'engaging posts', defined as the top 10% of posts in terms of upvotes (as the primary ranking factor) and number of replies (as the secondary factor), within a group of 10 posts written around the same time in the same comment section of an article. Conversely, 'regular posts' are classified as the bottom 10% of posts in the respective grouping.

Additionally, this research explores explainable features, defined as easily understandable and interpretable by humans, alongside non-explainable features. The performance of these features is analyzed using explainable models, defined as models where humans can clearly understand the reasoning behind the decisions; interpretable models, where humans can identify key influencing factors; and deep learning models, which function as black boxes and cannot simply be interpreted by humans without additional assistance.

The research is structured around the following core questions:

R1. Do the explainable features created in this work differentiate significantly between the two classes of 'engaging' and 'regular posts'?

R2. How do different black-box, deep learning models perform in predicting post popularity compared to simple baseline models and models trained on the explainable features introduced in work?

These research questions correspond to the following null hypothesis:

$H_0$1. There is no significant difference in features between the classes of 'engaging' and 'regular posts.'

$H_0$2. There is no significant performance difference between deep learning models (such as a Long Short Term Memory (LSTM) or Bidirectional Encoder Representations from Transformers (BERT) models), fully explainable models, interpretable models, and the baseline solutions introduced in this work when predicting post popularity.

1.3 Expected Contribution

This thesis introduces a structured pipeline for creating explainable features to predict post popularity and provides a detailed evaluation of the algorithmic solutions applied. The code base for this project is publicly available in a GitHub repository[1] under the MIT license[2].

Rather than solely prioritizing quantitative prediction accuracy, this research focuses on understanding the underlying factors that drive user engagement and seeks to connect these findings with prior research.

1.4 Structure of the Thesis

Section 2 introduces the dataset, highlights its unique characteristics, and provides a detailed data analysis. Section 3 reviews related work, covering both studies that utilize the same dataset and research within the broader area. Section 4 outlines possibilities for generating explainable features for popularity prediction, by presenting the results of a respective literature review and then continues to present those concrete features that were developed together with the models in the experiments. Section 5 discusses the experimental setup for popularity prediction and introduces the applied models. The results of the experiments are presented in Section 6, followed by a discussion and analysis in Section 7. Finally, Section 8 concludes the thesis by summarizing key findings, listing the limitations, and suggesting directions for future research.

[1] https://github.com/dario-x/user-post-popularity-prediction
[2] https://opensource.org/licenses/MIT

CHAPTER 2

The 'One Million Posts Corpus' dataset

The One Million Posts Corpus dataset, introduced by Schabus et al. (2017), contains information on over one million comments related to articles from the Austrian daily newspaper *DerStandard*. This dataset, created by the Austrian Research Institute for Artificial Intelligence (OFAI), originates from the newspaper's website, where registered users can post comments below news articles (Schabus et al., 2017). Users can also reply to earlier comments, creating tree-like discussion thread structures (Schabus et al., 2017).

The dataset includes the following data columns for these posts:

- ***Post ID*** - unique identifier for each post
- ***Article ID*** - identifier for the article in whose comment section the post appears
- ***User ID*** - anonymized identifier for the user who commented
- ***Headline*** - headline of the post (max. 250 characters)
- ***Main Body*** - main content of the post (max. 750 characters)
- ***Time Stamp*** - time when the post was created
- ***Parent Post*** - identifier for the parent post if the comment is a reply
- ***Status*** - indicates if the post is online or was deleted by a moderator
- ***Positive Votes*** - number of positive votes by other users
- ***Negative Votes*** - number of negative votes by other users

Additionally, the dataset includes details about the articles under which users have posted their comments. This information includes:

- ***Article ID*** - unique identifier for each article
- ***Path*** - topic of the article (e.g., 'Newsroom/Sports/')
- ***Publishing Date*** - timestamp of when the article was published
- ***Title*** - headline of the article
- ***Body*** - full text of the article

The dataset contains 11,773 labeled posts and an additional one million unlabeled posts, totaling 1,011,773 posts (Schabus et al., 2017). The term 'labeled' refers to posts categorized into seven categories designed for tasks such as hate speech detection. These categories are (Schabus et al., 2017):

- ***Sentiment*** - detecting tone shifts, e.g., positive, neutral, or negative
- ***Off-Topic*** - posts unrelated to the article's subject
- ***Inappropriate*** - containing insults, threats, or offensive language
- ***Discriminating*** - e.g., sexist, racist, or misanthropic content
- ***Feedback*** - comments asking questions or requiring replies from the author
- ***Personal Stories*** - users sharing experiences, private stories, or anecdotes
- ***Arguments*** - providing logical reasoning and/or sources for their claims

The annotation process involved three rounds conducted by four professional forum moderators working for *DerStandard* newspaper (Schabus et al., 2017). The first stage served as a trial run to fine-tune the annotation procedure and clarify category definitions and was excluded from the final dataset (Schabus et al., 2017). Annotators labeled 160 posts, written in the comment section of a recent article, in parallel and then discussed the differences in their annotations to establish common definitions for labels (Schabus et al., 2017).

The second stage aimed to establish category distributions, measure inter-annotator agreement, and produce an initial body of labeled data (Schabus et al., 2017). In this phase, three moderators independently annotated a randomly selected sample of 1,000 posts, followed by another round of discussions (Schabus et al., 2017).

The third stage prioritized increasing samples for categories that were underrepresented in the second phase (Schabus et al., 2017). To achieve this, posts were selected using three

targeted strategies: (1) 2,599 posts were taken from articles with a high percentage of comments that were deleted from moderators, aimed at capturing categories like *negative sentiment*, or *inappropriate*; (2) 5,737 posts were taken from a 'share your thoughts' section on the newspaper platform, aimed to find more posts that share *personal stories*; and (3) 2,439 posts were chosen in that hope that they contained *feedback*, based on the fact that they received direct replies from staff members or were labeled as feedback by a classifier developed by the authors (Schabus et al., 2017).

The completed dataset comprises 3,599 posts annotated for all categories, along with 5,737 posts labeled for *personal stories* and 2,439 posts labeled for *feedback*, amounting to a total of 58,568 individual expert judgments. Inter-annotator agreement, was assessed using Cohen's Kappa based on the second round (which involved three annotators) (Schabus et al., 2017). Results ranged from 0.3 (fair) for *off-topic* and *discriminating* to 0.5 (moderate) for *inappropriate* and *feedback*, highlighting the inherent complexity of the categories (Schabus et al., 2017). Furthermore, pairwise agreement was notably higher between some annotators, particularly between AB, compared to other combinations (Schabus et al., 2017).

The authors highlighted the dataset's versatility, noting its potential to support additional use cases (Schabus et al., 2017). One such additional application is predicting post popularity based on positive and negative vote counts. This task benefits from a massive amount of ground-truth data generated directly by a large user pool. Unlike other typical NLP tasks such as sentiment detection, which often require manual annotation by researchers or domain experts.

Out of the 1.01 million posts, 0.69 million have received at least one upvote or downvote, as shown in Table 2.1. Specifically, 0.63 million posts have received at least one upvote, while only 0.3 million posts have received at least one downvote. Overall, upvotes are more common than downvotes. The median (1) and mean (3.78) number of upvotes per post are significantly higher than the median (0) and mean (1.08) number of downvotes per post. This disparity may result from several factors: offensive comments being deleted, users being more reluctant to post comments that might receive negative attention, or the tendency of like-minded individuals to comment on certain articles. Further research is needed to determine the exact reasons.

Table 2.1: Number of Posts in Different Categories

Category	**Number of Posts (in millions)**
All Posts	1.01
Posts with Votes	0.69
Posts with UpVotes	0.63
Posts with DownVotes	0.30

Figure 2.1 shows a heatmap of the posts, clustered according to the number of upvotes and downvotes. Each cell represents the count of posts with a specific number of upvotes and downvotes. For example, 126 posts have 0 downvotes and between 100 and 500 upvotes. Summing up all the cells in the heatmap gives 1.01 million posts. The heatmap shows that most posts with more than 100 upvotes have either 0 or just a few downvotes. All posts with more than 100 downvotes have at least one upvote. In addition, we can see that about 0.32 million posts have 0 up- and downvotes, which is the highest count for any combination of up- and downvotes.

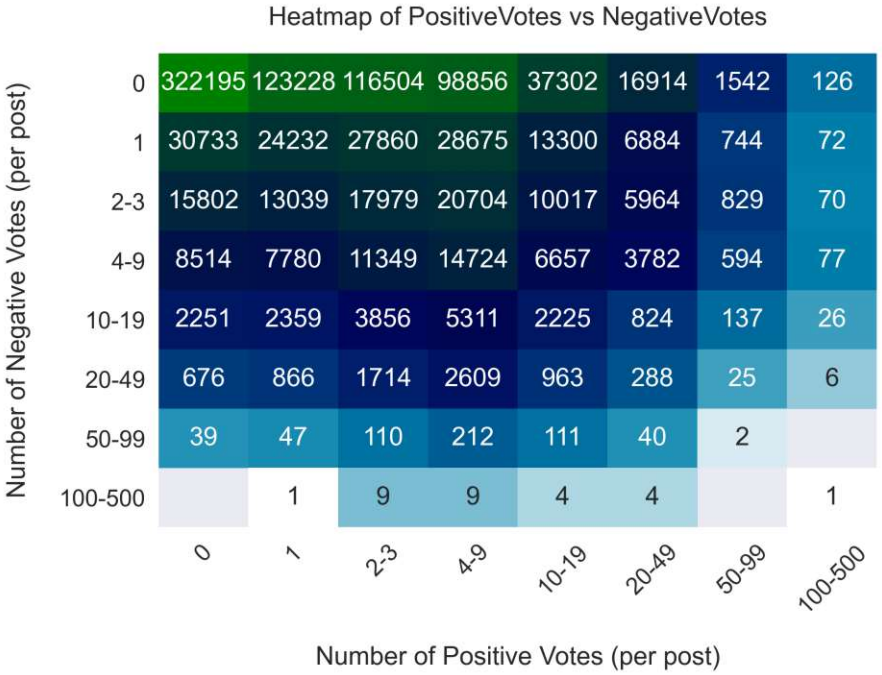

Figure 2.1: Heatmap of Posts, Clustered by Up- and Downvotes

A reason why many posts receive very little attention (in terms of upvotes and downvotes) is simply that the vast number of posts limits what users can read (Risch & Krestel, 2020a). Consequently, many posts do not get enough feedback to determine whether they are perceived positively, neutrally, or negatively. Posts with no votes or an equal number of down- and upvotes present a general problem, as users do not have the option to give a post a neutral vote, and the number of reads per post is not collected.

A further limitation of the amount of attention a post can receive lies in the continuous publication of news articles, leading to older articles and their comments being forgotten. This naturally caps the amount of interaction a post can receive, to about 500 upvotes and 500 downvotes.

2.1 Biases of the Dataset

This section highlights a few important biases to be considered when working with the dataset. As they can distort the experiments of this work, some of them are partially corrected by applying additional preprocessing. For transparency purposes, all of them (correctable and non-correctable) are listed in this section.

Bias 1: Position Bias - the Privilege of Pinned Posts

One reason why only a few posts receive substantially more votes than others is that posts can be pinned by moderators of the *Standard* forum. These are then highlighted on the website, as shown in Figure 2.2 which depicts the layout of the comment section on the website. These comments are listed above the text input field and all other posts. Each article can have one or up to ten pinned posts, and they remain in this prominent position until other posts receive even more attention.

Figure 2.2: Layout of the Comment Section

The present layout boosts the visibility of posts written shortly after the article is released. Later comments, even if more interesting, have little chance of becoming pinned. The prominent position enables pinned posts to collect more and more votes, making it difficult for later comments to accumulate a similar amount of votes. This is why most other newspaper platforms have reverted to listing posts solely in chronological order (Risch & Krestel, 2020a). Users have the option to sort comments based on upvotes, downvotes, or oldest first, as shown in Figure 2.2. However, this requires additional effort, which is unlikely to be performed frequently by the average reader.

Bias 2: Time Bias

Closely related to the first bias, is the second bias: posts written later tend to receive less interaction, even in a chronological layout. Comments written shortly after the article is posted simply have more time to accumulate reactions. Additionally, the number of users clicking on an article declines as more time passes after its publication, eventually dropping to nearly zero as newer articles are published. This decline naturally reduces interactions in the comment section, negatively affecting later-written posts. The time bias manifests in two ways: the order in which posts are written (e.g., first, second, etc.) and the time passed after the article's publication and the creation of a post.

Bias 3: Comment Removal by Moderators

A possible explanation why upvotes are more common than downvotes might come from the fact that many posts contain offensive content or violate community guidelines and have therefore been removed - a problem, that has even led to newspapers shutting down their comments section (to save money on the employment of moderators checking these guidelines) (Nelson, Ksiazek & Springer, 2021). This naturally distorts the number of received up- and downvotes of these posts, as users can only see them for a limited time.

Bias 4: Discussion Dynamics

Another bias arises from the tree-like structure of the discussion threads in the dataset. In the comments section of the newspaper, users can reply to posts, leading to a hierarchical, tree-like structure where multiple threads emerge. This structure introduces several complications:

The first and perhaps easiest to understand is that the reply situation is inherently tied to the time bias, as replies are naturally written after the original posts. Consequently, replies typically have later timestamps, and as discussed earlier, posts written later tend to receive fewer votes on average. This could explain why replies generally receive fewer votes compared to standalone posts. However, there may be other contributing factors. Since the status of being a reply cannot be clearly separated from the time bias, it becomes challenging to determine whether replies are inherently less popular or if their lower average vote count is simply a result of this timing effect.

Secondly, the popularity of a parent post can be significantly influenced by the discussions that unfold beneath it. An engaging discussion can boost the visibility of the parent post, leading to higher vote counts. However, this introduces a potential distortion: the parent post may not be particularly interesting or valuable on its own but might receive additional votes due to the quality of its replies (including replies to other replies), rather than its content.

Thirdly, the popularity of replies can be significantly influenced by their predecessors. For instance, if a comment is widely perceived as 'stupid' by the community, a subsequent reply pointing this out might gain popularity, not because it offers valuable insights, but simply because it criticizes the other post. This can lead to misleading engagement for posts that do not add much meaningful content on their own, but instead derive their popularity from the shortcomings of others. Figure 2.3 illustrates such a situation. Initially, a user writes an engaging comment that gains considerable attention and upvotes, suggesting that in the long run. different cultures within a country should live together in mutual respect and understanding ('miteinander auskommen'). Next, another user criticizes this comment, calling it ironic; this post receives almost no upvotes. Finally, a third user writes a reply containing only '?' to criticize the previous post, which then gains a substantial number of upvotes despite adding no meaningful content.

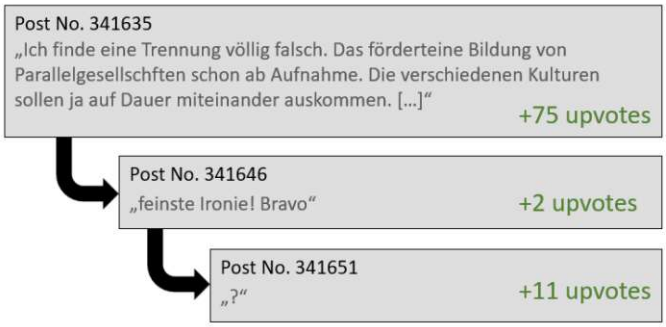

Figure 2.3: Example of Engagement Influenced by Preceding Posts

Bias 5: Unique User Group

The dataset reflects interactions from a specific demographic group of users. Approximately 46 percent of all *DerStandard* readers hold an academic degree, and the newspaper is the most widely read quality newspaper among decision-makers in Austria[1]. The community has a gender-imbalance with only 39 percent of readers being women, and the average reader is 47 years old. The newspaper is also known for its left-leaning political stance, which naturally attracts a particular audience.

As a result, what makes a post trending or popular within this community may differ significantly from what drives popularity in other contexts, whether at the national or global level. As previously mentioned, each community has its unique characteristics and dynamics, making it essential to document these specifics when analyzing the data.

[1] https://www.derstandard.at/story/3000000212511/standard-ist-meistgenutztes-qualitaetsmedium-bei-entscheidungstraegern

CHAPTER 3

Related Work

3.1 Previous Research on the One Million Posts Dataset

As previously mentioned, the dataset used in this research was originally designed for text classification tasks, mainly for sentiment analysis and hate speech detection (Schabus et al., 2017). The authors of the dataset developed several approaches to predict whether a post falls into one of nine predefined categories: *Negative Sentiment*, *Positive Sentiment*, *Neutral Sentiment*, *Off Topic*, *Inappropriate*, *Discriminatory*, *Asking for Feedback*, *Shares Personal Story*, and *Uses Rational Argumentation and Reasoning* (Schabus et al., 2017).

For their baseline solutions, the authors employed a Bag of Words (BOW) model in conjunction with a Support Vector Machine (SVM) (referred to as BOW). They also explored more advanced approaches, including a doc2vec-based document embedding combined with an SVM (D2V), an SVM trained on a dimensionally reduced version of the BOW (termed bag of cluster IDs or BOCID), an SVM using log-count ratios derived from a naive Bayes model (NBSVM), and a neural network architecture utilizing a LSTM model (LSTM) (Schabus et al., 2017). The performance of these models was evaluated using precision, recall, and F1 score metrics (Schabus et al., 2017). The results varied significantly across models and classification tasks (Schabus et al., 2017).

The BOW model demonstrated robust performance across most tasks, achieving the highest precision for identifying posts containing arguments (0.61) and personal stories (0.69) (Schabus et al., 2017). In contrast, the more advanced LSTM model excelled in terms of F1 score for identifying negative posts (0.61) and posts asking for feedback (0.62) but notably failed to identify any posts with positive sentiment (Schabus et al., 2017). D2V achieved the highest F1 score for posts containing arguments (0.60) and personal stories (0.70) (Schabus et al., 2017). BOCID performed best for off-topic (0.31), inappropriate (0.21), and discriminatory posts (0.18) (Schabus et al., 2017). Meanwhile, NBSVM achieved the highest F1 score for positive posts (0.13) (Schabus et al., 2017).

These results highlight the complexity of text classification tasks and the varying effectiveness of different models depending on the specific label and evaluation metric. This underscores the importance of testing diverse approaches and illustrates the challenges inherent in solving such tasks.

In 2018, Wiedemann, Ruppert, Jindal und Biemann (2018) explored the 11,773 labeled posts from the dataset to investigate the potential of transfer learning for the task of offensive content classification. The researchers chose the label 'offensive' since they believed it is generally easier for people to agree on compared to more fine-grained categories like 'hostile', 'discriminatory', or 'abusive' (Wiedemann et al., 2018). Their approach testing out different transfer learning strategy on two different cases and then comparing the results to a baseline model which did not make use of transfer learning (Wiedemann et al., 2018). Despite their effort, transfer learning demonstrated only minor improvements, raising accuracy from roughly 80% to 82% in the first case, and from 75% to 0.76% (Wiedemann et al., 2018). The limited success may stem from their use of the labels 'inappropriate' and 'discriminatory' as proxies for determining whether a comment was offensive (Wiedemann et al., 2018). While inappropriateness and discrimination are closely related to offensiveness, the authors acknowledged that these terms cannot be used interchangeably, as the latter categories are inherently more specific, as just mentioned (Wiedemann et al., 2018). To reflect this limitation, they described their task as near-category transfer learning (Wiedemann et al., 2018).

Risch and Krestel emphasized the value of the labeled part of the dataset (Risch & Krestel, 2020b), who listed it as a common dataset for supervised training on the detection of toxic comments. They defined toxic comments as a complex concept primarily employed by spammers, haters, and trolls, which reduces user engagement on the platform (Risch & Krestel, 2020b). Furthermore, they explained that toxicity is a challenging topic because users who post toxic comments often intentionally try to conceal the actual meaning of their posts. Stylistic devices such as irony further hinder classification (Risch & Krestel, 2020b). These remarks are valuable as these challenges could also apply when determining the popularity of a comment. For instance, some top comments might exhibit a high level of irony and be celebrated for this circumstance, as they are perceived as particularly funny.

Scheible et al. (2024) utilized a subset of the One Million Posts dataset, known as the Ten Thousand German News Articles Dataset (10kGNAD) [1], to evaluate their newly developed transformer model, German OSCAR text trained BERT (BERT). At the time, GottBERT was the first A Robustly Optimized BERT Pretraining Approach (BERT) (Zhuang, Wayne, Ya & Jun, 2021) model specifically designed as a monolingual transformer for German (Scheible et al., 2024). It was trained on 145 GB of data from sources such as Wikipedia, the EU Bookshop corpus, and the German portion of the Super-large Crawled Aggregated Corpus (OSCAR) (Scheible et al., 2024). The 10kGNAD dataset is derived from the `article` table of the One Million Posts dataset and leverages

[1] `https://tblock.github.io/10kGNAD/`

the second part of the path variable to construct ground-truth topic labels. For example, the path 'Newsroom/Wirtschaft/Wirtschaftpolitik/Finanzmaerkte/Griechenlandkrise' is mapped to the topic group `Wirtschaft` (economy). The GottBERT model was evaluated the predictive ability of the model in topic classification using this dataset. In addition to topic classification, the model was tested on two Named Entity Recognition (NER) tasks and two other classification tasks (Scheible et al., 2024).

For the 10kGNAD task, GottBERT was outperformed by the dbmz BERT model [2], which achieved an F1 score of 0.90, slightly higher than GottBERT's F1 score of 0.89 (Scheible et al., 2024). In both NER tasks, GottBERT achieved the highest F1 scores of 0.84 and 0.87, outperforming other models (Scheible et al., 2024). For another classification taks - GermEval2018 (Montani & Schüller, 2018), which focused on classifying Twitter posts into offensive and non-offensive categories (binary task) and further into four categories—profanity, insult, abuse, and other (fine-grained classification)—GottBERT achieved the highest F1 score (0.51) for the fine-grained task (Scheible et al., 2024). However, for the binary classification task, dbmz BERT achieved an F1 score of 0.77, slightly surpassing GottBERT's F1 score of 0.76 (Scheible et al., 2024). These results highlight that even the granularity level of a classification problem can significantly affect model performance, emphasizing the importance of tailoring models to specific tasks.

Inspired by 10kGNAD, Wehrli, Arnrich und Irrgang (2023) developed two new sub-datasets: 'TenKGnadClusteringS2S', consisting only of the titles of news articles, and 'TenKGnadClusteringP2P', consisting of the full texts of the articles. Each dataset contains slightly more than 10,000 samples and is divided into nine clusters, as indicated by the path variable (Wehrli et al., 2023). These sub-datasets were utilized to perform text clustering on news articles (excluding user posts) as part of the German Text Embedding Clustering Benchmark (Wehrli et al., 2023). Out of the 12 clustering algorithms evaluated, GBERT-large achieved the best performance for the dataset containing only titles, with a V-measure score of 34.97 (Wehrli et al., 2023). For the dataset containing full texts, Sentence-T5 performed best, achieving a V-measure score of 44.88 (Wehrli et al., 2023).

Pachinger et al. (2024) point out the absence of target and vulgarity spans in the 'One Million Dataset', which are essential for fine-grained hate speech detection and analysis. To address this gap, they introduced a new dataset called *Austrotox*, comprising of 4,562 posts from *DerStandard* that include these spans.

[2] https://github.com/dbmdz/berts#german-bert

3.2 Popularity Prediction

In 2020, Risch and Krestel addressed the problem of classifying user posts on the English newspaper site *The Guardian* into the categories 'top' and 'flop' posts (based on the number of upvotes and replies they received) (Risch & Krestel, 2020a). Their primary motivation was to demonstrate an automated method for identifying engaging posts without requiring expensive manual annotation efforts by editors (Risch & Krestel, 2020a). Such a method could improve user experience by recommending posts for readers to read or reply to (Risch & Krestel, 2020a). This research is arguably the most related to this work. However, the main difference lies in their approach, which focused on designing a new deep learning architecture and embeddings to improve accuracy, without emphasizing explainability. Furthermore, the domain and problem are different, as the sorting algorithm applied by *DerStandard* differs from that of *The Guardian*, where posts are ranked chronologically and pinned comments are not present (Risch & Krestel, 2020a).

Most interestingly, they identified challenges similar to those found in the 'One Million' dataset mentioned in Section 2 of this paper, such as earlier comments receiving more upvotes—a phenomenon they referred to as position bias, as comments are ranked chronologically (Risch & Krestel, 2020a). To address this, they grouped all comments written first in a comment section into one group, then grouped the second set of comments, and so on (Risch & Krestel, 2020a). They then sorted these groups by the share of upvotes or replies received within their original comment sections and selected the top and flop comments from each group (Risch & Krestel, 2020a).

Another similar challenge they faced, also present in the 'One Million' dataset, is that some articles have very few comments, making it difficult to estimate popularity accurately (Risch & Krestel, 2020a). They addressed this by removing such posts from their analysis (Risch & Krestel, 2020a).

For predicting 'top' and 'flop' comments, they employed four approaches: a baseline logistic regression trained on text length; logistic regression with user features (such as previous upvotes of users, user activity, etc.) proposed by Park et al. (2016); Convolutional Neural Network (CNN); and their newly designed Gated Recurrent Unit (GRU) model (Risch & Krestel, 2020a). The features introduced by Park et al. (2016) included text length, readability, and averages of text length, number of received comment upvotes, and readability per user. Their newly proposed solution outperformed the other models, achieving an F1 score of 0.71 on a balanced dataset of the top and flop 10% in terms of upvotes (Risch & Krestel, 2020a). The logistic regression achieved an F1 score of 0.61, the CNN 0.65, and the approach by Park et al. (2016) 0.65 (Risch & Krestel, 2020a). When decided by replies, these figures were 0.70 (GRU), 0.63 (logistic regression), 0.69 (CNN), and 0.61 (Park et al. (2016)) (Risch & Krestel, 2020a).

Furthermore, they tested their models on another dataset—Amazon product reviews—where the F1 scores were significantly better: 0.67 (baseline logistic regression), 0.67 (GRU), and 0.76 (CNN) (Risch & Krestel, 2020a). However, the better results can be explained by the simpler use case, which is less critical than political discussions. The features proposed by Park et al. (2016) also appear interesting for this research, as they are explainable. These features include post readability, length, average length (that the user usually reads), average readability, and upvotes (Park et al., 2016). However, only length and readability are relevant to this work (and are explored later), as this paper focuses on predicting post engagement solely based on the post itself, without relying on historical information. Such information is not always available—this was also the case for the Amazon dataset (Risch & Krestel, 2020a)—and could introduce new biases. For example, just because a user used to write interesting posts (and received upvotes for them), it does not mean they will continue to do so. A possible solution based on these historical features would be prone to biases arising from this assumption. Hence, such historical user features are excluded from this research.

Besides the endeavor of identifying valuable and engaging user posts by the number of votes they receive, there has been ongoing research on how articles could be classified as particularly valuable by looking at the posts (comments) that they receive (Ambroselli, Risch, Krestel & Loos, 2018; Bandari, Asur & Huberman, 2012; Tsagkias, Weerkamp & De Rijke, 2009). Tsagkias et al. (2009) applied Random Forests to identify popular articles through a two-step process. First, they determined if users would comment on an article at all. Second, they predicted the number of comments that would appear under the article (Tsagkias et al., 2009). They noted that the second task is significantly more challenging, and accurately predicting the exact number of comments is practically infeasible (Tsagkias et al., 2009). Similarly, Bandari et al. (2012) studied this problem in the context of social media and reached comparable conclusions. They found that predicting popularity as a regression task results in large errors, so they redefined it as a classification task by grouping articles based on the number of comments they receive (Bandari et al., 2012).

These findings are highly relevant to the problem addressed in this work, as discussed in Section 5.1, since both distributions — of the number of comments an article receives and of the number of votes a post garners — exhibit similar characteristics, with a small fraction of posts or articles receiving a disproportionately high level of attention, while the majority receive little to none.

3.3 German NLP research

In the context of German NLP tasks, the *20th Conference on Natural Language Processing (KONVENS 2024)* (De Araujo et al., 2024) lists several papers relevant to this work. These studies collectively illustrate the breadth of research in German NLP. Although they address different primary objectives, their methodologies and findings provide useful insights and potential applications for this thesis.

Hellwig, Fehle, Bink und Wolff (2024) introduce the dataset *GERestaurant*, which contains German-language reviews from the website *Tripadvisor*. While *Tripadvisor*, like newspaper websites, is not a traditional social media platform, it shares similar characteristics in its reliance on user-generated content that are also similar to this sutdy where the online community on newsapper websites is analysed. The dataset is manually labeled for the task of Aspect Based Sentiment Analysis (ABSA), categorizing reviews into sentiment classes—positive, negative, or neutral—and further distinguishing them as explicit or implicit sentiments (Hellwig et al., 2024). Additionally, the dataset groups posts into aspect categories such as *food*, *service*, and *ambience* (Hellwig et al., 2024). This dataset, while focused on sentiment analysis, also shows the possibility of exploring the popularity of reviews. For example, understanding what makes certain reviews more engaging could be beneficial for food critics or professional reviewers, who might need to know how to write engaging reviews. Similarly, restaurants might gain insights into whether there are specific aspects, such as *ambience* or *service*, that they must prioritize to achieve better reviews, as posts discussing these aspects may be more likely to trend.

Benaicha, Thulke und Turan (2024) present a Cross-Lingual Transfer Learning model for the task of Spoken NER. NER focuses on identifying and classifying entities within text, such as public figures (e.g., Angela Merkel) or locations (e.g., the United States) (Benaicha et al., 2024). While in this thesis NER is utilized as an explainable feature for predicting post popularity, Benaicha et al. (2024) do not cover this topic, but focus simply on the task of extracting named entities directly from voice rather than text (as it is the case in this study), their findings remain relevant to this research. Although most current newspaper website only support text-based comments, allowing users to leave voice recordings might become a feature in the future. Such a development would make spoken NER approaches directly applicable. In general, the identification of named entities in textual comments remains an important factor in understanding post popularity, as discussed later in this work.

Petersen-Frey und Biemann (2024) investigate the detection and attribution of quotations in German news articles. Their work involves building models capable of identifying not only obvious cases, such as direct quotes but also more challenging instances, such as indirect or atypical quotations. While this study does not explicitly incorporate such models, it is relevant because users often quote text in their comments, including excerpts from newspaper articles or other sources. Developing a dedicated quotation detection model to feed this data into an explainable popularity prediction model might go beyond

the scope of this work. However, the frequent appearance of direct quotes is still leveraged as an explainable feature in this study, reflecting their association with engaging and popular posts.

3.4 Explainability

As already mentioned, this study focuses on the explainable prediction of user post popularity. To clarify, it is beneficial to discuss the concepts of *explainability* and *interpretability*. Explainability, in the context of making predictions, refers to the ability to present the necessary textual or visual information to the user or creator of a model in a way that enables them to sufficiently understand the connection between the input features and the output predictions (Ribeiro, Singh & Guestrin, 2016). This makes explainability a valuable tool for building human trust in AI models (Ribeiro et al., 2016).

Interpretability, on the other hand, is a closely related concept that refers to how well a user can directly understand and make sense of the predictions made by an AI model (Elshawi, Al-Mallah & Sakr, 2019). The two terms are sometimes applied interchangeably, however, interpretability typically is more often used in the context of classifying models that are inherently understandable, such as decision trees, while explainability often involves post hoc methods that aim to make complex, opaque models more comprehensible (Elshawi et al., 2019; Ribeiro et al., 2016). LIME (Local Interpretable Model-agnostic Explanations) is a method that explains the predictions of complex, opaque models (such as sophisticated deep learning models) by creating a simpler, interpretable model that can then be applied to at least partially explain the prediction of the complex model (Ribeiro et al., 2016). It does this by slightly altering the input, observing how the complex model's prediction changes, and then utilizing these changes to create a straightforward model highlighting which features were most important for the prediction (Ribeiro et al., 2016).

In this study, explainability first refers to the use of intuitively understandable features for the task of popularity prediction. These include features that a human can easily grasp, such as the length of a text. Subsequently, this work explores models with varying levels of explainability and interpretability. These models utilize a combination of the explainable features generated in this study and features that are less inherently understandable, such as embeddings. The aim is to provide a comprehensive overview of different levels and approaches to explainability and interpretability in predicting post popularity.

CHAPTER 4

Features for Popularity Prediction

4.1 Literature Review

Predicting the popularity of online user-generated content, particularly posts involving upvotes and downvotes, has attracted significant attention in recent years due to the increasing use of social media and other content platforms. To ensure a comprehensive understanding of the various approaches and methodologies used to predict content popularity, a semi-structured literature review was conducted. This review largely follows the systematic literature review approach outlined by Kitchenham (2004), which provides a rigorous and reproducible method for identifying, evaluating, and synthesizing relevant studies in a field of interest. In the context of this study, the field of interest involves identifying features generated from user posts that can be utilized to predict their popularity.

The literature gathering process was divided into two strategies.

4.1.1 Gathering Stage

Gathering Method 1: Identifying Dataset-Specific Studies

In the first phase, Google Scholar was used to identify all papers that cited the original paper introducing the dataset utilized in this study. A total of 66 papers were collected in this manner.

Gathering Method 2: Broader Literature Search

In the second phase, a broader search was conducted to explore general research on predicting content popularity. For this, the ProQuest database was selected due to its ability to facilitate precise search queries and systematic filtering of results. The following search query was used to find relevant studies:

```
TIAB(('posts' OR 'postings' OR 'text' OR 'textual') AND
('news' OR 'reddit' OR 'twitter' OR 'facebook') AND
('prediction' OR 'predict' OR 'predictive' OR 'machine learning')
AND ('popularity' OR 'classification' OR
'detection' OR 'sentiment' OR 'polarity') AND
'features')
```

The keyword `TIAB` was used to limit the search to the title and abstract, ensuring that only papers directly relevant to the topic were gathered. The search query was constructed in several stages. First, textual data, including synonyms such as 'postings' and 'texts', was targeted. Next, the environment in which the posts occur, ranging from news websites to popular social media platforms, was captured to expand the potential field of investigation. Although news websites represent a smaller field, they may still provide inspiration for transferable features. Predictive tasks were then prioritized, including not only popularity prediction but also related fields such as sentiment analysis and hate speech detection, to yield potentially useful feature sets. Finally, the inclusion of 'features' in the query ensured the collection of papers directly relevant to the extraction and use of features in prediction tasks.

By September 2024, this search query returned a total of 317 papers. These were subsequently filtered according to the inclusion criteria outlined below.

4.1.2 Inclusion Criteria & Analysis

The inclusion criteria for selecting papers were:

- Peer-reviewed scholarly journals to ensure a high standard of quality,
- Written in English,
- Published between January 2014 (2014/01/01) and September 2024 (2024/09/01) to ensure recent, up-to-date research (and as anyway more than 95 percent of the found studies were written in this time).

After applying these criteria, the search results were narrowed down to 173 papers, mainly because 130 of the original results were working papers and therefore excluded.

In total, 247 papers (66 from Gathering Method 1 and 181 from Gathering Method 2) were further analyzed based on their titles and abstracts to assess their relevance to the topic. Papers that explicitly listed the features used for prediction and had at least a remotely similar use case, as for example predicting the number of retweets or generating - where features that could be potentially be recycled could appear - were used for further analysis. Furthermore, this study focused on finding the explainable features, more complex features such as embeddings, are only briefly mentioned, as they are not the core focus of this thesis.

4.1.3 Findings and Synthesis

In the final step, the information from all the papers was analyzed and then summarized in the following overview, which groups all named features and lists those papers that mentioned the respective feature. The features are grouped into the following categories: Lexical, Sentiment, Syntactic, Surface-Level, Context, and Multidimensional (although the categorization can be partially fluid, with overlaps between categories).

Lexical Features

These features relate to vocabulary and word usage in the text. These features are divided into three categories: **individual word-level features**, **summarized word-level features**, and **rule-based metrics**. The first set contains the following approaches:

- **BOW:** This feature counts occurrences of individual words, also called unigrams (from the Greek **uni** - one - and - **grámma** - written character/text unit).
  Cited by 9 papers: (ALSaif & Alotaibi, 2019; Ambroselli et al., 2018; Chew et al., 2021; Dixit & Soni, 2024; Kamran, Alghamdi, Saeed & Alsubaei, 2024; Khanday, Wani, Rabani, Khan & Abd El-Latif, 2024; Mujahid et al., 2024; Ranathunga & Liyanage, 2021; Sandrilla & Devi, 2022)
- **N-grams:** N-grams are unigrams or combinations of 2 words (bigrams, such as 'thank you'), 3 words (trigrams), or any number of words (n). These words often appear together in specific contexts, providing meaningful insights.
  Cited by 12 papers: (A. M. Ali, Ghaleb, Al-Rimy, Alsolami & Khan, 2022; ALSaif & Alotaibi, 2019; Ambroselli et al., 2018; Assenmacher et al., 2021; Burnap & Williams, 2015; Chew et al., 2021; Dixit & Soni, 2024; Kamran et al., 2024; Kavitha & Akila, 2024; Mossie & Wang, 2020; Ranathunga & Liyanage, 2021; Sarsam, Al-Samarraie, Alzahrani & Wright, 2020)
- **Term frequency–inverse document frequency (TF-IDF):** This measure evaluates word importance relative to other documents. Words that appear frequently in a post but infrequently across all posts receive a high score.
  Cited by 17 papers: (A. M. Ali et al., 2022; Alkomah, Salati & Ma, 2022; Assenmacher et al., 2021; Chew et al., 2021; Dixit & Soni, 2024; Geetha, Karthika, Sowmika & Janani, 2021; Häring, Loosen & Maalej, 2018; Kamran et al., 2024; Kavitha & Akila, 2024; Khanday et al., 2024; Mossie & Wang, 2020; Mujahid et al., 2024; Ranathunga & Liyanage, 2021; Sandrilla & Devi, 2022; Sarsam et al., 2020; Thilagam et al., 2023; Wiedemann et al., 2018)
- **NER:** can capture the mentions of entities such as organizations, locations, or people in a post, by applying, for example, a one-hot encoding of the entities (Eder, Krieg-Holz & Wiegand, 2023). It can also measure how many NER instances certain posts share, contributing to constructing a similarity metric, as done by Haneczok und Piskorski (2020).
  Cited by 2 papers: (Eder et al., 2023; Sarsam et al., 2020)

The next set of features often summarizes a range of different words to create combined features such as the combined frequencies of certain vocabulary:

- **Toxic and explicit terms:** This feature quantifies the usage of negative or swear words in a text, sometimes referred to as a profanity counter (Stemmer, Parmet & Ravid, 2022). Such a counter may also include explicit sexual language (Singh, Ghosh & Sonagara, 2021).
  Cited by 4 papers: (Arunthavachelvan, Raza & Ding, 2024; Jain, Gopalani & Meena, 2024; Singh et al., 2021; Stemmer et al., 2022)

- **Personal pronouns:** The frequency of personal pronouns words such as 'we', 'he', and 'I' can be analyzed, with potential subdivisions into first- and third-person pronouns, as discussed by Eder et al. (2023).
  Cited by 6 papers: (Alkomah et al., 2022; Eder et al., 2023; Jain et al., 2024; Risch & Krestel, 2020a; Singh et al., 2021; Stemmer et al., 2022)

- **Function words:** The usage of function words, including particles, prepositions, auxiliary verbs, and modal verbs.
  Cited by 3 papers: (Pérez-Landa, Loyola-González & Medina-Pérez, 2021; Risch & Krestel, 2020a; Singh et al., 2021)

- **Stop words** This feature assesses the number of irrelevant terms or filler words in a text, which may be calculated as a ratio as well.
  Cited by 2 papers: (Jain et al., 2024; Singh et al., 2021)

The last set applies a rule-based approach to construct relevant features:

- **Lexical diversity:** can be measured, for example by the number of unique words relative to the total text (Jain et al., 2024). Another idea would be counting the number of synonyms used or how often the same words are applied for identical entities.
  Cited by 1 paper: (Jain et al., 2024)

- **Consistency:** measures how similar the title of a post is to its content, measured for example by checking whether the words in the title repeat or match the words in the body, as demonstrated in (Ma, Chen, Chen & Huang, 2022).
  Cited by 1 paper: (Ma et al., 2022)

Sentiment Features

- **Sentiment / Polarity:** These features capture the emotional tone and sentiment conveyed by the author of the text. A text can be classified as positive, negative, or neutral. These basic sentiments can then be further divided into subcategories, such as anger, anxiety, and sadness, as shown in (Arunthavachelvan et al., 2024). Individual word sentiments can be obtained from dictionaries. The polarity of a text can be captured in a number of different ways, the most common being sentiment categories or, in cases of ambiguity, as the ratio of positive to negative words (Geetha et al., 2021).
  Cited by 12 papers: (Alkomah et al., 2022; Arora et al., 2023; Arunthavachelvan et al., 2024; Burnap & Williams, 2015; Eder et al., 2023; Geetha et al., 2021; Häring et al., 2018; Khanday et al., 2024; Li, Chen & Zhang, 2021; Risch & Krestel, 2020a; Sarker et al., 2017; Stemmer et al., 2022)

A subcategory of sentiment analysis, which can also be analyzed as an individual feature, is the following:

- **Emojis usage:** Measured for example in the frequency or proportion of emojis in a text relative to standard characters. Emojis may be classified as positive or negative and be useful in identifying the emotion of a text (González-Ibánez, Muresan & Wacholder, 2011; Sarsam et al., 2020).
  Cited by 6 papers: (Alkomah et al., 2022; Chew et al., 2021; Eder et al., 2023; González-Ibánez et al., 2011; Sarsam et al., 2020; Stemmer et al., 2022)

Surface-level Features

Capture basic, measurable aspects of text, without exploring deeper meaning:

- **Text length:** measured, for example, in total or unique word count in the post or the number of characters. Additionally, counting sentences or syllables (using dictionaries such as LIWC) is possible, as suggested in (Jain et al., 2024). Metrics may include stop words or exclude them, as done in (Pérez-Landa et al., 2021)
  Cited by 10 papers: (Arora et al., 2023; Chew et al., 2021; Jain et al., 2024; Khanday et al., 2024; Ma et al., 2022; Mehravaran & Shamsinejadbabaki, 2023; Pérez-Landa et al., 2021; Risch & Krestel, 2020a; Sarker et al., 2017; Stemmer et al., 2022)
- **Punctuation & special character counts:** such as the frequency of punctuation marks or special characters, such as periods, hashtags, or question marks (for example to identify the number of questions as done by Häring et al. (2018)).
  Cited by 11 papers: (Alkomah et al., 2022; Burnap & Williams, 2015; Chew et al., 2021; Eder et al., 2023; Genç & Surer, 2023; Häring et al., 2018; Jain et al., 2024; Ma et al., 2022; Nesi, Pantaleo, Paoli & Zaza, 2018; Pérez-Landa et al., 2021; Stemmer et al., 2022)

Syntactic Features

These features relate to sentence structure and grammatical composition:

- **Part of Speech (POS) Tagging:** his technique identifies the grammatical roles of words and can be employed to analyze, for example, the relative frequency of specific POS tags, as demonstrated in (Eder et al., 2023).
  Cited by 11 papers: (S. F. Ali & Masood, 2024; Arunthavachelvan et al., 2024; Dixit & Soni, 2024; Eder et al., 2023; Geetha et al., 2021; Jain et al., 2024; Li et al., 2021; Pérez-Landa et al., 2021; Ranathunga & Liyanage, 2021; Sarsam et al., 2020; Thilagam et al., 2023)
- **Parse tree height:** This feature measures the average height of parse trees, providing insights into sentence complexity.
  Cited by 1 paper: (Eder et al., 2023)
- **Tense:** This feature captures the occurrence of grammatical structures associated with past, present, or future tenses.
  Cited by 1 paper: (Singh et al., 2021)
- **Sentence / word length and density:** measured for example in the average or variance of sentence and word lengths, which can contribute to readability metrics or be treated as independent features. Another example would be measuring the word density (e.g., the number of words per 100 characters).
  Cited by 7 papers: (Arora et al., 2023; Eder et al., 2023; Häring et al., 2018; Jain et al., 2024; Risch & Krestel, 2020a; Sarker et al., 2017; Singh et al., 2021)

Context Features

These features do not directly deal with the post itself but rather its context and the circumstances under which it created:

- **Publication time:** captures when the content was posted.
  Cited by 3 papers: (Ambroselli et al., 2018; Burnap & Williams, 2015; Häring et al., 2018)
- **Publication time rank:** AThis indicates the order in which the content was published (e.g., as the second post or the 105th).
  Cited by 1 paper: (Häring et al., 2018)
- **Quote/reply:** identifies whether the new post quotes previous content or is a response to another post.
  Cited by 1 paper: (Häring et al., 2018)
- **Environmental information:** includes circumstances at the time of posting, such as temperature, season, and humidity, that may for example impact the mood of users.
  Cited by 1 paper: (Ambroselli et al., 2018)

- **Competing content:** assesses, for example, the number of similar posts (under an article) available at the time of publication.
  Cited by 1 paper: (Ambroselli et al., 2018)
- **Author information:** encompasses various details about the user or publisher, such as follower count (Mehravaran & Shamsinejadbabaki, 2023), post volume (Stemmer et al., 2022), account creation date (Chew et al., 2021), and common topics of discussion (Chew et al., 2021).
  Cited by 5 papers: (Ambroselli et al., 2018; Chew et al., 2021; Mehravaran & Shamsinejadbabaki, 2023; Nesi et al., 2018; Stemmer et al., 2022)

Multidimensional Features

The following features combine a number of the following characteristic just named and are hence grouped in this new category:

- **Text patterns:** The use of custom regular expressions to identify patterns that may convey particular meanings. For example, the presence of long words (longer than n characters) and short words (fewer than 4 characters) can be analyzed, as shown in (Jain et al., 2024). Another example would be to check whether a text starts with a letter, or a number (Ma et al., 2022). Combinations such as '!!' may signify a certain tone that is associated with clickbait, as presented in (Chakraborty, Paranjape, Kakarla & Ganguly, 2016; Genç & Surer, 2023). As text pattern can be defined for almost all categories just named, they are listed as a multidimensional feature.
  Cited by 4 papers: (Genç & Surer, 2023; Häring et al., 2018; Jain et al., 2024; Ma et al., 2022)
- **Formality:** the degree of formality present in the text, by writing sentences in a formal structure (syntax) or counting formal words and expressions (like 'Dear Sir' which would be more lexical).
  Cited by 1 paper: (Eder et al., 2023)
- **Capitalization:** The usage of lowercase or uppercase letters, which may be employed for emphasis (e.g., 'THIS IS COMPLETELY WRONG!'). Other examples are the ratio between upper and lowercase letters or the count of fully capitalized words.
  Cited by 3 papers: (Alkomah et al., 2022; Eder et al., 2023; Häring et al., 2018)
- **Readability metrics:** Metrics such as the Automated Readability Index (ARI), introduced by Smith und Senter (1967), that assess the complexity of a text by examining factors like average characters per word and sentence length. Or the Flesch Reading Ease formula by Flesch (1948), which evaluates average sentence length and syllables per word, is another example.
  Cited by 5 papers: (Arora et al., 2023; Chew et al., 2021; Eder et al., 2023; Jain et al., 2024; Risch & Krestel, 2020a)

- **Gender and group identification:** This feature assesses whether the post references specific genders or groups, identifiable through the use of terms like 'he' or 'she' (e.g., in 'She is sooooo pretty') using a POS tagger (Li et al., 2021).
  Cited by 2 papers: (Geetha et al., 2021; Li et al., 2021)

- **Topic modeling:** This technique identifies latent themes, commonly referred to as topics, using methods such as Latent Dirichlet Allocation (LDA).
  Cited by 3 papers: (Ambroselli et al., 2018; Kavitha & Akila, 2024; Zosa, Shekhar, Karan & Purver, 2021)

Other Features

Embeddings are mentioned here as a special category due to their comparatively lower explainability compared to other features. They represent high-dimensional contextual representations of words or phrases and are increasingly employed in text data modeling.

- **Word Embeddings:** transform words into dense vector representations based on their context. These embeddings can be pretrained or specifically trained for a task. Notable examples include Doc2Vec, Word2Vec (developed by Google), GloVe (from Stanford), and fastText.
  Cited by more than 10 papers: (Ambroselli et al., 2018; Assenmacher et al., 2021; Kamran et al., 2024; Mossie & Wang, 2020; Ranathunga & Liyanage, 2021; Sandrilla & Devi, 2022; Thilagam et al., 2023; Wiedemann et al., 2018)

Insights

A notable insight from the literature review is the divergent approaches to feature selection during preprocessing. In some studies, specific parts of a text are removed without any assessment of their significance, while in others, these same parts are seen as valuable features. For example, in (Li et al., 2021), punctuation, numbers, and emoticons are discarded, whereas many other studies utilize these elements in model training. Another common practice is the removal of stop words; for instance, while some studies advocate for their exclusion, others highlight the importance of the ratio of filler words to relevant words, noting that fake news articles tend to contain fewer stop words than credible ones (Singh et al., 2021). This suggests a potential lesson: features such as filler words should not be discarded prematurely but rather evaluated and properly encoded as new features before determining their relevance.

A further insight from the literature review is that the identified features naturally vary across use cases and generally exhibit a high degree of variability. Moreover, it is evident that the same aspects of a text can be measured in numerous ways. Often, it remains debatable which method is optimal, as this may also be project-specific.

4.2 Explainable Features for Popularity Prediction

Determining the right data to feed into a statistical model to predict a certain outcome is a central task in NLP. Raw data from datasets is often insufficient for effective modeling without transformation. As a result, the literature discusses approaches to obtain the right input for prediction, such as *feature engineering*, which involves selecting the right subset of informative features or combining existing features to create new ones (Garla & Brandt, 2012). Or *feature generation* by utilizing domain-specific and common-sense knowledge (Gabrilovich & Markovitch, 2005).

Based on findings from the literature review and the requirements of the dataset and use case, the following features were identified as potential interpretable predictors for popularity:

Category	Feature Name	Description
Lexical	Top 200 TF-IDF	Words with the highest TF-IDF scores.
	Top 150 N-gram	Most common two-word, three-word and four-word combinations (bigrams, trigrams and fourgrams)
	Top 50 Named Entities	Most common entities, one-hot encoded.
	ProfanityFreq	Frequency of swear words.
	DiversityRatio	Ratio of unique words to total words.
	FirstSingFreq	Frequency of first-person singular pronouns (*ich, mich*).
	FirstPluralFreq	Frequency of first-person plural pronouns (*wir, uns*).
	SecondPluralFreq	Frequency of second-person plural pronouns (*ihr, euch*).
	ModalObligationFreq	Frequency of modal verbs indicating obligation (*muss, darf, soll*).
	ModalPossibilityFreq	Frequency of modal verbs indicating possibility (*kann, will*).
Sentiment	SentimentScore	Sentiment polarity score of the text. On a spectrum of 1 (completely positive) to 0.5 (neutral) to 0 (completely negative).
	StronglyPositive	Texts with a sentiment score >0.9
	StronglyNegative	Texts with a sentiment score <0.1
	EmojiPositiveFreq	Frequency of positive emojis (e.g., :)).
	EmojiNegativeFreq	Frequency of negative emojis (e.g., :( ).
	EmojiSurpriseFreq	Frequency of surprise emojis (e.g., :-o).
	EmojiSarcasticFreq	Frequency of sarcastic emojis (e.g., ;)).

Continued on next page

Category	Feature Name	Description
Surface-Level (Text Length)	CharCount	Total number of characters in the text.
	SyllableCount	Total number of syllables in the text.
	WordCount	Total number of words in the text.
	UniqueWords	Number of unique words in the text.
	PolysyllableCount	Total number of polysyllabic words.
	WordsPer100Chars	Number of words per 100 characters.
	SentCount	Total number of sentences in the text.
	AvgSentLength	Average sentence length (in words).
	TitleToBodyRatio	Ratio of title length to body length.
	AvgWordLength	Average word length (in characters).
Surface-Level (Symbol-Based)	ExclaimFreq	Frequency of exclamation marks (`!`).
	QuestionFreq	Frequency of question marks (`?`).
	PeriodFreq	Frequency of periods (`.`).
	QuoteFreq	Frequency of quotation marks (`'` or `"`)
	DigitsFreq	Frequency of numeric digits.
	PunctCount	Total number of punctuation marks.
	PunctToTextRatio	Ratio of punctuation to total characters.
	CapLetterFreq	Frequency of capital letters.
	FullCapsFreq	Frequency of fully capitalized words.
Syntactic Features	AvgParseTreeHeight	Average height of syntactic parse tree.
	ConjunctiveFreq	Frequency of conjunctive verb forms (*hätte, würde, könnte, sollte, etc.*).
	PastTenseFreq	Frequency of verbs in past tense (*war, machte, kamen, etc.*).
Contextual Features	SimilarityArticleBody	Cosine similarity between the post and the article body.
	SimilarityArticleTitle	Cosine similarity between the post and the article title.
Multi-dim. (Readability)	ARIScore	Using the ARI formula.
	SMOGScore	Using the Simple Measurement of Gobbledygook (SMOG) formula.
	FleschEaseScore	Using Flesch Reading Ease formula.

Continued on next page

Category	Feature Name	Description
Multi-dim. (Rule-Based)	LongWordsFreq	Frequency of Custom pattern to recognize long words
	LongWordComboFreq	Custom Pattern - sequence long words.
	ShortWordsFreq	Custom Pattern - short words
	ShortWordsFreq	Custom Pattern - sequence short words.
	ConcisePhraseFreq	Custom Pattern - Stylistic choice
	RepeatedWordsFreq	Custom Pattern - Repetition of a Word
	ExaggerateFreq	Custom Pattern - Exaggeration
	AffectionFreq	Custom pattern - positive affection
	WonderingFreq	Custom Pattern - personal wondering
	DesireFreq	Custom Pattern - personal wishes
	OnlySymbolsFreq	Custom Pattern - low-afford text
	OnlyWordsFreq	Custom Pattern - low-afford text
	NoCapsShortFreq	Custom Pattern - low-afford text
	OnlyLinkFreq	Custom pattern - containing only link
	FormalityFreq	Custom Pattern - formality
	RandomFeature1	Random feature used for testing the validity of the approach. Randomly oscillating between 0 and 300.
	RandomFeature2	Randomly oscillating between 0 and 1.
	RandomFeature3	Randomly binary (0 or 1).

Table 4.1: Complete Feature Table for Text Analysis

4.2.1 Custom Patterns

To capture specific, subtle characteristics within the dataset, this work introduces a set of custom regular expression (regex) features tailored to the use case. So-called regex patterns provide a method to search a text for specific sequences of characters (Thompson, 1968). These conditions can include quantifiers (such as 'one or more') and logical operators (such as 'and' or 'or') (Thompson, 1968). A simple example would be a sequence starting with an 'a' or 'b', followed by one or more whitespaces and another 'a'. This pattern can be expressed in regex terms as: '(a|b)\w+a'. The features introduced in this work are inspired by existing literature and patterns observed in the data itself.

To quickly check whether a designed pattern is a possible useful predictive feature, algorithm 4.1 was developed. It assesses if a pattern is likely to have a positive effect on the number of upvotes (more upvotes), a neutral effect, or a negative effect, how strong this effect may be and how many posts contain the specific pattern. Finally, it collects examples of the most prevalent case, which can later be analyzed and visualized:

Algorithm 4.1: Fast Text Pattern Engagement Estimate

Input: Dataset **D** with columns: `FullText`, `UpVotes`, `article_vote_median_tc`*,
Dictionary **P** with patterns to evaluate
Output: Results **R**: Engagement metrics and examples for each pattern

foreach *pattern* $p \in$ **P do**
  Filter **D** to rows where `FullText` matches p;
  if *no rows match* **then**
    Skip to next p;
  foreach *post in filtered rows* **do**
    if `UpVotes` > `article_vote_median_tc` **then**
      Assign *Votes_Category* = *Higher*;
    else if `UpVotes` < `article_vote_median_tc` **then**
      Assign *Votes_Category* = *Lower*;
    else
      Assign *Votes_Category* = *Neutral*;
    end
  end
  Count occurrences of each *Votes_Category*;
  Identify the *Majority_Category* (most frequent);
  Extract up to 3 example posts matching the *Majority_Category* for pattern p;
  Store results for p, category counts, *Majority_Category*, and examples;
end
Sort **R** by *Majority_Category* and engagement ratios;
return R

*article_vote_median_tc stands for the median number of upvotes of a group of 10 posts written in the comment section of the same article at approximately the same time.

Pattern: 'LongWords'

When analyzing the data, more complex texts appeared to receive more upvotes. To explore this, a regex pattern was created to identify words with over 17 characters (including German 'umlauts'), possibly signaling detailed or sophisticated language:

```
\b[A-ZÄÖÜa-zäöüß]{17,}\b
```

Figure 4.1 shows an exemplary post from the dataset where the regex pattern matches such words, highlighted in blue. The title of the corresponding article for each post is shown above. Its upvotes, along with the median upvotes for that article's comment section, are shown below. Matches include words like 'Einbürgerungstests' (naturalization tests), reflecting more complex language.

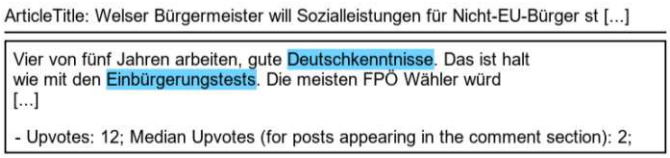

Figure 4.1: Example of a Post Containing Long Words

Pattern: 'LongWordCombo'

Similar to the previous pattern, this one detects parts of a text that might indicate a more sophisticated use of language. It identifies posts containing a combination of two long words (each at least 13 characters) in succession. The threshold for word length is lower than in the previous pattern to account for the fact that such combinations are much rarer than single long words. The regex pattern is defined as '\b', marking the boundary of a word, followed by the first words, separated by one or more spaces or a dash, and then followed by the second long word, and the final word boundary:

```
\b[A-ZÄÖÜa-zäöüß]{13,}(\s|-)+[A-ZÄÖÜa-zäöüß]{13,}\b
```

Figure 4.2 illustrates an example of this pattern in the dataset. The phrase 'beispiellose Menschlichkeit' (unparalleled humanity) demonstrates the use of more complex language, which is likely to appeal to an audience that values intellectual or nuanced content.

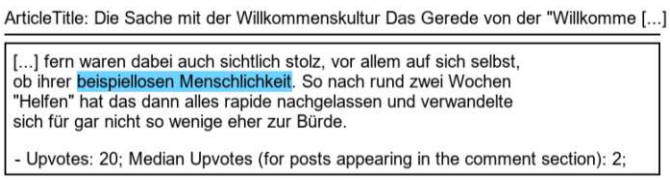

Figure 4.2: Example of Two Long Words in Succession Within a Post

Patterns: 'ShortWords' and 'ShortWordCombo'

Analogously to the long word patterns, the patterns 'ShortWords' and 'ShortWordCombo'. Short words were defined as having 2-5 characters (for both patterns).

Pattern: 'ConcisePhrase'

This pattern identifies posts with a specific writing style, matching either a word (with 4 or more characters) and exactly one full stop (the ending of another sentence) or the start of the text, followed by one or more spaces, followed by a new word, and ending with exactly one punctuation mark. The regex pattern is:

```
([A-ZÄÖÜa-zäöüß]{4,}\.{1}|^)\s+[A-ZÄÖÜa-zäöüß]{4,}(\.{1}|\,{1})
```

The regex-sequence was created after observing that highly upvoted posts often exhibit a concise and impactful writing style, sometimes associated with a stream-of-consciousness approach. Figure 4.3 provides an example where the user shares a brief thought ('Tragic.'), demonstrating an ability for crafting content that effectively resonates with readers.

ArticleTitle: Prozess gegen inhaftierte ukrainische Pilotin in Russland begonne [...]

[...] eschossen. Dabei sind auch zwei russische Kriegsberichterstatter ums Leben gekommen. Tragisch. Gehört zum Berufsrisiko. Die abgeschossene Pilotin als Kriegsgefangene, und das ist sie nach Völkerrecht, nun des Mordes an den Kriegsberichterstattern anzuklagen, ist ziemlich dämlich.

- Upvotes: 17; Median Upvotes (for posts appearing in the comment section): 4;

Figure 4.3: Example of a Concise Phrase Within a Post

Pattern: 'RepeatedWords'

The next pattern finds repeated words, which can again indicate an attempt to emphasize certain parts of a text. This emphasis might provoke a stronger reaction from readers. In the example shown in Figure 4.4, the user emphasizes that incomplete knowledge does not lead to complete understanding ('Halbwissen plus Halbwissen ergibt kein ganzes Wissen'). The pattern is defined as:

```
((\s|^)+[A-ZÄÖÜa-zäöüß]{6,}(\s|\.|\,)+)[\s\S\r\n]*\1
```

ArticleTitle: Das China-Modell hat ausgedient Die Börsenturbulenzen sind Sympto [...]

[...] bissi Halbwissen über China und über die Wirtschaft. Nur leider, Herr Frey: Halbwissen plus Halbwissen ergibt kein ganzes Wissen. Tut mir leid für Sie!

- Upvotes: 6; Median Upvotes (for posts appearing in the comment section): 1;

Figure 4.4: Example of Repeated Words in a Post

Pattern: 'Exaggeration'

This pattern was created after observing that engaging posts with many upvotes often emphasize certain parts of a sentence (e.g., using capital letters) or reinforce their message with repeated punctuation marks such as '!!!'. These stylistic choices may draw more attention and result in higher upvotes. The pattern is defined as:

```
!{2,}|\?{2,}|[A-ZÄÖÜ]{4,}
```

Figure 4.5 shows two examples of this pattern. In the first example, the user emphasizes disbelief by writing 'REALLY?' in all caps. In the second example, the user highlights that 17 billion is a large number by using repeated exclamation marks ('!!!'). In both cases, the number of upvotes exceeds the median number of upvotes in the respective comment section.

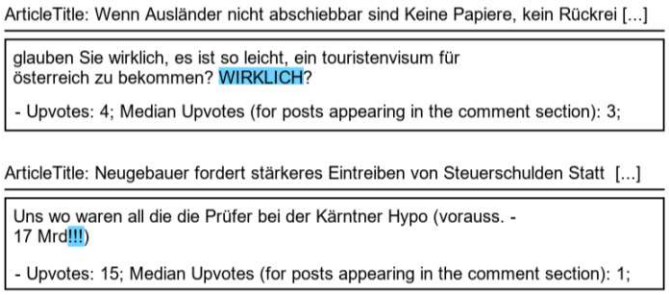

Figure 4.5: Examples of Posts with Exaggeration

Pattern: 'Affection'

The idea behind this pattern is based on the observation that many popular posts often include strongly positive emotional terms, expressing that something is exceptionally great. Figure 4.6 shows an example where a user expresses strong support for Angela Merkel's politics by using the word 'love'. The regex term for this pattern is defined as:

```
\b((liebe)|(wunder(schön|bar|voll)+
|herrlich|grandios|großartig|fantastisch)+(e|r|s)*)\b
```

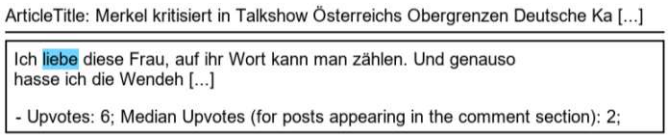

Figure 4.6: Examples of Posts Containing the Pattern of Wondering

Pattern: 'Wondering'

This pattern was designed after noticing that many popular posts include parts where the user expresses curiosity or wonders about something. For instance, users might question a perspective or share their thoughts, making the post more engaging for readers. The pattern is defined as:

```
\b(ich|mich)\s+(wunder(t)*|frage|überlege)\b
```

Figure 4.7 shows two examples of this pattern in posts. In the first example, the user applies the phrase 'I am asking myself' ('ich frage mich') to share their stream of thoughts. In the second example, the user jokes by stating that something does not surprise them ('mich wundert nicht mal'), humorously expressing their thoughts.

ArticleTitle: Krim: Der weite Weg nach Russland Seit März 2014 zählt der Kreml [...]

[...] mation veröffentlicht. Und was hat ihnen der Google Translator verraten? Ich frage mich, warum sie sich die Frage nach der Wahlbeteiligung nicht selbst beantworten können?

- Upvotes: 5; Median Upvotes (for posts appearing in the comment section): 3;

ArticleTitle: Experte: Wohnraum billig vermieten statt renovieren Finanzrechtle [...]

Mich wundert nicht mal, wenn Ihnen die Fabrik davonläuft ....

- Upvotes: 5; Median Upvotes (for posts appearing in the comment section): 1;

Figure 4.7: Examples of Posts containing the pattern of wondering

Pattern: 'Desire'

Another observation drawn from the data is that posts expressing personal desires or wishes often result in more engagement. The following pattern was created:

```
\b(ich)\s+(hoff|wünsch|sehne|verlang)(e)*\b
```

Figure 4.8 illustrates an example where a user expresses the wish that the then-presidential candidate of Austria, Alexander Van der Bellen, will follow in the footsteps of the former president. This example also serves as a reminder of the political relevance of the dataset.

ArticleTitle: Van der Bellen veröffentlicht Spenden Auf der Webseite des Verein [...]

[...] de Minister. Das zeigt, dass man als Bundespräsident schon wirken kann und ich hoffe sehr, dass vdB, wenn er es denn wird, in die Fußstapfen seines Vorgängers tritt.

- Upvotes: 6; Median Upvotes (for posts appearing in the comment section): 1;

Figure 4.8: Example of a Post Containing a Personal Wish

Pattern: 'OnlySymbols'

Finding patterns for low engagement proved more challenging than for engaging posts. Observations suggested that low engagement often stemmed from the absence of engaging elements rather than the presence of features driving low popularity. Following this idea, the 'Only Symbols' pattern was developed, potentially serving as a negative predictor for popularity. It matches strings consisting only of symbols, spaces, or digits, which may indicate posts with little or no meaningful content. Figure 4.9 shows two examples from the dataset. In both cases, the reason for low upvotes is evident—simply writing a question mark or smiley is ambiguous and lacks depth. The regex code is:

```
^[\W\s\d]+$
```

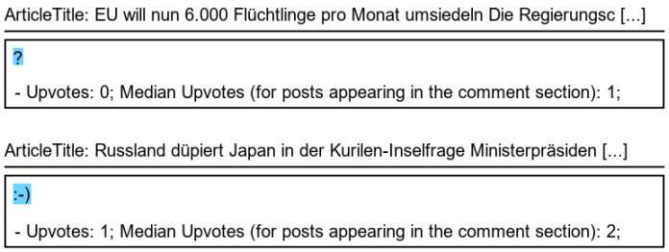

Figure 4.9: Examples of Posts Containing Only Symbols

Pattern: 'OnlyWords'

This pattern also focuses on the absence of engaging elements. Here, it matches strings consisting only of word characters or digits:

```
^[\w\d]+$
```

The pattern is designed to identify low-quality text or posts with minimal information. Figure 4.10 shows two examples matching this pattern. In the first, a single word, 'shaking head' ('kopfschütteln'), provides little information. In the second, the matched text is a rare jargon word, which may not be widely understood by users.

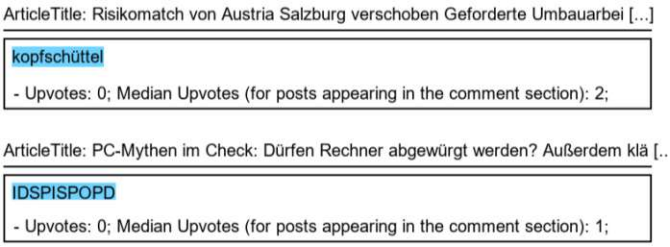

Figure 4.10: Examples of Posts Containing Only Word Characters or Digits

Pattern: 'NoCapsShort'

The next pattern also identifies texts that appear to be written with little effort or seem somewhat careless. Specifically, it targets posts where no capital letters are used, and the total text is relatively short. Figure 4.11 illustrates such an example. The overall writing style seems unmotivated, lacking depth, and grammatically incorrect, making it unlikely to draw significant attention. The regex for this pattern is:

```
^[a-zäöüß\s\W\d]{1,50}$
```

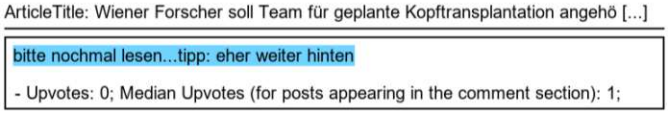

Figure 4.11: Example of a Short Post Containing No Capital Letters

Pattern: 'Uncompassion'

This pattern detects posts that lack strong opinions, often signaled by words like 'okay', 'anyways', and other similar expressions, combined with a short text length (20 characters or less). The pattern is defined as:

```
^(?=.{1,20}$).*\b(eh|schon|naja|ok(ay)*|also|wtf|
lol|\.\.\.|hm|klar|sicher|toll|passt)+\b.*
```

Figure 4.12 illustrates two examples of such posts. In the first example, someone says something like; 'obviously, they are that way' ('Sind sie eh'). The word 'eh', a common Austrian German term, expresses that something is so obvious it is not worth any further discussion, making the comment unremarkable. In the second example, the comment 'okay, that's right' merely repeats or confirms a previous statement without adding value, which explains why these posts attract little attention or upvotes.

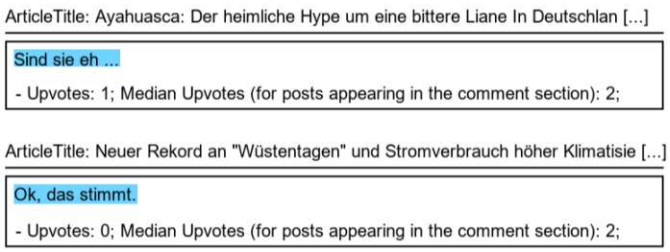

Figure 4.12: Examples of Posts with Uncompassionate Language

Pattern: 'OnlyLink'

The next pattern, which whose core component is described in detail in section 5.3 is simply matching posts that consist only of a link without any additional information about the link - such as the post seen in Figure 4.13. The regex is defined as:

```
^(ht+ps*\s?:*\s?\/+(upload\.|de\.|en\.)*|w+\s?\.r*i*s*\.*|m+\s?\.)+
\s?(\w+-*\w*-*\w*)(.gv|.europa)*\.*\s?
(com|tv|info|co|at|eu|adww|de|ch|uk|org|net|ee)(\s?\/\s?\S*)*$
```

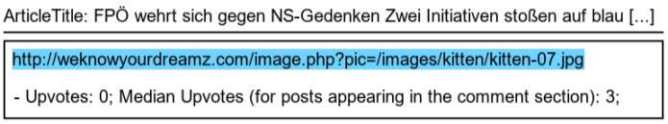

Figure 4.13: Example of Posts Consisting Only of a Link

Pattern: 'Formality'

This pattern identifies the presence of formality in text, which in German is indicated by the 'Höflichkeitsform' (Courtesy form). The most common building blocks are easily recognizable through the use of 'Sie' or variations of the word 'Ihr'. A important characteristic of the German courtesy form is the capitalization of the first letter. The pattern is defined as:

```
\b(Sie|Ihr(e)*(r)*(nen)*)\b
```

Figure 4.14 provides two examples where the formal address ('Anrede') is used. Overall, this pattern seems to negatively impact a post's likelihood of receiving upvotes. A possible explanation is that it often appears as a way to address other users in the forum, as seen in the examples. Since replies generally tend to receive fewer upvotes than original posts, as discussed in section 2.1, this could contribute to the observed trend.

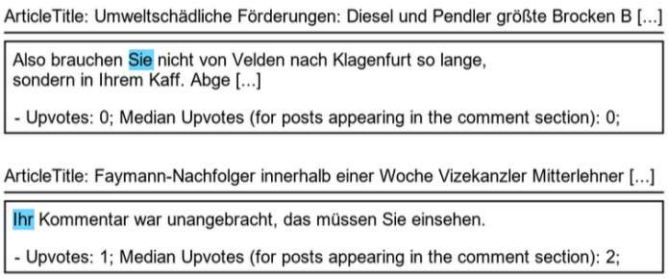

Figure 4.14: Examples of Posts Containing Formal Language

Emoji Patterns

To capture the use of emojis in the dataset—limited to text-based variations such as smilies since proper emojis are not included—four custom regex patterns were developed. These patterns detect different variations of the base forms for positive ':)' (first line below), negative ':(' (second line), surprise ':O' (third line), and sarcastic ';)' (fourth line):

```
(:\s*(-|\^)*\s*(\)|>|D\s*|3)|(\(|<)\s*(-|\^)*\s*:|<\s*3)
(:\s*(-|\^)*\s*(\(|<|8|\[|\|)|\s+(\)|>|\^)\s*(-|\^)?\s*:)
:\s*-*\s*(o|O)+(\s|$|\)|\()+
(;\s*(-|\^)*\s*(\)|>)|(\(|<)\s*(-|\^)*\s*;)
```

Smilies are often used to express emotions. In particular, negative smilies appear frequently when users complain about a topic. For instance, Figure 4.15 shows an example where a user uses the negative smiley to express disapproval of a new law in Austria that imposes taxes on purchasing digital storage devices (e.g., hard drives). Negative smilies, especially when used as in this example, seem to increase the likelihood of upvotes by sharing personal emotions.

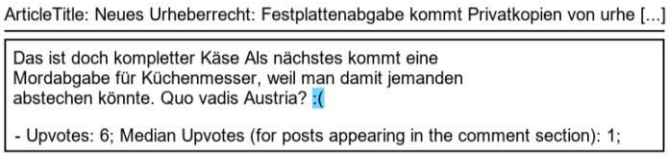

Figure 4.15: Example of a Post Containing a Negative Smiley

Custom Profanity Counter

To address the culture-specific nature of (Austrian) German profanity, a custom regex pattern was developed to create a custom profanity counter. As shown in figure 4.16, such posts often receive more upvotes, likely due to the strong reactions elicited by complaints or frustration. The pattern is defined as:

```
\b((voll)*(idiot|depp|trottel|wappler)(e|n)*|scheiß(e|n)*|
fick(t|e)*|wichs(e|r)*|mist|arsch(l(o|ö)che*r*)*|verpiss(t)*)\b
```

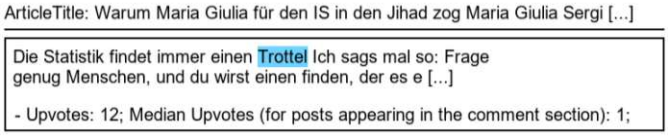

Figure 4.16: Example of a Post Containing a Swear Word

4.2.2 Readability Features

Readability scores essentially assess how complex a text is and how easy or hard it is to digest. To capture the variety and complexity of readability measurements, this work employs three different approaches. Each method focuses on distinct aspects of readability and is calculated using the following formulas:

The first method is the ARI, developed by Smith und Senter (1967), which evaluates the understandability of a text. Originally, the ARI was intended to indicate the U.S. school grade level required to comprehend the text, with a higher score suggesting greater difficulty and a need for a higher grade level (Smith & Senter, 1967). Although initially designed for English texts (Smith & Senter, 1967), the general nature of the formula allows it to be applied to German texts as well, especially since the focus in this work is not on mapping the score to school grades, but rather using it as a general complexity measure. The formula for ARI is as follows (Smith & Senter, 1967):

$$\text{ARI} = 4.71 \times \frac{\text{characters}}{\text{words}} + 0.5 \times \frac{\text{words}}{\text{sentences}} - 21.43 \tag{4.1}$$

The second method is the Flesch Reading Ease score, initially developed by Flesch (1948), which focuses on the average number of words in a sentence and the average number of syllables in words. Since this work deals with German texts, the adjusted formula by Lisch und Kriz (1978) is used:

$$\text{Flesch Ease} = 180 - \frac{\text{words}}{\text{sentences}} - 58.5 \times \frac{\text{syllables}}{\text{words}} \tag{4.2}$$

The third method is the SMOG readability score, developed by McLaughlin (1969), which emphasizes the number of polysyllabic words (in this case defined as words with more than 3 syllables) in relation to the number of sentences. The idea behind this method is that the presence of complex words with multiple syllables contributes to the overall difficulty of the text:

$$\text{SMOG} = 1.043 \times \sqrt{30 \times \frac{\text{polysyllables}}{\text{sentences}}} + 3.1291 \tag{4.3}$$

Together, these readability features provide a comprehensive evaluation of a text's complexity and ease of understanding, each focusing on different linguistic characteristics.

4.2.3 Context Features

As mentioned earlier, literature sometimes suggests context features that do not directly relate to the content of a post but instead refer to factors such as the publication time (when the post was created) or features like the upvotes that authors have previously received for their other posts, as argued by Park et al. (2016). In this work, these types of features are purposefully not considered as predictors but rather as biases. This decision is based on the argument that the content itself should be the primary factor in determining whether a post is engaging.

Thus, the only truly relevant feature explored in this work is the similarity between the post and the article's text and body under whose comment section it was released. For this, cosine similarity, as defined by Madylova und Oguducu (2009), is utilized. We can conceptualize the text of a post as a so-called document, denoted as d, which is a collection of terms or words, as previously described under the BOW model, whose frequency we can count.

Similarly, we can refer to the text we want to compare it with (for example the article title) as q (Madylova & Oguducu, 2009). Both d and q can be represented as vectors, where the vector's dimensions correspond to the terms in the text (Madylova & Oguducu, 2009). The value of each dimension represents the frequency of the corresponding term in the text (Madylova & Oguducu, 2009):

$$\mathbf{d} = (d_1, d_2, \ldots, d_n) \quad \mathbf{q} = (q_1, q_2, \ldots, q_n) \tag{4.4}$$

For example, d_1 represents the frequency of term 1 in the user post, while q_1 represents its frequency in the article's title (Madylova & Oguducu, 2009). Now that we have vectorized the text, we can compute the similarity between the two vectors using cosine similarity (Madylova & Oguducu, 2009):

$$\text{Cosine Similarity}(\mathbf{q}, \mathbf{d}) = \frac{\mathbf{q} \cdot \mathbf{d}}{\|\mathbf{q}\|\|\mathbf{d}\|} = \frac{\sum_{i=1}^{n} q_i d_i}{\sqrt{\sum_{i=1}^{n} q_i^2}\sqrt{\sum_{i=1}^{n} d_i^2}} \tag{4.5}$$

The advantage of cosine similarity is that it is scale-invariant, meaning we can compare both long and short texts without the length impacting the similarity score (Madylova & Oguducu, 2009). This is particularly useful, as the article title and body may differ significantly in length, just as user posts may vary greatly in size.

4.2.4 Occurrences of Features

Figure shows the result of applying Algorithm 4.1 on the filtered dataset, containing 500k posts (as described in section 5.3. Columns 1-3 list the number of posts that have a lower, equal, or higher upvote count than the respective median of the article and time group. Counting all three together gives the total occurrence of the pattern.

Table 4.2: Patterns and Their Distribution Across Categories

Pattern	Higher	Equal	Lower	Majority	Ratio High/Low
Wondering	628	90	318	Higher	1.974843
Affection	2641	453	1464	Higher	1.803962
Desire	1397	270	806	Higher	1.733251
LongWordCombo	10815	2162	7306	Higher	1.480290
Profanity	1333	287	935	Higher	1.425668
RepeatedWords	48134	9868	33937	Higher	1.418334
FirstPlural	29356	5795	20785	Higher	1.412365
ConsicePhrase	5882	1220	4222	Higher	1.393179
SecondPlural	8596	1699	6234	Higher	1.378890
LongWords	56164	11755	41435	Higher	1.355472
FullCaps	54679	11740	41559	Higher	1.315696
ModalObligation	34234	7108	26435	Higher	1.295026
Quote	50669	10923	40011	Higher	1.266377
Exaggerate	18762	4068	14961	Higher	1.254061
Exclaim	40589	8391	32439	Higher	1.251241
ModalPossibility	42341	9348	34288	Higher	1.234864
EmojiNegative	1034	211	844	Higher	1.225118
Digits	53717	11950	44211	Higher	1.215014
Conjunctive	46847	10570	39249	Higher	1.193585
PastTense	31930	7345	27426	Higher	1.164224
FirstSing	66250	15304	59155	Higher	1.119939
Period	188078	44308	172310	Higher	1.091509
ShortWordCombo	218915	52505	207568	Higher	1.054666
CapLetter	203215	48563	193450	Higher	1.050478
ShortWords	227713	55295	222365	Higher	1.024051
EmojiPositive	8495	2216	8525	Lower	1.003531
Question	56852	13949	57977	Lower	1.019788
Formality	19708	5757	23290	Lower	1.181754
EmojiSarcastic	5915	1841	7733	Lower	1.307354
Anglicisms	1102	276	1449	Lower	1.314882
EmojiSurprise	87	26	118	Lower	1.356322
NoCapsShort	5771	2170	11856	Lower	2.054410
OnlyWords	462	220	1356	Lower	2.935065
OnlyLink	249	147	853	Lower	3.425703
Uncompassion	142	71	530	Lower	3.732394
OnlySymbols	55	29	262	Lower	4.763636

An important takeaway from here seems to be that those patterns that have a large high-to-low-ratio, appear relatively rarely.

CHAPTER 5

Experimental Setup for Popularity Prediction

The primary objective of this research is to explore the task of predicting the popularity of posts. This task is subdivided into three main stages:

- **Preprocessing**

  The preprocessing step focuses on transforming the raw data into a form that can be used to extract features for predictive modeling. In the case of deep learning, this step prepares the data for further processing by normalizing and scaling.

- **Feature Generation**

  This step is concerned with generating features that hold predictive power for the task at hand. These features will serve as inputs for the subsequent models used in prediction.

- **Prediction & Evaluation**

  In the final stage, various models are built to predict the popularity of posts, which is the core focus of this research. In a further step these models are then evaluated on new unseen data. Furthermore, the different explainable features are analyszed for their importance.

5.1 Target Variable and Cut-off Value

Popularity is an ambiguous term that can be interpreted in various ways. It could refer to posts that receive the most upvotes or the most votes overall (positive and negative). These two concepts are not always equivalent. Popularity can also be viewed as both regression analysis (Chatterjee & Hadi, 2015), where the response variable is continuous (e.g., the exact number of upvotes), or classification task, where the output is a discrete variable (e.g., whether a post is considered popular or not). Hence to find the right, concrete definition of the task of popularity prediction a data-driven analysis is necessary.

When analyzing popularity in terms of the exact, absolute number of votes a post receives, it can be observed that the distribution of votes received by each post strongly follows a power law distribution, as shown in figure 5.1 below. Meaning that some posts receive the majority of votes, while most receive relatively few interactions. This phenomenon is explained by the ranking algorithm of the *Standard* website, which places those posts with the most votes at the top of the comment section. This process leads to a situation where popular posts accumulate more and more votes, resulting in an imbalance (for example when pinned by a moderator, as outlined in section 2.1). Consequently, determining the exact number of votes as a regression task is not particularly insightful, as the primary reason posts get vast amounts is heavily influenced by the ranking system itself and the internet traffic under a certain article at a certain time.

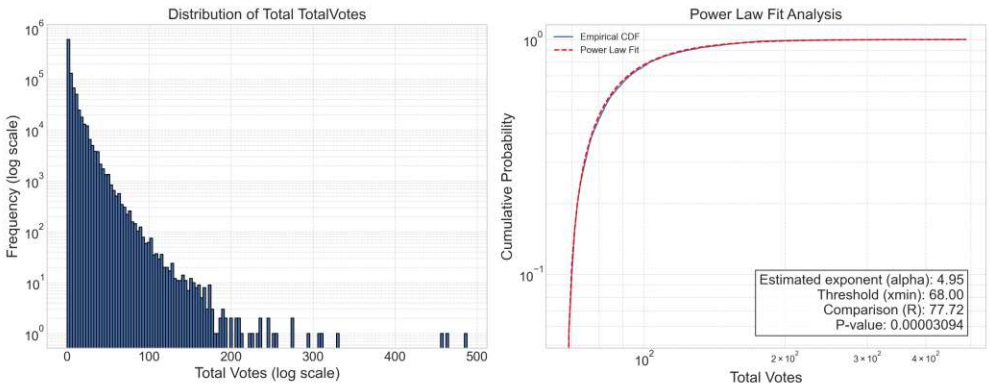

Figure 5.1: Distribution of Total Votes Per Post

Given these considerations, framing the problem as a classification task proves more meaningful. While regression analysis is still conducted for completeness, it is not the primary focus. Rather than predicting the precise number of votes, the main goal shifts to identifying posts with the potential to attract significant attention, regardless of systemic biases introduced by the ranking algorithm. These posts, labeled “engaging posts”, are those that attract a substantial fraction of the total popularity. The remainder, referred to as “regular posts”, are those that generate little to no engagement. This classification approach aligns more closely with the research objectives, as it allows for

a focused examination of the underlying characteristics (e.g., content, writing style, or sentiment) that make a post likely to attract widespread popularity. By framing the task as a classification problem, the study prioritizes the broader objective of understanding the factors that drive engagement, rather than allowing the constraints imposed by the ranking system to dominate the analysis.

When determining whether popularity should be defined by negative votes or positive votes, a data analysis was conducted to base this decision on the characteristics of the use case. Table 5.1 summarizes the vote statistics for the dataset. As shown, the total number of upvotes is significantly higher than the number of downvotes, with nearly four out of five votes being upvotes. Hence, the distribution is heavily skewed.

Table 5.1: Summary of Vote Statistics

Metric	Count
Total Positive Votes	3,758,636
Total Negative Votes	1,056,715
Total Votes	4,815,351

Analyzing the upvote-to-downvote score (where -1 represents only negative votes, 0 represents an equal number of upvotes and downvotes, and 1 represents only positive votes) reveals that the vast majority of posts have more upvotes than downvotes. Among those posts that account for 90% of the total vote interaction about 79% exhibit a positive upvote-to-downvote ratio, as shown in Figure 5.2, while only 17% exhibit a negative score. This indicates that posts with a negative ratio are generally too underrepresented among enging posts to justify an additional classification category. Furthermore, the role of downvotes seems ambiguous, as they might be given for example as a sign of dislike or as a sign of marking wrong information. In light of this and since other researchers like Risch und Krestel (2020a) have questioned their usability as an estimator for the relevance of comments - popularity is defined in terms of positive votes.

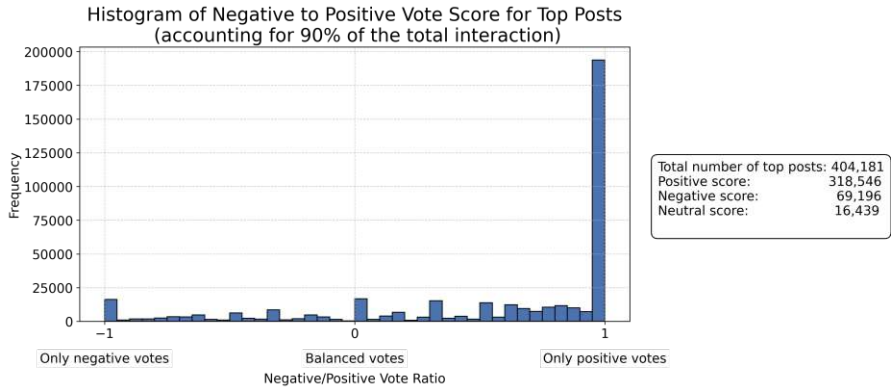

Figure 5.2: Frequency of Upvote-Downvote Scores for Engaging Posts

Cut-off Value

The next step involves determining a cut-off value to differentiate between 'engaging posts' and 'regular posts.' This requires further data analysis. The distribution of total up votes per post follows a power-law distribution, as discussed earlier. Furthermore, the distribution of the number of posts per article appears to follow a power-law pattern, as shown in Figure 5.3. Although the statistical test yields a P-value of 0.51, which is not sufficiently low for strong significance, the R-value of 5.24 and the visual alignment of the data suggest that the power-law remains the best fit for this distribution. It is visible that only a few articles have many posts, while most articles have only a few.

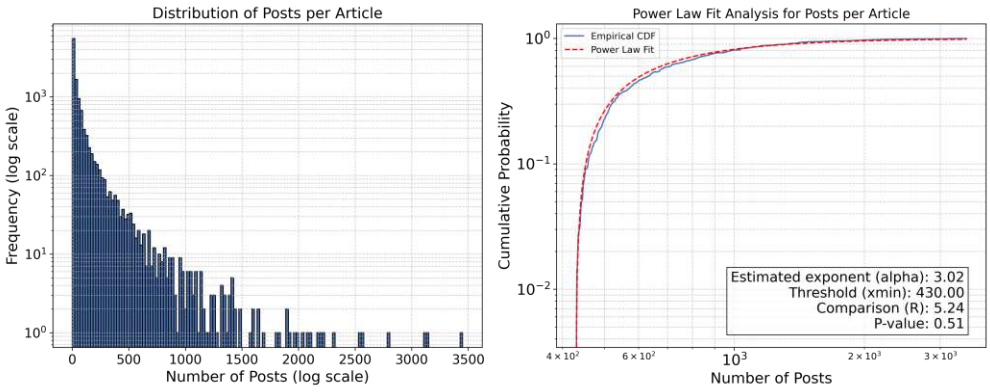

Figure 5.3: Distribution of Number of Posts Per Article

Given these power-law distributions, it is desirable to focus on capturing most of the attention without disproportionately emphasizing only the top posts (because of the issues just discussed earlier). To achieve this, a cut-off was initially set to include posts that contribute to roughly 90% of the total interactions. This ensures that almost all of the attention is captured while excluding posts with very little interaction. Based on the data, around 40% of the posts contribute to this 90% threshold, as shown in Figure 5.4.

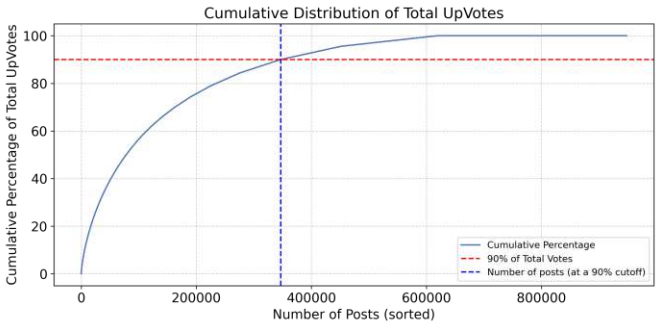

Figure 5.4: Total Votes Cumulative and 90% Interaction Threshold

To classify posts into 'engaging' and 'regular', while ensuring that 'engaging' posts account for most of the total interactions, several approaches exist. A straightforward approach involves ranking all posts by the total votes received and labeling enough top-ranked posts as 'engaging' to reach the threshold. However, this approach disproportionately classifies posts from articles with relatively few posts as 'regular', even if those posts were significant for the respective articles. Instead, it is necessary to label posts according to their relative popularity for each article. Table 5.5 highlights an example of the problem, listing all posts in the comment section of a specific article. Unfortunately, simple statistical thresholds like the median or mean prove to be inadequate to solve this problem. For instance, using the median to classify posts would frequently result in posts with zero votes being labeled as 'engaging', as demonstrated in the example below. Adjusting the threshold to be higher than the median mitigates this issue but risks including posts with minimal interaction. Employing the mean is overly influenced by outliers—posts with disproportionately high vote counts. A solution would be to look at the share of votes a post has and then set a threshold for the cumulative share (like 12%), which leads to the result in the table below.

ID_Article	FullText	TotalInteraction	ShareOfInteraction	CumulativeInteraction	Post_Engagement
103	Herzlichen Dank für den interessanten Tip! Cas...	0	0.000000	0.000000	regular post
103	Ausbildung in Italien gut und schön, aber muss...	0	0.000000	0.000000	regular post
103	Cassata geht mir in Wien auch furchtbar ab. Mi...	0	0.000000	0.000000	regular post
103	"...und der Eisverkäufer ruft "Gelati Gelati"....	0	0.000000	0.000000	regular post
103	Cassata gibt's in der krugerstraße, derzeit mu...	0	0.000000	0.000000	regular post
103	Ich stehe ja auf das Mozart Eis und Raphaello ...	0	0.000000	0.000000	regular post
103	Eis das mit der Spachtel aufgetragen wird ist ...	0	0.000000	0.000000	regular post
103	Laienhaft frage ich nach: Fuer mich wirkt das ...	0	0.000000	0.000000	regular post
103	In Oesterreich gibt es vielleicht 900 Eisgesch...	0	0.000000	0.000000	regular post
103	"Eh, wasse wolle due, hä?"	1	0.166667	0.333333	engaging post
103	Danke für den Tip. Werde dort vorbei schauen.	1	0.166667	0.333333	engaging post
103	Wann gibt es endlich wieder Cassata, Peach-Mel...	4	0.666667	1.000000	engaging post

Figure 5.5: Example of Post Labeling Task (for Comments on Article #103)

However, this does still not solve the problem, as the number of posts is heavily influenced by the time that passed after the posts were written and the order in which they were written, as can be seen in Figure 5.6.

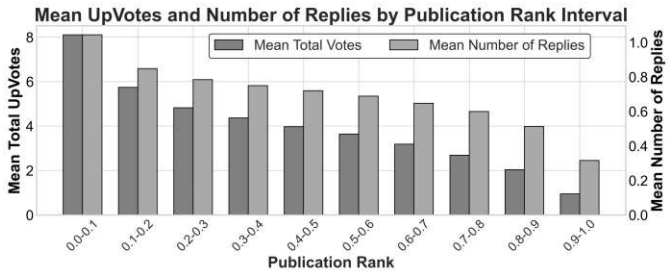

Figure 5.6: Relationship Between Votes Received and Time Passed

Risch und Krestel (2020a), who worked on a post dataset from *The Guardian* with partially similar characteristics, encountered a similar problem. They proposed a solution to address two biases: the **article bias**, where some articles' comment sections receive far more attention than others, and the **time bias**, where posts written later are less likely to gain attention. Their approach operates as follows: they first select the first ten comments in each article's comment section (Risch & Krestel, 2020a). Next, they compute the share of votes a post has within its respective comment section to eliminate the article bias. To address the time bias, they group posts into ten equal-sized groups based on their time rank (e.g., first, second, etc., within their article). Within each time group, posts are sorted by their share of interaction value. Finally, they label the top 10% of posts in each group as *top/flop posts 10%* and the bottom 10% as *top/flop posts 10%*, and do the same for the top/flop 25% and top/flop 50 % (Risch & Krestel, 2020a).

However, this approach overlooks some essential aspects of the problem. First, time bias does not only depend on the publishing rank but also on the time passed after a comment is written. Both of these factors significantly influence the share of votes a post will receive, yet the influences differ in their characteristics, as can be see in Figure 5.7 where they are fitted onto a polynomial regression model.

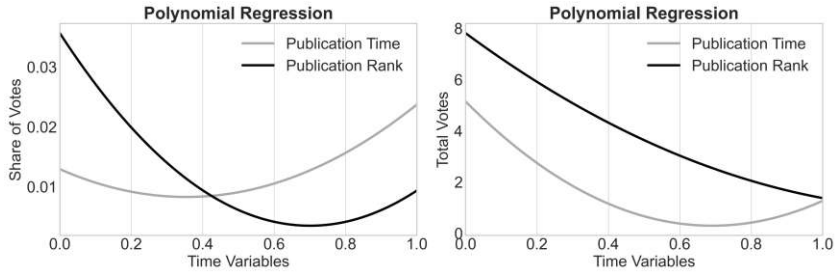

Figure 5.7: Influence of the Time Variables on ShareOfVotes and TotalVotes

Furthermore, their approach relies on the ranking system of *the Guardian* in which the first 10 posts are ranked chronologically and other posts are hidden on the next page. The ranking system of *DerStandard* however works differently than that, as outlined in section 2.1 (with pinned post being displayed and then all the rest). Hence, their solution is unfortunately not directly transferrable. To address these challenges and provide a reproducible algorithm suitable for other use cases, a novel algorithm was developed, which tackles the issues described above and offers a solution to the labeling problem.

To identify the most effective labeling approach, three methods were tested to mitigate the time bias. Initially, no time correction was applied and posts were labeled based on a cumulative upvote threshold (of 0.05), which resulted in a strong point-biserial correlation with publication time of -0.19 (P-value < 0.00001) and with publication rank of -0.27 (P-value < 0.00001). Next, the approach by Risch und Krestel (2020a) was explored, which would have caused an extreme reduction in data as only the first 10 comments per article would have been considered. This led to a decreased correlation with time of -0.065 (P-value: 4.2×10^{-30}) and with publication rank of 1.03×10^{-17} (P-value: 1).

Finally, the labeling algorithm in Algorithm 5.1, inspired by Risch und Krestel (2020a), was introduced. This algorithm accounts for the influence of both publication time and rank on votes, adapting to the nature of the data. Instead of only considering the first 10 posts, it identifies groups of 10 posts within the same article written around the same time, having enough votes as evidence of engagement to make a labeling decision. With this new algorithm, the correlation with publication time is -0.00083 (P-value: 0.55) and with publication rank was -0.0014 (P-value: 0.32). Like in algorithm of Risch und Krestel (2020a) three sets of labels are created (Top/Flop 50, 20 and 10 %). For the final evaluation, the Top/Flop 10 % set is used, since this set is less likely to be affected by edge cases (posts which are neither clearly engaging nor non-engaging).

Algorithm 5.1: Post Filtering and Engagement Labeling

Input: Dataset **D** containing posts
Output: Labeled and filtered dataset **D′**
Create combined time column as (publication time $\times$ 1.5 + rank)/2.5;
foreach *article_ID* **do**
  Create `initial_interval` with 10 bins based on the percentiles of `combined_time`;
  Sort posts by `initial_interval` and `combined_time`;
  foreach *interval with more than 10 posts* **do**
    Split the interval into smaller subgroups by sequentially filling new groups up to 10 posts;
  end
  Define resulting intervals as `time_groups` **tg**;
end
foreach *time_group* **tg do**
  Compute total upvotes, median upvotes, and vote shares for each time group;
  if *median upvotes* $= 0$ ***or*** *total upvotes* ≤ 15 ***or*** *post count* $\neq 10$ **then**
    Drop all posts of **tg** from the dataframe;
end
foreach *time_group* **tg do**
  Sort posts by descending number of upvotes and replies;
  Label first 5 posts as *Top/Flop 50% engaging*;
  Label last 5 posts as *Top/Flop 50% regular*;
  Label first 2 posts as *Top/Flop 20% engaging*;
  Label last 2 posts as *Top/Flop 20% regular*;
  Label posts in positions 3-8 as *Top/Flop 20% undefined*;
  Label first post as *Top/Flop 10% engaging*;
  Label last post as *Top/Flop 10% regular*;
  Label posts in positions 2-9 as *Top/Flop 10% undefined*;
end
Summarize filtering and labeling statistics;
return D′;

Figure 5.8 shows an example of the labeling outcome produced by Algorithm 5.1, in this case, for posts written in the comment section of an article about the German musician Xavier Naidoo. On the left, we have posts labeled as engaging by the algorithm, and on the right, posts labeled as regular. The engaging posts generally exhibit deeper analysis, greater length, and more interesting information. In contrast, the regular posts tend to be more casual and contain less substantial content. These observations, along with other examples, suggest that the labeling produced by Algorithm 5.1 can actually yield meaningful results. However, the examples also highlight the diversity among engaging posts (which might make it difficult to design classification algorithms) —comparing the first two examples, which are lengthy and in-depth, to the last one, which is brief and draws a simple connection between two celebrities.

Xavier Naidoo: Als Jesus einmal zum Alkotest musste Der deutsche Sänger spielt am Sonntag in Wien ein ausverkauftes Konzert. Seinen missionarischen Schlager nährt ein Gebräu aus Verschwörung, Paranoia und Gewalt. Eine Betrachtung

Engaging Posts	Regular Posts
Also sollte der Naive Xaidoo mal wegen irgendwas eingekapselt werden reicht dieser Text als Beweis fuer seine Unzurechnungsfaehigkeit. Meine Highlights: * "Ihr toetet (...) Foeten" * "Warum liebst Du keine Moese?" * "Wo sind unsere Fuehrer?" [...] (4 [...] (13 more words)	the alktest made my day.
Früher fand ich ihn einfach nur lächerlich - mit diesem Pathos, diesen aufgesetzten 'Ich will kein Foto von mir mit einem Preis, da ich nicht möchte, dass die Menschen mich wie einen Götzen anbeten.' Lächerlich, sich selbst zu wichtig nehmend, 'Phili [...] (23 more words)	frag mal deine freundin was sie von ihm hält;-)
Naidoo und Gabalier ja... das passt.	also bitte, vergleichen sie den großartigen rudi schurike nicht mit dieser clownnase.

Figure 5.8: Example of a Labeling Outcome

5.2 Evaluation Methods

This section outlines the methodology used to address the research questions. The evaluation process is divided into two main parts: (1) analyzing whether the explainable features developed in this study differ significantly between engaging and regular posts and (2) building predictive models to assess the explainability and performance of these features.

5.2.1 Feature Importance Testing

In this study, feature importance is assessed in two ways - as the direct influence of explanatory variables on the class - by conducting statistical tests to determine whether the explainable features developed differ significantly between the two classes, while also evaluating the magnitude of these effects. And second - as the importance of explanatory variables within models, in particular the random forests, as discussed in later.

The Mann-Whitney U test, as introduced by McKnight und Najab (2010), is applied as the primary statistical test since most features do not follow a normal distribution. However, since some of the features, such as the stopword-to-text ratio, follow at least partially normal distribution, the independent t-test, as introduced by Kim (2015), is added as an additional metric. Both tests examine whether the observed differences in means between the two classes are statistically significant or could simply be coincidental (Kim, 2015; McKnight & Najab, 2010).

To quantify the magnitude of the differences, Cohen's d, as introduced by (Cohen, 2013), is computed. Cohen's d measures the standardized difference between the means of two groups, providing an intuitive interpretation of effect size (e.g., small, medium, or large effects) (Cohen, 2013). Additionally, point-biserial correlation, as described by Tate (1954) is used to determine the direction and strength of the relationship between individual features and engagement, indicating whether a feature positively or negatively influences engagement. This combination of methods ensures a thorough evaluation of each feature's significance and impact.

Algorithm 5.2 is applied for feature testing. It is a simple custom algorithm created for this use case that extends and combines the ideas of classical feature importance testing approaches such as the Mann-Whitney U and Cohen's d test:

Algorithm 5.2: Feature Importance Testing

Input: Feature set **F**, engagement classes **C**, significance threshold $\alpha = 0.05$
Output: Feature importance results **R**

```
foreach feature f ∈ F do
    Compute means μ_class 0, μ_class 1;
    Calculate absolute and relative differences between classes;
    Perform Mann-Whitney U test and t-test for significance;
    Compute Cohen's d and point-biserial correlation;
    Determine significance based on p < α;
    Store results in R;
end
Sort R by significance and composite ranking;
return R;
```

For each feature, Algorithm 5.2 calculates the mean values for engaging and regular posts, performs statistical tests, and computes effect sizes and correlation values. The results

are ranked based on a composite metric that combines absolute differences, relative differences, P-values, and effect sizes.

5.2.2 Model Training and Evaluation

The dataset is randomly divided into training and test sets, using an 80%-20% split. This split is widely used in literature and ensures a sufficient amount of data for training while preserving enough data for evaluating the model performance on new, unseen data.

Quantitative Analysis

Quantitative metrics are chosen to evaluate classification and regression tasks separately, reflecting the distinct goals of each of the two tasks.

Classification Metrics

Classification aims to distinguish 'engaging posts' from 'regular posts', where engagement is defined based on the interaction threshold described in Section 5.1. The following metrics are employed:

- **Accuracy:** Accuracy measures the overall correctness of the model's predictions and is defined as:

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Predictions}} \tag{5.1}$$

  Accuracy provides a general overview of the model's performance but can be misleading for imbalanced datasets, making it complementary to other metrics.

- **Precision:** Precision measures the proportion of correctly identified engaging posts (true positives) among all posts predicted as engaging. It is defined as:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{5.2}$$

  Precision ensures that posts labeled as engaging genuinely exhibit high interaction levels, minimizing false positives and improving reliability.

- **Recall:** Recall quantifies the proportion of actual engaging posts that were correctly identified. It is defined as:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{5.3}$$

  High recall is essential for identifying as many engaging posts as possible, especially when missing engaging posts would have a high cost.

- **F1 Score:** The F1 score, the harmonic mean of precision and recall, provides a balanced measure of a model's performance:

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5.4}$$

Regression Metrics

While classification is the primary focus, regression metrics are used to evaluate the accuracy of predictions for continuous outcomes, such as the number of interactions. The following metrics are applied:

- **Root Mean Squared Error (RMSE):** Defined as:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \tag{5.5}$$

  RMSE highlights larger errors, making it suitable for tasks where large deviations from the actual values are particularly undesirable.

- **Mean Absolute Error (MAE):** Defined as:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i| \tag{5.6}$$

  MAE provides an intuitive average error magnitude and is less sensitive to outliers than RMSE.

The inclusion of RMSE and MAE ensures a balanced assessment of prediction performance, focusing on both error magnitude and variability.

5.2.3 Qualitative Analysis

In addition to quantitative metrics, qualitative analysis is conducted to contextualize and interpret the results. This involves:

- Analyzing the models themselves and assessing which features they prioritize (features importance within the models.
- Examining individual predictions to understand the model's decision-making process, particularly for posts, where the models make errors.
- Comparing findings with existing literature to situate the results within broader discussions.

The qualitative analysis complements the quantitative evaluation, offering deeper insights into model behavior and its implications for understanding post popularity and interaction patterns.

5.3 Prediction Pipeline

Figure 5.9 illustrates the designed pipeline for preprocessing data, generating features, training the models, making predictions and ultimately evaluating the output.

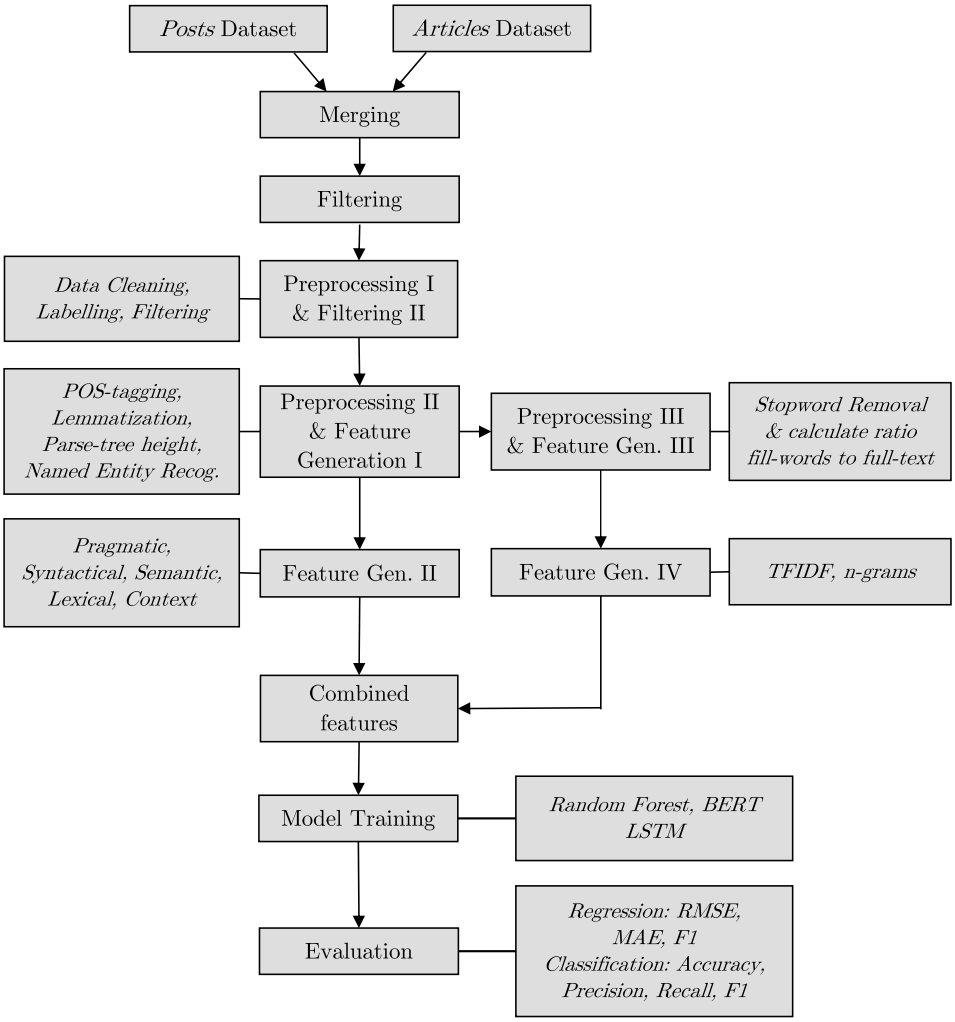

Figure 5.9: Prediction Pipeline: From Dataset Loading to Evaluation

Merging

The preprocessing begins by merging two datasets: one containing information about the posts themselves and another detailing the newspaper articles under which the posts appeared. This merge is crucial as it allows the post-level data to be enriched with the context provided by the associated articles.

Filtering

In this step, posts that are incomplete are removed. Specifically, posts without a body or heading (7 cases) and those associated with articles where no other posts received votes (1257 cases) are excluded from the analysis. This ensures that only posts with sufficient engagement and data are considered for further processing.

Feature Generation I, Labeling, Filtering

Several basic features are created in this step, the most important beeing full text, generated by combining both the article body and heading. To ensure no valuable information is not lost in this step, the ratio of body length to heading length is calculated, as well as the lengths of both individually. In addition, data is labeled and filtered as described in Algorithm 5.1.

Data Cleaning

As the next step, the data undergoes initial cleaning. Since a large portion of the data contains links to external websites, the following custom regex pattern was developed to turn links like the following: `https://www.youtube.com/watch?v=zxPWk-F39P8&list=LL` into texts such as: `youtubeLink`. This approach captures both the fact that there is a link and the site to which it directs, thus allowing for more effective analysis of the data later. The regex pattern developed for this task is:

```
(ht+ps*\s*:*\s?\/+(upload\.|de\.|en\.)*|w+\s*\.|w+\s*\.ris\.*|m+\s*\.)+
\s*(\w+-*\w*-*\w*)
(.gv|.europa)*
\.*\s*(com|tv|info|co|at|eu|adww|de|ch|uk|org|net|ee)'
(\s*\/\s*\S*)*'
```

Group 1: Detects possible prefixes before the website name and spelling version of it. **Group 2:** Matches the website name. **Group 3:** Optionally captures prefixes like `.gv` or `.europa`.**Group 4:** Matches the domain type, such as `.com`, `.org`, or `.net`.**Group 5:** Captures paths or query strings after the domain.

By formulating this regex in the form of groups, we can extract the second match (the website name) and retain it in the text. This way, `www.youtube.com` becomes `youtubeLink`.

Additionally, tokenization issues are addressed by normalizing certain constructions such as `und/oder` by splitting them into `und / oder`, which allows for the later removal of stopwords that are often incorrectly processed by tokenizers.

Parsing Operations: Lemmatization, POS-Tagging & NER Recognition

The text is processed with the SpaCy library to perform tokenization, part-of-speech tagging, and NER. Lemmatization is applied to convert verbs, nouns, and auxiliary words to their base forms (e.g., “ging” becomes “geht”), simplifying words into their roots for improved feature generation. Additionally, named entities are extracted, and the following features are computed:**Average Parse Tree Height**: Represents syntactic complexity by calculating the average depth of the parse tree. **Modal Verb Count**: Captures the frequency of modal verbs (e.g., ’sollte’, ’möchte’) to analyze the intent expressed in the text. **Custom Normalization**: Handles variations like “danke” and “Dankeschön” by normalizing them to a standard form.

The pseudo-code for the parsing and feature generation pipeline is as follows:

Algorithm 5.3: Text Parsing, Normalization and NE Extraction

Input: Text **T**
Output: Normalized text, Average parse tree height, Named entities, Modal verb count
Data: SpaCy language model

```
foreach text in dataset do
    Tokenize text;
    foreach token in text do
        if token POS-tag is VERB, NOUN, or AUX then
            Lemmatize token;
        end
        if token tag is VMFIN then
            Count Modal Verb;
        end
    end
    Extract All Named Entities and their occurrence;
    Calculate Average Parse Tree Height;
    Apply custom normalization with regex;
end
return Normalized text, Parse tree height, Named entities, Modal verb count;
```

An example can be seen in figure 5.10

FullText	CleanedText	AvgParseTHeight	ModelVerbs_Freq	NE_Regierung	NE_Facebook
Es gibt auch andere Möglichkeiten, gegen die schlechte Politik der Regierung zu protestieren.	Es geben auch andere Möglichkeit , gegen die schlechte Politik der Regierung zu protestieren .	2.800000	0	1	0
Facebook wird von vielen genutzt, auch von rechten Islamisten!	Facebook werden von vielen nutzen , auch von rechten Islamist !	2.000000	0	0	1
welche rechten Islamisten haben da kandidiert ?	welche rechten Islamist haben da kandidieren ?	1.285714	0	0	0

Figure 5.10: Example of NER

Stop Word Removal and Feature Creation Next, stopwords are removed from the text, and additional features such as the stopword ratio (the proportion of stopwords relative to total words) are calculated. This helps in reducing noise from the text and allows for more meaningful feature extraction in subsequent steps. The custom stopword list is based on the base German stopword set from the Natural Language Processing Tool Kit (NLTK) [1] and adjusted for the specific use case. An example of the operation can be see in Figure 5.11

Algorithm 5.4: Stop Word Removal and Ratio Calculation

Input: Text $\mathbf{T}$, Custom stopword list $\mathbf{S}$
Output: Cleaned text $\mathbf{T_{cleaned}}$, Stopword ratio

Remove pronouns and digits from text;
Tokenize text into words;
Initialize $stopword\_count = 0$, $total\_words = 0$;
foreach *word in* $\mathbf{T}$ **do**
    if *word is in* $\mathbf{S}$ **then**
        Increment $stopword\_count$;
    end
    Increment $total\_words$;
end
Calculate stopword ratio as $stopword\_ratio = \frac{stopword\_count}{total\_words}$;
Remove stopwords from text and create cleaned text;
return *Cleaned text, Stopword ratio*;

FullText	CleanedText	stopwords_to_words_ratio
Den Newsletter können Sie für die Dauer Ihres Urlaubes nicht deaktivieren, Sie können ihn nur abmelden und nach dem Urlaub wieder anmelden. Wir werden aber Ihre Idee gerne weiterleiten. MfG	Newsletter Dauer Urlaubes nicht deaktivieren , abmelden Urlaub anmelden . Idee gerne weiterleiten . MfG	0.482759
Wissenschaft - Newsletter Habe mich gerade für den newsletter angemeldet, doch der eigentliche Grund dafür wären die wissenschafts Informationen gewäsen, nun frage ich mich ob das noch möglich ist...?	Wissenschaft - Newsletter newsletter anmelden , eigentliche Grund wissenschafts Information gewäsen , fragen möglich ... ?	0.428571
ich kann keinen hinweis finden, wo man sich hinwenden muss, sollte man als abonnent des standard, die zeitung nicht bekommt, ist dass bewusst so arrangiert?	keinen hinweis finden , hinwenden muss , abonnent standard , zeitung nicht bekommen , bewusst arrangieren ?	0.392857

Figure 5.11: Example of Stop Word Removal

[1] https://www.nltk.org/

Feature Generation II At this point, additional features such as the number of words, sentiment scores, and specific token counts are generated. Advanced feature extraction methods such as TF-IDF and n-grams (bigrams, trigrams, and fourgrams) are also applied. These features help in capturing the underlying patterns in the text that are indicative of post popularity.

Train-Test Split The dataset is then split into training and testing sets, ensuring that the model will be evaluated on unseen data. This step is crucial for preventing data leakage during the feature generation process.

TF-IDF and N-gram Feature Generation After splitting the data, the TF-IDF and n-gram features are generated. First, a TF-IDF vectorizer is fit on the training data and used to extract the top 200 features based on the average TF-IDF scores. The same vectorizer is then used to transform both the training and test sets.

Overview of Data Filtering

To provide a comprehensive overview of the data filtering process, table 5.2 is introduced, that summarizes the filtering stages that take place in the whole pipeline, including when and why certain data points were excluded. Despite these filtering steps, the final dataset used for evaluation still consists of a large number of posts, with 100k posts included for the final analysis. Additionally, datasets of 500k and 200k posts are used for evaluating the models, but they contain more edge cases and are less ideal for evaluation, as discussed earlier. The filtering process involves excluding posts based on various criteria, as outlined below:

Table 5.2: Overview of Data Filtering Process

Filter Criterion	**Number of Posts**
Initial Number of Posts	1,011,773
Posts Removed Due to Missing Body and Headline	7
Posts Removed Due to Previous Ban by Moderators	62,313
Posts Removed Due to Missing Votes	1,554
Posts Removed Due to Low Engagement in Comment Section	435,419
Number of Posts Available for Top/Flop 50%	512,480
Number of Posts Available for Top/Flop 20%	204,992
Number of Posts Available for Top/Flop 10%	102,496

5.4 Models for Prediction

The selection of models is based on the idea of providing different levels of explainability, ranging from fully explainable models like decision trees trained on interpretable features to more complex deep learning models, such as BERT. For most models, standard configurations are applied, as the primary focus of this work is not on improving accuracy, but rather on ensuring a solid comparison across models with varying levels of explainability.

Baseline 1

The most straightforward baseline approach relies solely on the mean value of the target variable from the training dataset to make predictions for the regression task. For the classification task, the baseline model assigns the label 'engaging post' to all instances.

Baseline 2: Logistic Regression + Text Length

As a second baseline, the model from Risch und Krestel (2020a) — a logistic regression model using the text length — is adopted. In their experiment with data from *The Guardian*, this model achieved an accuracy of 61% when predicting whether a post is a top or flop post using a balanced dataset (equal numbers of engaging and non-engaging posts).

Explainable Model 1: k Nearest Neighbors (KNN)

The first fully explainable model is a KNN, as introduced by Fix (1985), which is trained using the selected explainable features. The name stems from the approach it employs: predictions are determined based on the k closest neighbors within the feature space—essentially, the k most comparable data points (Fix, 1985). In classification tasks, it assigns the class that appears most frequently among these neighbors, while in regression, it simply computes the mean (Fix, 1985). KNN is a straightforward and transparent model that requires no explicit training phase, making it a great option for achieving full explainability.

Explainable Model 2: Decision Tree

The second model is a decision tree, also trained on the same explainable features. Its name reflects its hierarchical structure, where predictions are derived by navigating from the root node to a leaf node according to specific feature-based decision points/thresholds (Charbuty & Abdulazeez, 2021). Decision trees come with the advantage of segmenting data into well-defined, rule-based groups, providing an easily interpretable framework that explains the factors influencing predictions, making them another excellent choice for explainable modeling (Charbuty & Abdulazeez, 2021).

Interpretable Model 1: RandomForest + Surface-Level Features

Interpretable Model 1 utilizes a small set of lightweight features, such as word count and punctuation count, in combination with a shallow model, specifically a Random Forest (Breiman, 2001), as illustrated in Figure 5.12. Random Forests were selected as the interpretable model due to their superior performance over SVM models in preliminary tests on the dataset. Additionally, Random Forests provide valuable insights into feature importance, and their internal workings are relatively easy to understand. A Random Forest model operates by constructing multiple decision trees (hence the name), each trained on a different subset of the data and features (Breiman, 2001). Predictions are then made by aggregating the outputs of these individual trees, either through majority voting (for classification tasks) or averaging (for regression tasks) (Breiman, 2001).

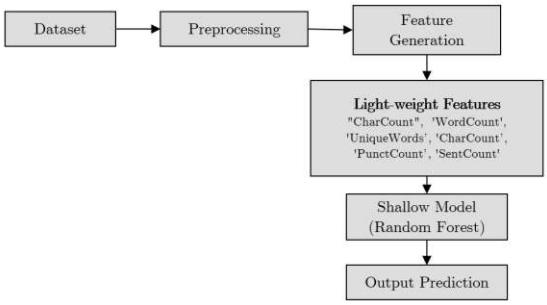

Figure 5.12: Interpretable Model 1: Lightweight Features + Random Forest

Interpretable Model 2: RandomForest + Complex Features

Interpretable Model 2 utilizes the full set of features introduced in Section 4.2, excluding sentiment information (discussed in the next section), and feeds them into a shallow classifier (Random Forest), as illustrated in Figure 5.13.

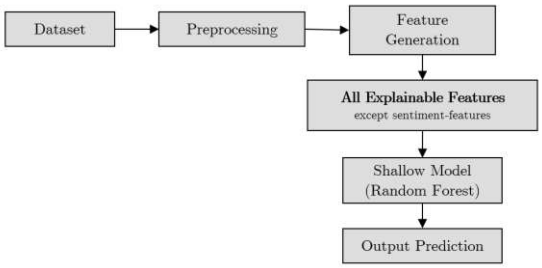

Figure 5.13: Interpretable Model 2: Complex Features + Random Forest

Interpretable Model 3: RandomForest + Complex and Sentiment Features

Interpretable Model 3 builds on Model 2 by adding a sentiment score generated using a pre-trained BERT model, as introduced by Guhr, Schumann, Bahrmann und Böhme (2020). This new sentiment feature is combined with the other features and fed into a shallow model (Random Forest), as depicted in Figure 5.14.

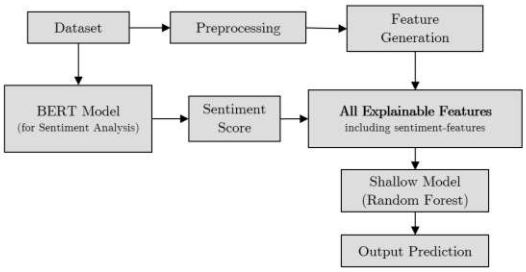

Figure 5.14: Interpretable Model 3: Semi-Explainable Features + Random Forest

Deep Learning Model 1: LSTM

The first deep learning model employs a LSTM network. LSTMs, originally introduced by Hochreiter (1997), are a type of Recurrent Neural Network (RNN) designed to capture 'long-term' (i.e., distant parts of a sequence) patterns in data by overcoming the issue of fading signals that many traditional models have to deal with (Hochreiter, 1997). While not the newest approach, LSTMs remain among the most widely used deep learning architectures, particularly for text-based tasks, due to their ability to effectively capture temporal patterns and contextual information. This enduring popularity and their robust performance justify their use in this work.

In this model, textual inputs (the raw body and headline of each post) and temporal inputs (the time elapsed since the creation of the article under which the post was made and the post's rank in the sequence of responses) are provided to the LSTM. The model architecture is depicted in Figure 5.15.

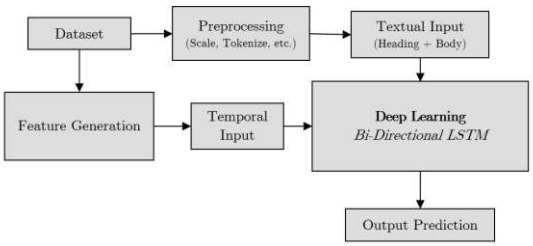

Figure 5.15: Deep Learning Model 1: LSTM

Deep Learning Model 2: Bidirectional Long Term Short Memory (BiLSTM)

The second deep learning model builds upon the previous architecture but utilizes a BiLSTM, as illustrated in Figure 5.16. The bidirectional design allows the LSTM to process input sequences in both forward and backward directions, capturing richer contextual information (Zhang, Zheng, Hu & Yang, 2015).

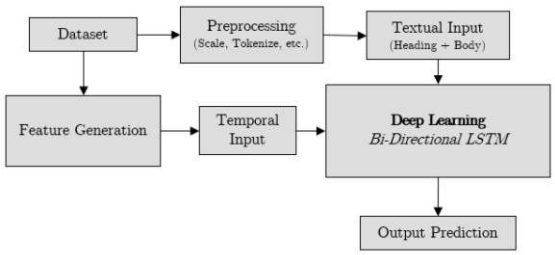

Figure 5.16: Deep Learning Model 2: Bidirectional LSTM

Deep Learning Model 3: GRU

Deep learning model 3 employs a GRU (Cho et al., 2014), with the same architecture as described by Risch und Krestel (2020a). A GRU is a type of RNN similar to LSTMs, but with a simpler architecture (Cho et al., 2014). They utilize an update gate and a reset gate, which offer computational efficiency and give them performance advantages for data-intensive applications (Cho et al., 2014). Since the GRU outperformed all other models with an accurracy of roughly 71% in predicting top and flop comments for the *The Guardian* dataset in the experiments conducted by Risch und Krestel (2020a), it was deemed appropriate to include this architecture in the present comparison.

However, the embeddings from Risch und Krestel (2020a) could not be directly reused, as they were designed for English-language applications. To adapt the model for this study, German fastText embeddings were incorporated using pre-trained vectors from the fastText library[2] which were constructed using Common Crawl and Wikipedia data.

Deep Learning Model 4: BERT Standalone

Deep Learning Model 4 uses DistillBERT [3], a distilled version of the BERT model tailored for the German language. Due to computational limitations of this research, the distilled version of the base German cased model is used, as it provides a more efficient alternative while retaining much of the performance of the full model.

[2] `https://dl.fbaipublicfiles.com/fasttext/vectors-crawl/cc.de.300.bin.gz`
[3] https://huggingface.co/distilbert/distilbert-base-german-cased

CHAPTER 6

Results

This chapter presents the results of the experiments and is organized into two main sections:

1. **Evaluation of Feature Importance**
   a) Evaluation of the Rule-Based Features
   b) Evaluation of Named Entity Features
   c) Evaluation of TF-IDF Term Features
   d) Evaluation of N-gram Features
2. **Model Performance on Popularity Prediction**
   a) Classification Task Performance:
   Differentiating Between 'Engaging' and 'Regular' Posts
   b) Regression Task Performance:
   Predicting the Absolute Votes a Post Will Get

The first section focuses on the evaluation of feature importance, where it analyzes whether the generated features significantly deviate between the two classes. Additionally, it examines the magnitude and direction (positive or negative) of these effects. Since some features, such as named entities, top TF-IDF vectors, and n-grams, have a large number of individual components, these are discussed separately from the other features.

The second section evaluates the performance of the model in predicting post popularity. It is subdivided into two tasks: classification and regression. The classification task assesses how well the model can differentiate between engaging and regular posts, while the regression task evaluates the model's ability to predict the absolute number of upvotes a post will receive.

6.1 Evaluation of Feature Importance

The evaluation of feature importance provides a detailed understanding of how each set of features contributes to the predictive performance of the model. The following subsections present the importance of different feature categories.

6.1.1 Evaluation of Rule-Based Features

Table 6.1, as well as the three next ones, presents the results obtained by applying Algorithm 5.2. The table shows where the differences between the two classes are significant for features by marking significant features with a 'Y' in the last column and non-significant with a 'N'. Based on the results $H_0 1$. can be rejected. The features are ranked based on the magnitude of their correlation strength, regardless of direction (positive or negative), and absolute Cohen's d value. However, the magnitude of these differences and the frequency with which these features have values above zero vary considerably. For instance, some features, like the average word length, are common across most rows, while others, such as the NOCApsShort frequency, are much rarer. Nevertheless, when these rare features do occur, they serve as strong indicators of whether a post is engaging or regular.

Table 6.1: Comparison of Rule Based Features Between Classes

Feature	Mean Regular	Mean Engaging	Correlation	PValue Corr.	PValue MannW.	|CohenD|	S
SyllableCount	43.684	73.822	0.114	<0.001	<0.001	0.667	Y
CharCount	167.886	281.602	0.113	<0.001	<0.001	0.662	Y
UniqueWords	22.568	36.395	0.111	<0.001	<0.001	0.653	Y
WordCount	25.004	41.446	0.111	<0.001	<0.001	0.653	Y
PolysyllableCount	4.683	8.364	0.112	<0.001	<0.001	0.635	Y
ShortWordsFreq	15.967	25.542	0.102	<0.001	<0.001	0.593	Y
StopWordFreq	11.000	18.072	0.100	<0.001	<0.001	0.593	Y
ShortWordComboFreq	5.534	8.785	0.095	<0.001	<0.001	0.552	Y
CapLetterFreq	6.999	12.216	0.100	<0.001	<0.001	0.548	Y
AvgParseTreeHeight	1.790	2.169	0.084	<0.001	<0.001	0.516	Y
BodyLength	155.704	259.193	0.085	<0.001	<0.001	0.500	Y
PunctCount	6.291	9.314	0.083	<0.001	<0.001	0.454	Y
SMOGScore	9.448	11.065	0.073	<0.001	<0.001	0.446	Y
SimilarityArticleBody	0.020	0.041	0.090	<0.001	<0.001	0.438	Y
DiversityRatio	0.952	0.923	-0.076	<0.001	<0.001	0.413	Y
SentCount	3.142	4.360	0.075	<0.001	<0.001	0.405	Y
AvgSentLength	9.285	11.624	0.065	<0.001	<0.001	0.400	Y
PeriodFreq	2.487	3.569	0.068	<0.001	<0.001	0.373	Y
SimilarityArticleTitle	0.015	0.034	0.081	<0.001	<0.001	0.372	Y
TitleLength	11.957	22.002	0.075	<0.001	<0.001	0.352	Y
NamedEntityFreq	0.214	0.441	0.066	<0.001	<0.001	0.332	Y
RepeatedWordsFreq	0.140	0.289	0.063	<0.001	<0.001	0.330	Y
LongWordsFreq	0.215	0.415	0.060	<0.001	<0.001	0.304	Y
FullCapsFreq	0.266	0.502	0.053	<0.001	<0.001	0.269	Y
ModelVerbsFreq	0.341	0.564	0.053	<0.001	<0.001	0.284	Y
NoCapsShortFreq	0.067	0.015	-0.041	<0.001	<0.001	0.267	Y
ARIScore	18.219	16.959	0.034	<0.001	<0.001	0.183	Y

Continued on next page

Table 6.1: Comparison of Rule Based Features Between Classes

Feature	Mean Regular	Mean Engaging	Corr.	PValue Corr.	PValue MannW.	|CohenD|	S
PunctToTextRatio	0.052	0.038	-0.049	<0.001	<0.001	0.307	Y
QuoteFreq	0.411	0.694	0.044	<0.001	<0.001	0.225	Y
FirstPluralFreq	0.119	0.244	0.046	<0.001	<0.001	0.218	Y
DigitsFreq	0.431	0.657	0.039	<0.001	<0.001	0.190	Y
ModalObligationFreq	0.116	0.208	0.038	<0.001	<0.001	0.207	Y
FirstSingFreq	0.432	0.638	0.040	<0.001	<0.001	0.182	Y
ModalPossibilityFreq	0.162	0.256	0.034	<0.001	<0.001	0.190	Y
ExclaimFreq	0.204	0.332	0.039	<0.001	<0.001	0.172	Y
ConjunctiveFreq	0.208	0.310	0.029	<0.001	<0.001	0.169	Y
LongWordComboFreq	0.031	0.062	0.030	<0.001	<0.001	0.140	Y
ExaggerateFreq	0.090	0.149	0.029	<0.001	<0.001	0.129	Y
SecondPluralFreq	0.028	0.057	0.025	<0.001	<0.001	0.120	Y
WordsPer100Chars	15.616	14.991	-0.017	<0.001	<0.001	0.079	Y
PastTenseFreq	0.155	0.222	0.023	<0.001	<0.001	0.119	Y
ShortPhraseFreq	0.024	0.045	0.024	<0.001	<0.001	0.106	Y
TitleToBodyRatio	0.144	0.152	0.015	<0.001	<0.001	0.049	Y
AffectionFreq	0.006	0.017	0.025	<0.001	<0.001	0.099	Y
OnlyWordsFreq	0.009	<0.001	-0.017	<0.001	<0.001	0.116	Y
ConsicePhraseFreq	0.019	0.034	0.022	<0.001	<0.001	0.091	Y
FormalityFreq	0.135	0.108	-0.023	<0.001	<0.001	0.070	Y
OnlyLinkFreq	0.005	<0.001	-0.014	<0.001	<0.001	0.094	Y
SentimentScore	0.350	0.330	-0.015	<0.001	<0.001	0.070	Y
EmojiSarcasticFreq	0.037	0.023	-0.016	<0.001	<0.001	0.081	Y
AvgWordLength	6.521	5.924	0.009	<0.001	<0.001	0.027	Y
StopWordsRatio	0.364	0.381	0.017	<0.001	<0.001	0.156	Y
EmojiNoseFreq	0.034	0.022	-0.013	<0.001	<0.001	0.069	Y
StronglyNegative	0.360	0.388	0.016	<0.001	<0.001	0.057	Y
UncompassionFreq	0.003	<0.001	-0.010	<0.001	<0.001	0.073	Y
DesireFreq	0.003	0.007	0.012	<0.001	<0.001	0.057	Y
WonderingFreq	0.001	0.003	0.011	<0.001	<0.001	0.046	Y
QuestionFreq	0.335	0.405	0.008	<0.001	<0.001	0.069	Y
StronglyPositive	0.070	0.054	-0.005	<0.001	<0.001	0.066	Y
OnlySymbolsFreq	0.002	<0.001	-0.006	<0.001	<0.001	0.047	Y
NoWordsFreq	0.002	<0.001	-0.006	<0.001	<0.001	0.047	Y
AnglicismsFreq	0.008	0.004	-0.006	<0.001	<0.001	0.046	Y
ProfanityFreq	0.004	0.007	0.006	<0.001	<0.001	0.033	Y
EmojiNegativeFreq	0.004	0.005	0.006	<0.001	<0.001	0.022	Y
FleschEaseScore	61.522	63.024	0.002	0.302	<0.001	0.024	N
CapToLowerRatio	0.066	0.063	<0.001	0.984	<0.001	0.032	N
RandomFeature2	0.498	0.497	-0.003	0.077	0.353	0.002	N
EmojiPositiveFreq	0.042	0.041	0.001	0.480	0.145	0.007	N
RandomFeature3	0.504	0.500	-0.001	0.681	0.118	0.008	N
EmojiSurpriseFreq	<0.001	<0.001	-<0.001	0.900	0.171	0.007	N
RandomFeature1	149.941	150.203	<0.001	0.853	0.332	0.001	N

6.1.2 Evaluation of Named Entity Features

Table 6.2 presents the results for NER features, sorted again by the magnitude of their absolute correlation strengths and absolute Cohen's *d* values, and again based on Algorithm 5.2. To keep the presentation concise and avoid overwhelming the reader, only the most and least significant features are included, leaving out those in the middle. The analysis of NER reveals that engaging posts generally contain a higher number of named entities compared to regular posts. Additionally, the political nature of the content is evident, with prominent Austrian political parties, such as the ÖVP, FPÖ, and SPÖ, frequently appearing among the top-named entities. In general, the presence of named entities in a post serves as a strong positive indicator of its relevance, if it is present in the text. However, the infrequent occurrence of many named entities dramatically reduces their total correlation strength and Cohen's d score.

Table 6.2: Comparison of Named Entity Features Between Classes

Feature	Mean Regular	Mean Engaging	Corr.	PValue Corr.	PValue MannW.	|CohenD|	S
Österreich	0.036	0.074	0.034	<0.001	<0.001	0.159	Y
Europa	0.010	0.027	0.027	<0.001	<0.001	0.125	Y
FPÖ	0.012	0.027	0.022	<0.001	<0.001	0.102	Y
SPÖ	0.007	0.016	0.020	<0.001	<0.001	0.093	Y
EU	0.014	0.027	0.018	<0.001	<0.001	0.091	Y
ÖVP	0.007	0.016	0.018	<0.001	<0.001	0.085	Y
Merkel	0.003	0.010	0.019	<0.001	<0.001	0.083	Y
deutsche	0.018	0.032	0.017	<0.001	<0.001	0.085	Y
Türkei	0.006	0.013	0.019	<0.001	<0.001	0.077	Y
USA	0.009	0.018	0.014	<0.001	<0.001	0.080	Y
Griechen	0.009	0.017	0.014	<0.001	<0.001	0.068	Y
Wien	0.011	0.019	0.013	<0.001	<0.001	0.065	Y
Faymann	0.002	0.005	0.013	<0.001	<0.001	0.058	Y
IS	0.003	0.008	0.011	<0.001	<0.001	0.063	Y
Erdogan	0.001	0.004	0.012	<0.001	<0.001	0.053	Y
...	...	...	...	...	...	...	...
Iran	0.001	0.002	0.003	0.095	0.003	0.019	N
Assad	0.001	0.002	0.002	0.266	0.002	0.021	N
Salzburg	0.002	0.002	0.002	0.165	0.098	0.009	N
Schweiz	0.001	0.002	0.001	0.451	0.020	0.014	N
China	0.002	0.003	0.001	0.474	0.027	0.013	N

6.1.3 Evaluation of TF-IDF Features

A similar situation exists for the TF-IDF vectors. Table 6.3 shows the results of the experiments. Once again, the differences between the two classes are statistically significant for most features. Again, we can see that political words such as 'FPÖ' seem to be positively correlated with engagement, as well. Another positive indicator seems to be talking about your country, continent or the community ('Austria', 'Country', 'Europe', 'All') and mentioning of time-related words ('Finally', 'Day', 'Year', 'Always'). However, the effect size is again relatively weak, reflecting mainly the fact that the individual terms only appear relatively rarely in posts and posts in general seem to exhibit a large diversity.

Table 6.3: Comparison of TF-IDF Vectors Between Classes

Feature	Mean Regular	Mean Engaging	Corr.	PValue Corr.	PValue MannW.	|CohenD|	S
endlich	0.003	0.010	0.027	<0.001	<0.001	0.115	Y
österreich	0.011	0.020	0.023	<0.001	<0.001	0.111	Y
jahr	0.014	0.024	0.023	<0.001	<0.001	0.113	Y
herr	0.005	0.012	0.025	<0.001	<0.001	0.105	Y
land	0.010	0.018	0.021	<0.001	<0.001	0.110	Y
europa	0.005	0.011	0.023	<0.001	<0.001	0.105	Y
mensch	0.011	0.019	0.019	<0.001	<0.001	0.106	Y
frau	0.008	0.016	0.023	<0.001	<0.001	0.100	Y
alle	0.016	0.024	0.018	<0.001	<0.001	0.096	Y
flüchtling	0.007	0.013	0.019	<0.001	<0.001	0.100	Y
fpö	0.007	0.014	0.020	<0.001	<0.001	0.098	Y
gegen	0.011	0.019	0.019	<0.001	<0.001	0.097	Y
mann	0.005	0.011	0.020	<0.001	<0.001	0.089	Y
immer	0.020	0.029	0.017	<0.001	<0.001	0.093	Y
tag	0.004	0.010	0.019	<0.001	<0.001	0.089	Y
...	...	...	...	...	...	...	...
genauso	0.004	0.004	-0.002	0.319	0.069	<0.001	N
falsch	0.004	0.004	-0.003	0.063	0.474	0.012	N
wahrscheinlich	0.005	0.004	-0.001	0.368	0.270	0.015	N
danke	0.013	0.009	-0.001	0.358	0.330	0.037	N
vergessen	0.005	0.004	<0.001	0.901	0.057	0.006	N

6.1.4 Evaluation of N-gram Features

As for the n-grams, the situation further deteriorates, as all the combinations are extremely rare for these features, as shown in Table 6.4. Additionally, it becomes evident that bi-grams appear way more often than tri-gram or n-gram combinations, which would be even rarer. Nevertheless, many of them are still significant, attributed to the fact that if they appear the post is likely to be engaging. The phrase 'verstehen nicht' (do not understand) for example appears three times more often in engaging posts than in regular posts and most of the time highlights the critical nature of a comment, making it more interesting for other users.

Table 6.4: Comparison of N-Gram Features Between Classes

Feature	Mean Regular	Mean Engaging	Corr.	PValue Corr.	PValue MannW.	|CohenD|	S
einfach nicht	0.002	0.006	0.013	<0.001	<0.001	0.063	Y
seit jahr	0.003	0.006	0.010	<0.001	<0.001	0.050	Y
saudi arabien	<0.001	0.003	0.010	<0.001	<0.001	0.049	Y
van bellen	0.001	0.003	0.011	<0.001	<0.001	0.045	Y
letzten jahr	0.002	0.005	0.011	<0.001	<0.001	0.045	Y
fpö wähler	<0.001	0.002	0.009	<0.001	<0.001	0.046	Y
jeden tag	<0.001	0.003	0.009	<0.001	<0.001	0.041	Y
steuer zahlen	<0.001	0.002	0.009	<0.001	<0.001	0.044	Y
österreich nicht	0.001	0.003	0.010	<0.001	<0.001	0.039	Y
frau merkel	<0.001	0.002	0.010	<0.001	<0.001	0.039	Y
überhaupt nicht	0.001	0.003	0.008	<0.001	<0.001	0.041	Y
kein wunder	<0.001	0.002	0.009	<0.001	<0.001	0.036	Y
spö övp	<0.001	0.002	0.009	<0.001	<0.001	0.035	Y
schwarz blau	<0.001	0.002	0.008	<0.001	<0.001	0.036	Y
verstehen nicht	0.001	0.003	0.007	<0.001	<0.001	0.038	Y
...	...	...	...	...	...	...	...
nicht besser	0.001	0.002	<0.001	0.930	0.338	0.001	N
leider nicht	0.003	0.003	-<0.001	0.914	0.453	0.001	N
jeden fall	0.002	0.002	<0.001	0.967	0.369	0.003	N
nicht lesen	0.001	0.001	-<0.001	0.927	0.500	<0.001	N

6.2 Model Performance on Popularity Prediction

This section presents the results of the models applied to predict post popularity.

6.2.1 Classification Task Performance

Table 6.5 presents the results for the classification task on the test dataset. The BiLSTM and BERT models achieve the highest overall performance. However, it is noteworthy that the Random Forest models trained with complex features (I2+I3) achieve competitive results, coming close to the performance of the two deep learning models. Interestingly, adding sentiment information to the feature set (I3) results in no significant impact on the model's ability to predict engaging posts. The Random Forest models show clear improvement when trained on more complex features. While these models are less interpretable than Decision Trees, they effectively balance performance and explainability.

Simpler models, such as Decision Trees and KNN, while inherently interpretable, perform poorly overall. Both models tend to overfit the training data, as evidenced by their significantly better training performance compared to their poor generalization capabilities on the test set. The Decision Tree and KNN however still perform better than the first baseline, which simply predicts the majority class and achieves minimal performance metrics. This indicates that it at least captures some useful patterns, but they both do not perform better than the second baseline, a logistic regression trained on text length alone. This suggests that the patterns identified by the Decision Tree are not more useful than the text length alone.

The BiLSTM clearly outperforms its unidirectional counterpart, demonstrating the benefits of incorporating bidirectional context in the classification task.

Table 6.5: Classification Task Results on the Test Dataset

Model	Regular Posts			Engaging Posts			Acc.
	P	R	F1	P	R	F1	
Baseline 1	0.00	0.00	0.00	50.00	100.00	66.67	50.00
Baseline 2 LogRegression	59.25	75.68	66.46	66.35	47.95	55.67	61.81
X1 DecisionTree	57.49	58.50	57.99	57.76	56.75	57.25	58.74
X2 KNN	60.31	59.74	60.02	60.12	60.69	60.40	60.21
I1 RandomForest Shallow	61.74	58.53	60.09	60.58	63.74	62.12	61.13
I2 RandomForest Complex	66.49	66.17	66.33	66.33	66.65	66.49	66.41
I3 RandomForest CF+Sent.	65.07	68.60	66.79	66.80	63.18	64.94	65.89
D1 LSTM	63.85	68.88	66.27	66.22	61.00	63.50	64.94
D2 BiLSTM	65.79	75.39	70.26	71.18	60.80	65.58	68.09
D3 BiGRU	67.98	64.23	66.05	66.10	69.74	67.87	66.99
D4 BERT	67.08	69.26	68.15	68.23	66.01	67.10	67.63

To address whether these performance differences are statistically significant, the experiment was repeated 10 times with different train-test splits. Afterward, the accuracy scores of each pair of models were compared using paired t-tests.

This test revealed that the best-performing interpretable model, I2, significantly outperforms Baseline 2 (P-value < 0.05) and consistently outperforms the baseline across all splits. Furthermore, the best deep learning model, D2 (BiLSTM), significantly outperforms the best interpretable model, I2 (P-value < 0.05). As a result hypothesis $H_0 2$. can be rejected.

Although the deep learning models were certainly not fully optimized due to computational constraints and the study's focus on explainability, their superior performance underscores their potential. These findings suggest that deep learning models generally hold an advantage over interpretable models in this domain, when it comes to prediction accuracy. With further improved hyperparameter optimization and optimized embeddings these models might as well be able to surpass the threshold of 70% accuracy, as in the paper by Risch und Krestel (2020a), where they introduced a custom embedding for the GRU. Using the generic German fast-text embedding most likely led to less optimal performance of this model in this study. The fact that the logistic regression model baseline from Risch und Krestel (2020a), achieved almost the same accuracy in the experiments, is a good indication for a solid comparability of the two problems.

Figure 6.6 shows the classification accuracy for the different labeling approaches. Contrary to the paper by Risch und Krestel (2020a) the labeling approach in this experiment has a strong effect on the accuracy since the time groups are substantially smaller than in their experiments.

Table 6.6: Classification Results for Different Top/Flop Splits

Model	Accuracy (%)		
	Top/Flop 50%	Top/Flop 20%	Top/Flop 10%
Baseline 1	50.00	50.00	50.00
Baseline 2 LogRegression	55.84	60.20	61.81
X1 DecisionTree	52.38	56.51	58.74
X2 KNN	53.04	58.24	60.21
I1 RandomForest Shallow	54.71	59.07	61.13
I2 RandomForest Complex	58.12	63.83	66.41
I3 RandomForest CF+Sent.	58.16	63.88	65.89
D1 LSTM	52.37	63.23	64.94
D2 BiLSTM	58.21	65.15	68.09
D3 BiGRU	57.16	63.04	66.99
D4 BERT	57.68	63.69	67.63

Random Forest: Feature Importance

Figure 6.1 presents the 15 most important features in the Random Forest model, where the values represent each feature's contribution to minimizing the Gini impurity. Notably, the most influential feature is the cosine similarity between the post text and the article body, followed by the cosine similarity of the post title. Several subsequent features capture the length of the text, measured through word count, syllables, or characters, which all share a similar features importance score.

Additionally, the model appears to prioritize features that consistently have many non-zero values across many instances, rather than relying on features that are rare but highly decisive when they do occur.

It is also noteworthy that if the feature importance were extended to include an additional 15 features, these would have slightly lower importance but still hold meaningful values. This highlights the robust nature of the Random Forest model: removing a few features is unlikely to cause a dramatic drop in accuracy, as the model can rely on other features to compensate.

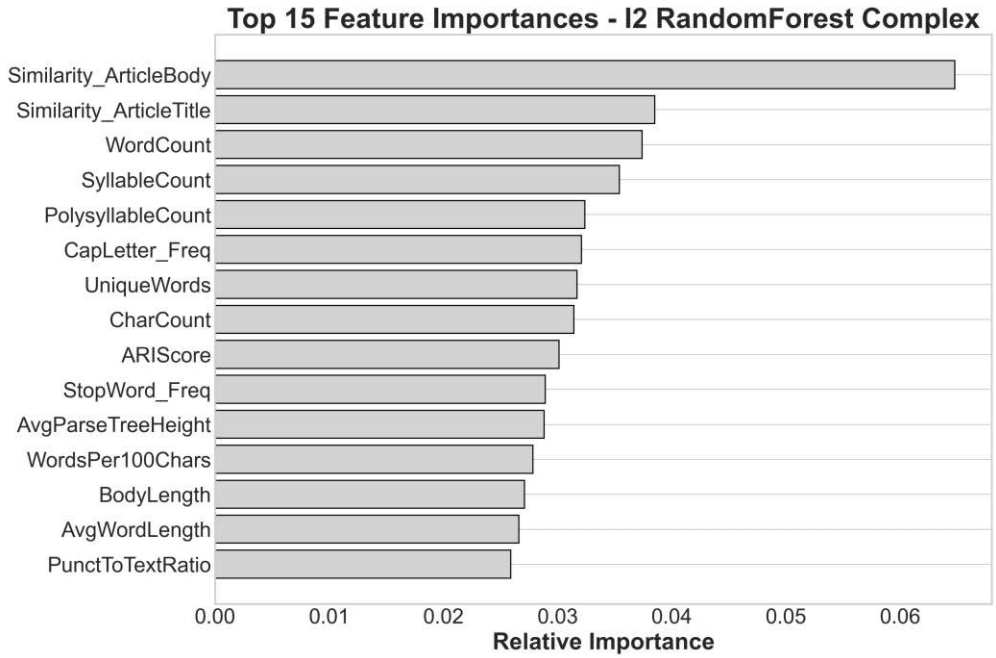

Figure 6.1: Feature Importance in Random Forest Model I2

Error Analysis

Figure 6.2 illustrates four posts classified by the Random Forest model I2. The top left and bottom right posts were correctly classified, while the top right and bottom left posts were misclassified.

The graphic highlights a clear trend: the model tends to favor longer, more complex posts over shorter, simpler ones. For example, the post in the bottom left is a strong example of an engaging post, sharing personal emotions, using expressive language, and aligning well with the topic. Conversely, the top left post is clearly a regular post—it merely confirms another opinion without contributing additional interesting information.

The misclassification in the top right is more challenging to assess. While the post exhibits some quality and poses an engaging question, it fails to offer new information or a unique perspective. Thus, the model's decision can be considered debatable, though not entirely incorrect.

The misclassification in the bottom left, however, is a clear mistake by the model. This post received 73 upvotes, making it one of the most upvoted in the dataset. Such a high level of engagement underscores a significant error in the model's prediction.

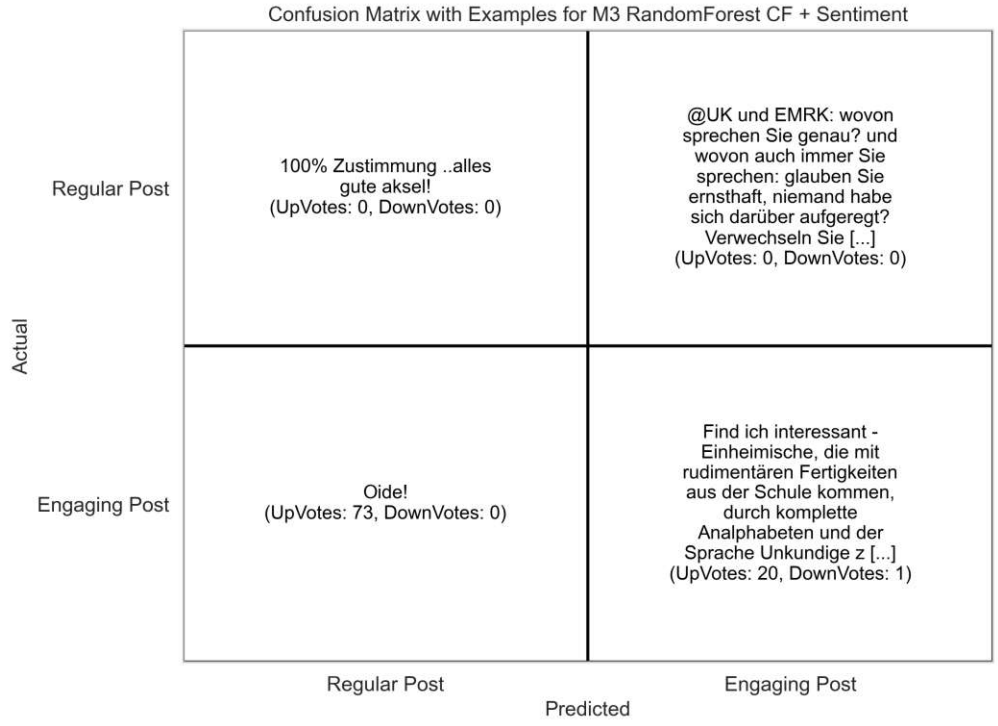

Figure 6.2: Different Types of Errors Produced by the Random Forest Model I2

Figure 6.3 displays the title of the article under which the misclassified post appeared, as discussed on the previous page. In this example, the user makes a joke by inventing the word 'oide', derived from the Austrian term 'oida', in response to the article's content. This case illustrates the challenges for any algorithm, as the post is not only extremely short but consists solely of a unique word not present elsewhere in the dataset. Moreover, understanding the base term 'oida' and its humorous intent requires cultural knowledge of Austrian culture and dialect, further complicating accurate classification.

Although it is understandable why an algorithmic solution might often overlook such examples—given their rarity—it would be a significant loss for the online community if such posts were not recognized as engaging by a sorting algorithm, potentially causing them to be overlooked and forgotten.

ArticleTitle	PostText	UpVotes	VoteMedian
Klagenfurter Obmann der Grünen wird eine "Parteiobfrau" Die Grünen bezeichnen sich im neuen Statut nur noch mit weiblicher Geschlechtsform	Oide!	73	3.0

Figure 6.3: Example of a Highly Difficult Classification Example

Figure 6.4 illustrates the total number of posts appearing in the comment sections of articles of a specific category, alongside the samples of those correctly classified. The majority of posts were predominantly written under the news categories 'Domestic' ('Inland'), 'Panorama', and 'Economy' ('Wirtschaft').

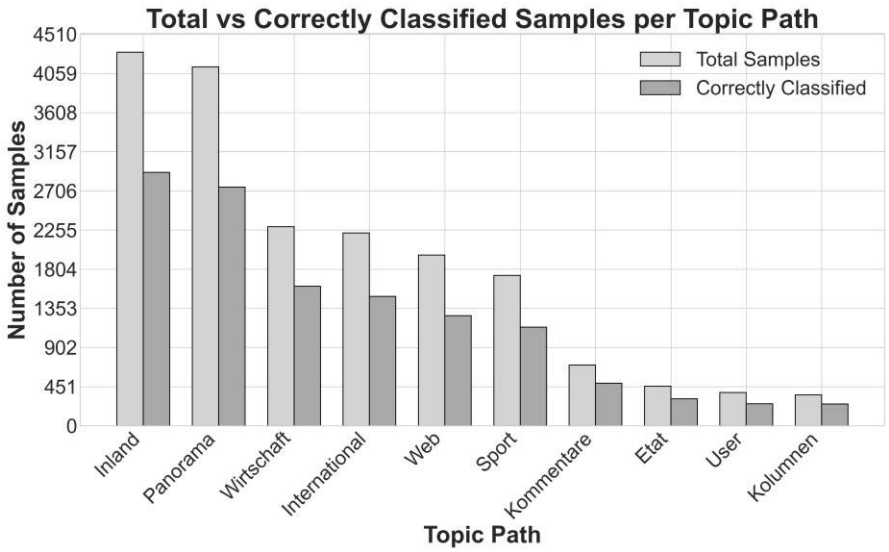

Figure 6.4: Total Samples and Correctly Classified Samples in Model I2

Figure 6.5 shows how the classification error varies across the comment sections of different news articles. The model, for instance, clearly performs worse with posts under the 'Sports' (3% worse than the average accuracy) and 'Web' (4% worse) categories. This likely reflects a bias toward considering engaging posts as longer and more complex. However, in sports-related discussions, posts are often short but can still be engaging, which might lead to misclassifications.

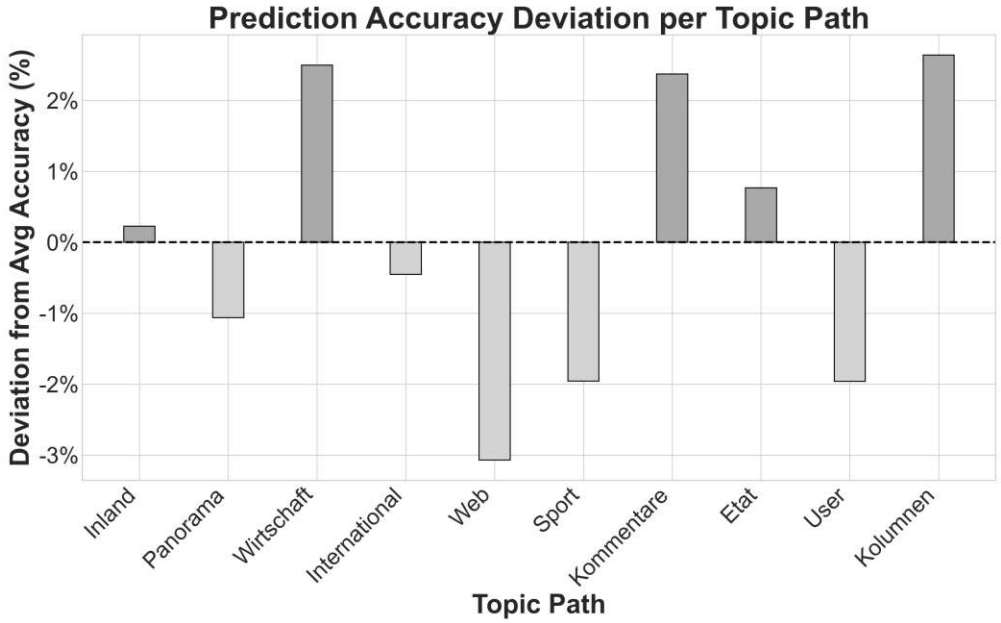

Figure 6.5: Classification Errors for Different Topic Paths in Model I2

6.2.2 Regression Task Performance

Table 6.7 shows the results from the regression tasks of predicting the total number of upvotes for each post. As already expected trying to predict the exact number of votes, leads to large errors, as Bandari et al. (2012) also observed, because of the reasons already discussed earlier. But for the sake of completeness, the results are included here.

Table 6.7: Regression Task Results on the Test Dataset

Model	MSE	MAE	R2
Baseline Regression	1.35618	0.80925	-0.000293192
I2 RandomForest Complex	1.29268	0.76988	0.0465441
I3 RandomForest CF+Sent.	1.29240	0.76979	0.0467443
D1 LSTM	1.34257	0.78501	0.00974544
D2 BiLSTM	1.32190	0.77848	0.0249916

CHAPTER 7

Discussion

The results suggest that explainable models, such as KNN and decision trees trained on interpretable features, offer inherent transparency but often lack the complexity required for competitive performance. In contrast, interpretable models like random forests, when trained on explainable features, strike a reasonable balance between human interoperability and predictive power, and can, when trained with the right features significantly outperform the baseline introduced by Risch und Krestel (2020a). However, their performance remains behind deep learning approaches.

Certain specialized features, such as 'text consisting only of special symbols and spaces', can help distinguish between classes. However, their effectiveness as strong predictors is limited by the rarity of positive occurrences (i.e., the pattern appearing in the text). In contrast, generic features like 'text length' have positive values across all data instances, revealing effective general trends but frequently leading to misclassifications due to counterexamples. While a custom pattern, when present, is often a stronger predictor for a single instance than a generic feature, its overall utility is diminished by its infrequent occurrence, making it less impactful across the dataset as a whole.

Overall, models like random forests tend to favor these features that are consistently available across all data points, such as text length or parse tree height score. While these features may not always be the most decisive, their consistent availability provides reliable input, making them advantageous in general, even if they do lead to many classification errors.

This suggests that generating highly specific features may not be the most effective strategy for creating features suitable for integration into interpretable or explainable models. Striking a balance between a feature's frequency of occurrence and its discriminative power when present is crucial. One possible approach is to merge features with few positive values, though this may come at the cost of reduced explainability.

Through the error analysis, it can be observed that certain model biases often arise. For example, models tend to favor longer and more complex texts, classifying them as 'engaging' more frequently. As a result, shorter texts, especially creative or brief comments or jokes, may be misclassified. However, this bias is based on the content itself, not on external factors like user history. This is in contrast to Park et al. (2016), where the model was influenced by the previous upvotes a user received, potentially introducing user model bias. On the other hand, in such cases, a user who consistently writes short, humorous posts that gain more upvotes might have their posts classified as engaging, not because of the content, but due to their engagement history. This work focuses on a content-based approach, which has the advantage of evaluating comments independently of the author's history. However, this also means that the model may inadvertently favor the standard content, potentially overlooking more creative or less conventional contributions.

Another noteworthy insight from the experiments was that the baseline from Risch und Krestel (2020a) performed almost the same as in their experiments (with an accuracy of 61%), serving as a solid indication for the validity of the approach taken by in this work.

Their GRU appraoch however failed to deliver the solid results of 71% of accuracy, as in their paper, and only achieved an accuracy of 67%. Most likely because their custom embeddings could not be directly transformed to German, and this study had to rely on generic fasttext-embedding. Additionally, differences in the layout and functionality of the websites might have influenced the different results. For instance, the average number of votes in the Guardian dataset was significantly higher than in this dataset.

A further interpretation could be that there is an inherent ceiling on how well a model can perform with this dataset, as the ground truth itself may not be entirely solid. This limitation could stem from time-related bias, that was corrected, but naturally could not be removed without any traces, or the broader question of whether the number of upvotes is truly a reliable indicator of what people want to see.

Another potential issue with training a model to sort articles based on previous upvotes is that it may reinforce the majority opinion, favoring a certain type of post. This can discourage other users from posting, as their contributions may not be ranked highly. Over time, this could lead to the formation of an echo chamber—a space where only users with similar opinions engage. This is problematic, as such communities can, for example, propagate hate speech (Bagavathi, Bashiri, Reid, Phillips & Krishnan, 2019), or increase the likelihood of fake news being shared (Sharma et al., 2019).

CHAPTER 8

Conclusion, Limitations & Future Work

8.1 Conclusion and Future Research

This work sheds light on the explainable prediction of user post popularity, presenting a comprehensive set of explainable features. Unlike most previous research, which often focuses on a limited subset of features or aims primarily at slightly improving accuracy with deep learning models, this study emphasizes understanding the 'how' behind the models' predictions. In many cases, especially when fine-tuning deep learning models, it is difficult to measure how well the model understands the use case, and in case of performance improvements to measure what additional knowledge it gained. In contrast, by introducing new explainable features, this work provides clear insights into what the models have actually 'learned'.

This thesis demonstrated that interpretable models, such as Random Forests trained with these explainable features, offer a solid compromise between predictive power and human understandability. Although such models may yield lower accuracy than more complex, deep learning models, they offer the critical advantage of providing transparent decision-making processes. This helps users understand the key factors that drive the model's predictions.

Predicting the popularity of user posts on newspaper websites remains a challenging task, as some influencing factors are difficult to capture in data—such as the general mood of the readership or specific political opinions. Additionally, the data is often incomplete. For instance, a comment containing only a link to a *YouTube* video may not capture the actual content of the video, which could significantly influence its popularity. Moreover, comments referencing past events or relying on cultural codes may not be

fully represented in the available data. In light of these challenges, future research could explore the integration of explainable models with knowledge graphs, which could provide richer context and improve predictions by accounting for external knowledge.

Another major insight from this work is the need for more data with a clearer ground truth. While large-scale datasets from online newspaper platforms provide valuable real-world data, they come with several limitations. One significant issue is the influence of ranking systems, such as the practice of highlighting posts with the most interactions. This can skew the data, as posts with higher visibility may naturally receive more interactions, regardless of their intrinsic quality. While corrective measures can reduce some of this bias, it can never be fully eliminated.

In addition, this work assumes that posts with the most votes are the most interesting, but this assumption may not always hold. For example, users may upvote, or downvote posts based on agreement or disagreement with the opinions expressed, rather than the post's intrinsic interest. Future research could focus on collecting new data where all comments receive equal attention and are ranked based on a broader scale of interest, rather than the simple binary of upvotes and downvotes. This could lead to more accurate predictions and allow for the development of new algorithms that leverage these nuanced labels.

A further limitation of this work can be found in the training of different models, particularly when exploring different model configurations, architectures and hyperparameter combinations. Due to resource and time constraints, this work only investigates the most promising combinations and does not exhaust all possibilities. As such, future research could further explore a wider range of model configurations and hyperparameters to improve performance.

An interesting field for future research would be to see if, the explainable factors that influence post popularity may evolve over time. For example, political views and societal trends can change, which may shift the characteristics of 'engaging posts' and 'regular posts.' This study focuses on a specific period—the late 2010s—so predictions made here may not generalize well to other periods. Future research could examine how the means and characteristics of engaging and regular posts shift over time, such as changes in text length or sentiment. This however would require a vast amount data, spanning over the course of many years.

Additionally, it would be interesting to explore whether the explainable features identified in this work are transferable to other domains, particularly in similar contexts, such as other German-speaking newspaper forums like *Die Zeit* or *Frankfurter Allgemeine*, but also in their applicability in the context German social media text. Further investigation could explore which features are language-specific and which can be generalized across multiple languages.

CHAPTER 9

Appendix

9.1 Stopwords List

['aber', 'ab', 'aha', 'aso', 'achso', 'ach', 'als','also','am','an','ander', 'andere', 'anderem', 'anderen', 'anderer', 'anderes', 'anderm', 'andern', 'auch','auf','aus', 'bei', 'bin', 'bis', 'bist', 'bzw', 'beim', 'bei', 'breits', 'da', 'damit', 'dann', 'der', 'den', 'des', 'dem', 'die', 'das', 'dass', 'daß', 'darüber', 'dazu', 'dafür', 'derselbe', 'derselben', 'denselben', 'desselben', 'demselben', 'denen', 'dieselbe', 'dieselben', 'dasselbe', 'denn', 'derer','dessen', 'dies', 'diese', 'diesem', 'diesen', 'dieser', 'dieses', 'doch','dort', 'durch', 'du', 'eben', 'ein', 'eigentlich', 'eine', 'einem', 'einen', 'einer', 'eines', 'einig', 'einigem', 'einmal', 'einigen', 'einiger', 'einiges', 'es', 'etwas', 'eher', 'eh', 'echt', 'erst', 'etc', 'er', 'für', 'fast', 'genau', 'gar', 'ganz', 'geht', 'gehört', 'gemacht', 'gerade', 'gehen', 'gesehen', 'gesehn', 'ganze', 'halt', 'hier','hierzu', 'hin', 'haben', 'hat', 'ihr', 'ihre', 'ihrem', 'ihn', 'ihren', 'ihrer','ihres', 'im', 'in', 'indem', 'ins', 'irgend', 'irgendwas', 'irgendwie', 'irgendwer', 'ist','is', 'ja', 'jede', 'jene', 'jenem', 'jenen', 'jener', 'jenes', 'jetzt', 'klar', 'kommt', 'lassen', 'lasst', 'lass', 'lieber', 'mit', 'mal', 'mehr', 'mir', 'mein', 'natürlich', 'na', 'nach', 'nun', 'noch', 'nur', 'man', 'ob', 'obwohl', 'oder', 'ohne', 'ohnehin', 'paar', 'schon', 'sehr', 'so', 'sozusagen', 'somit','solche', 'solchem', 'solchen', 'solcher', 'solches', 'sowieso', 'sondern', 'sind', 'sich', 'sieht', 'sonst', 'sehen', 'sicher', 'sowas', 'tatsächlich', 'über', 'um', 'und', 'überhautpt', 'unter' 'viel', 'vom', 'von','vor', 'während', 'was', 'wegen', 'weil', 'weiter', 'welche', 'welchem', 'welchen', 'wobei', 'wieso' 'welcher','welches', 'wenn', 'wie', 'wieder', 'wohl', 'wird', 'werden', 'wodurch', 'wo', 'weshalb', 'warum', 'wieso', 'weit','wer', 'zu', 'zum', 'zur', 'zwar', 'zwischen', 'ziemlich', 'artikel', 'inhalt', 'post', 'beitrag', 'seien', 'haben', 'habe', 'können', 'sollen', 'soll', 'müssen', 'werden', 'gehen', 'machen', 'helfen', 'bringen', 'wollen', 'brauchen', 'tun', 'sagen', 'bleiben', 'sein', 'hab', 'gibt', 'gibts', 'kommen', 'dürfen', 'gelten']

List of Figures

List of Tables

List of Algorithms

Acronyms

10kGNAD Ten Thousand German News Articles Dataset. 14, 15

ABSA Aspect Based Sentiment Analysis. 18

AI Artificial Intelligence. 2, 19

ARI Automated Readability Index. 27, 30, 41

BERT Bidirectional Encoder Representations from Transformers. 3, 14, 15, 61, 63, 64, 71, 89, 90

BiLSTM Bidirectional Long Term Short Memory. 64, 71, 72

BOW Bag of Words. 13, 23, 42

CNN Convolutional Neural Network. 16, 17

GottBERT German OSCAR text trained BERT. 14, 15

GRU Gated Recurrent Unit. 16, 17, 64, 72, 78

KNN k Nearest Neighbors. 61, 71, 77

LDA Latent Dirichlet Allocation. 28

LSTM Long Short Term Memory. 3, 13, 63, 64, 84

MAE Mean Absolute Error. 55

NER Named Entity Recognition. 15, 18, 23, 58, 68, 83

NLP Natural Language Processing. vii, ix, xv, 7, 18, 29

NLTK Natural Language Processing Tool Kit. 59

OFAI Austrian Research Institute for Artificial Intelligence. 5

OSCAR Super-large Crawled Aggregated Corpus. 14

POS Part of Speech. 26, 28, 58

regex regular expression. 32–35, 37–40, 57, 58

RMSE Root Mean Squared Error. 55

RNN Recurrent Neural Network. 63, 64

RoBERTa A Robustly Optimized BERT Pretraining Approach. 14

SMOG Simple Measurement of Gobbledygook. 30, 41

SMPD Social Media Popularity Prediction. 1

SVM Support Vector Machine. 13, 62

TF-IDF Term frequency–inverse document frequency. 23, 29, 60, 65, 69, 85

Bibliography

Ali, A. M., Ghaleb, F. A., Al-Rimy, B. A. S., Alsolami, F. J. & Khan, A. I. (2022). Deep ensemble fake news detection model using sequential deep learning technique. *Sensors*, *22* (18), 6970.

Ali, S. F. & Masood, N. (2024). Evaluation of adjective and adverb types for effective twitter sentiment classification. *Plos one*, *19* (5), e0302423.

Alkomah, F., Salati, S. & Ma, X. (2022). A new hate speech detection system based on textual and psychological features. *Int J Adv Comput Sci Appl.*, *13* (8), 860–869.

ALSaif, H. & Alotaibi, T. (2019). Arabic text classification using feature-reduction techniques for detecting violence on social media. *International Journal of Advanced Computer Science and Applications*, *10* (4).

Ambroselli, C., Risch, J., Krestel, R. & Loos, A. (2018). Prediction for the newsroom: Which articles will get the most comments? In *Proceedings of the 2018 conference of the north american chapter of the association for computational linguistics: Human language technologies, volume 3 (industry papers)* (S. 193–199).

Arora, A., Hassija, V., Bansal, S., Yadav, S., Chamola, V. & Hussain, A. (2023). A novel multimodal online news popularity prediction model based on ensemble learning. *Expert Systems*, *40* (8), e13336.

Arunthavachelvan, K., Raza, S. & Ding, C. (2024). A deep neural network approach for fake news detection using linguistic and psychological features. *User Modeling and User-Adapted Interaction*, *34* (4), 1043–1070.

Assenmacher, D., Niemann, M., Müller, K., Seiler, M., Riehle, D. M. & Trautmann, H. (2021). Rp-mod rp-crowd: Moderator-and crowd-annotated german news comment datasets. In *Thirty-fifth conference on neural information processing systems datasets and benchmarks track (round 2).*

Bagavathi, A., Bashiri, P., Reid, S., Phillips, M. & Krishnan, S. (2019). Examining untempered social media: analyzing cascades of polarized conversations. In *Proceedings of the 2019 ieee/acm international conference on advances in social networks analysis and mining* (S. 625–632).

Bandari, R., Asur, S. & Huberman, B. (2012). The pulse of news in social media: Forecasting popularity. In *Proceedings of the international aaai conference on web and social media* (Bd. 6, S. 26–33).

Benaicha, M., Thulke, D. & Turan, M. A. T. (2024). Leveraging cross-lingual transfer learning in spoken named entity recognition systems. In *Proceedings of the 20th*

conference on natural language processing (konvens 2024) (S. 98–105).
Boczkowski, P. J. & Mitchelstein, E. (2012). How users take advantage of different forms of interactivity on online news sites: Clicking, e-mailing, and commenting. *Human communication research*, *38* (1), 1–22.
Breiman, L. (2001). Random forests. *Machine learning*, *45*, 5–32.
Burnap, P. & Williams, M. L. (2015). Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making. *Policy & internet*, *7* (2), 223–242.
Chakraborty, A., Paranjape, B., Kakarla, S. & Ganguly, N. (2016). Stop clickbait: Detecting and preventing clickbaits in online news media. In *2016 ieee/acm international conference on advances in social networks analysis and mining (asonam)* (S. 9–16).
Charbuty, B. & Abdulazeez, A. (2021). Classification based on decision tree algorithm for machine learning. *Journal of Applied Science and Technology Trends*, *2* (01), 20–28.
Chatterjee, S. & Hadi, A. S. (2015). *Regression analysis by example*. John Wiley & Sons.
Chew, R., Kery, C., Baum, L., Bukowski, T., Kim, A., Navarro, M. et al. (2021). Predicting age groups of reddit users based on posting behavior and metadata: classification model development and validation. *JMIR Public Health and Surveillance*, *7* (3), e25807.
Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. & Bengio, Y. (2014, Oktober). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In A. Moschitti, B. Pang & W. Daelemans (Hrsg.), *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (S. 1724–1734). Doha, Qatar: Association for Computational Linguistics. Zugriff auf `https://aclanthology.org/D14-1179` doi: 10.3115/v1/D14-1179
Cohen, J. (2013). *Statistical power analysis for the behavioral sciences*. routledge.
Daga, I., Gupta, A., Vardhan, R. & Mukherjee, P. (2020). Prediction of likes and retweets using text information retrieval. *Procedia computer science*, *168*, 123–128.
De Araujo, P. H. L., Baumann, A., Gromann, D., Krenn, B., Roth, B. & Wiegand, M. (2024). Proceedings of the 20th conference on natural language processing (konvens 2024). In *Proceedings of the 20th conference on natural language processing (konvens 2024)*.
Ding, K., Wang, R. & Wang, S. (2019). Social media popularity prediction: A multiple feature fusion approach with deep neural networks. In *Proceedings of the 27th acm international conference on multimedia* (S. 2682–2686).
Dixit, S. & Soni, N. (2024). Enhancing stock market prediction using three-phase classifier and em-epo optimization with news feeds and historical data. *Multimedia Tools and Applications*, *83* (13), 37859–37887.
Eder, E., Krieg-Holz, U. & Wiegand, M. (2023). A question of style: A dataset for analyzing formality on different levels. In *Findings of the association for*

computational linguistics: Eacl 2023 (S. 580–593).
Elshawi, R., Al-Mallah, M. H. & Sakr, S. (2019). On the interpretability of machine learning-based model for predicting hypertension. *BMC medical informatics and decision making*, *19*, 1–32.
Fix, E. (1985). *Discriminatory analysis: nonparametric discrimination, consistency properties* (Bd. 1). USAF school of Aviation Medicine.
Flesch, R. (1948). A new readability yardstick. *Journal of applied psychology*, *32* (3), 221.
Gabrilovich, E. & Markovitch, S. (2005). Feature generation for text categorization using world knowledge. In *Ijcai* (Bd. 5, S. 1048–1053).
Garla, V. N. & Brandt, C. (2012). Ontology-guided feature engineering for clinical text classification. *Journal of biomedical informatics*, *45* (5), 992–998.
Geetha, R., Karthika, S., Sowmika, C. J. & Janani, B. M. (2021). Auto-off id: Automatic detection of offensive language in social media. In *Journal of physics: Conference series* (Bd. 1911, S. 012012).
Genç, Ş. & Surer, E. (2023). Clickbaittr: Dataset for clickbait detection from turkish news sites and social media with a comparative analysis via machine learning algorithms. *Journal of Information Science*, *49* (2), 480–499.
González-Ibánez, R., Muresan, S. & Wacholder, N. (2011). Identifying sarcasm in twitter: a closer look. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies* (S. 581–586).
Guhr, O., Schumann, A.-K., Bahrmann, F. & Böhme, H. J. (2020, May). Training a broad-coverage german sentiment classification model for dialog systems. In *Proceedings of the 12th language resources and evaluation conference* (S. 1620–1625). Marseille, France: European Language Resources Association. Zugriff auf `https://www.aclweb.org/anthology/2020.lrec-1.202`
Haneczok, J. & Piskorski, J. (2020). Shallow and deep learning for event relatedness classification. *Information Processing & Management*, *57* (6), 102371.
Häring, M., Loosen, W. & Maalej, W. (2018). Who is addressed in this comment? automatically classifying meta-comments in news comments. *Proceedings of the ACM on Human-Computer Interaction*, *2* (CSCW), 1–20.
Hellwig, N. C., Fehle, J., Bink, M. & Wolff, C. (2024). Gerestaurant: A german dataset of annotated restaurant reviews for aspect-based sentiment analysis. *arXiv preprint arXiv:2408.07955*.
Hochreiter, S. (1997). Long short-term memory. *Neural Computation MIT-Press*.
Jain, M. K., Gopalani, D. & Meena, Y. K. (2024). Confake: fake news identification using content based features. *Multimedia Tools and Applications*, *83* (3), 8729–8755.
Kamran, M., Alghamdi, A. S., Saeed, A. & Alsubaei, F. S. (2024). Mr-fnc: A fake news classification model to mitigate racism. *International Journal of Advanced Computer Science & Applications*, *15* (2).
Kavitha, M. & Akila, K. (2024). Amplifying document categorization with advanced features and deep learning. *Multimedia Tools and Applications*, 1–19.
Khanday, A. M. U. D., Wani, M. A., Rabani, S. T., Khan, Q. R. & Abd El-Latif, A. A.

(2024). Hapi: An efficient hybrid feature engineering-based approach for propaganda identification in social media. *Plos one*, *19* (7), e0302583.
Kim, T. K. (2015). T test as a parametric statistic. *Korean journal of anesthesiology*, *68* (6), 540–546.
Kitchenham, B. (2004). Procedures for performing systematic reviews. *Keele, UK, Keele University*, *33* (2004), 1–26.
Lai, X., Zhang, Y. & Zhang, W. (2020). Hyfea: winning solution to social media popularity prediction for multimedia grand challenge 2020. In *Proceedings of the 28th acm international conference on multimedia* (S. 4565–4569).
Li, C.-T., Chen, H.-Y. & Zhang, Y. (2021). On exploring feature representation learning of items to forecast their rise and fall in social media. *Journal of Intelligent Information Systems*, *56* (3), 409–433.
Lisch, R. & Kriz, J. (1978). Grundlagen und modelle der inhaltsanalyse. *Reinbek: Rowohlt*.
Liu, A.-A., Wang, X., Xu, N., Guo, J., Jin, G., Zhang, Q., ... Zhang, S. (2022). A review of feature fusion-based media popularity prediction methods. *Visual Informatics*, *6* (4), 78–89.
Ma, Y.-W., Chen, J.-L., Chen, L.-D. & Huang, Y.-M. (2022). Intelligent clickbait news detection system based on artificial intelligence and feature engineering. *IEEE Transactions on Engineering Management*.
Madylova, A. & Oguducu, S. G. (2009). A taxonomy based semantic similarity of documents using the cosine measure. In *2009 24th international symposium on computer and information sciences* (S. 129–134).
McKnight, P. E. & Najab, J. (2010). Mann-whitney u test. *The Corsini encyclopedia of psychology*, 1–1.
McLaughlin, G. (1969). *Smog grading–a new readability formula in the journal of reading.* May.
Mehravaran, S. & Shamsinejadbabaki, P. (2023). Devising a machine learning-based instagram fake news detection system using content and context features. *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, *47* (4), 1657–1666.
Montani, J. P. & Schüller, P. (2018). Tuwienkbs at germeval 2018: German abusive tweet detection.
Mossie, Z. & Wang, J.-H. (2020). Vulnerable community identification using hate speech detection on social media. *Information Processing & Management*, *57* (3), 102087.
Mujahid, M., Kına, E., Rustam, F., Villar, M. G., Alvarado, E. S., De La Torre Diez, I. & Ashraf, I. (2024). Data oversampling and imbalanced datasets: an investigation of performance for machine learning and feature engineering. *Journal of Big Data*, *11* (1), 87.
Nelson, M. N., Ksiazek, T. B. & Springer, N. (2021). Killing the comments: Why do news organizations remove user commentary functions? *Journalism and Media*, *2* (4), 572–583.
Nesi, P., Pantaleo, G., Paoli, I. & Zaza, I. (2018). Assessing the retweet proneness of

tweets: predictive models for retweeting. *Multimedia tools and applications*, *77* (20), 26371–26396.

Pachinger, P., Goldzycher, J., Planitzer, A. M., Kusa, W., Hanbury, A. & Neidhardt, J. (2024). Austrotox: A dataset for target-based austrian german offensive language detection. *arXiv preprint arXiv:2406.08080*.

Park, D., Sachar, S., Diakopoulos, N. & Elmqvist, N. (2016). Supporting comment moderators in identifying high quality online news comments. In *Proceedings of the 2016 chi conference on human factors in computing systems* (S. 1114–1125).

Pérez-Landa, G. I., Loyola-González, O. & Medina-Pérez, M. A. (2021). An explainable artificial intelligence model for detecting xenophobic tweets. *Applied Sciences*, *11* (22), 10801.

Petersen-Frey, F. & Biemann, C. (2024). Fine-grained quotation detection and attribution in german news articles. In *Proceedings of the 20th conference on natural language processing (konvens 2024)* (S. 196–208).

Ranathunga, S. & Liyanage, I. U. (2021). Sentiment analysis of sinhala news comments. *Transactions on Asian and Low-Resource Language Information Processing*, *20* (4), 1–23.

Ribeiro, M. T., Singh, S. & Guestrin, C. (2016). " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (S. 1135–1144).

Risch, J. & Krestel, R. (2020a). Top comment or flop comment? predicting and explaining user engagement in online news discussions. In *Proceedings of the international aaai conference on web and social media* (Bd. 14, S. 579–589).

Risch, J. & Krestel, R. (2020b). Toxic comment detection in online discussions. *Deep learning-based approaches for sentiment analysis*, 85–109.

Sandrilla, R. & Devi, M. S. (2022). Fnu-bicnn: Fake news and fake url detection using bi-cnn. *International Journal of Advanced Computer Science and Applications*, *13* (2).

Sarker, A., Chandrashekar, P., Magge, A., Cai, H., Klein, A. & Gonzalez, G. (2017). Discovering cohorts of pregnant women from social media for safety surveillance and analysis. *Journal of medical Internet research*, *19* (10), e361.

Sarsam, S. M., Al-Samarraie, H., Alzahrani, A. I. & Wright, B. (2020). Sarcasm detection using machine learning algorithms in twitter: A systematic review. *International Journal of Market Research*, *62* (5), 578–598.

Schabus, D., Skowron, M. & Trapp, M. (2017). One million posts: A data set of german online discussions. In *Proceedings of the 40th international acm sigir conference on research and development in information retrieval* (S. 1241–1244).

Scheible, R., Frei, J., Thomczyk, F., He, H., Tippmann, P., Knaus, J., ... Boeker, M. (2024, November). GottBERT: a pure German language model. In Y. Al-Onaizan, M. Bansal & Y.-N. Chen (Hrsg.), *Proceedings of the 2024 conference on empirical methods in natural language processing* (S. 21237–21250). Miami, Florida, USA: Association for Computational Linguistics. Zugriff auf `https://aclanthology.org/2024.emnlp-main.1183` doi: 10.18653/v1/2024.emnlp-main.1183

Sharma, K., Qian, F., Jiang, H., Ruchansky, N., Zhang, M. & Liu, Y. (2019). Combating fake news: A survey on identification and mitigation techniques. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *10* (3), 1–42.

Shmargad, Y. & Klar, S. (2020). Sorting the news: How ranking by popularity polarizes our politics. *Political Communication*, *37* (3), 423–446.

Singh, V. K., Ghosh, I. & Sonagara, D. (2021). Detecting fake news stories via multimodal analysis. *Journal of the Association for Information Science and Technology*, *72* (1), 3–17.

Smith, E. A. & Senter, R. (1967). *Automated readability index* (Bd. 66) (Nr. 220). Aerospace Medical Research Laboratories, Aerospace Medical Division, Air ....

Stemmer, M., Parmet, Y. & Ravid, G. (2022). Identifying patients with inflammatory bowel disease on twitter and learning from their personal experience: retrospective cohort study. *Journal of medical Internet research*, *24* (8), e29186.

Tate, R. F. (1954). Correlation between a discrete and a continuous variable. point-biserial correlation. *The Annals of mathematical statistics*, *25* (3), 603–607.

Thilagam, P. S. et al. (2023). Multi-layer perceptron based fake news classification using knowledge base triples. *Applied Intelligence*, *53* (6), 6276–6287.

Thompson, K. (1968). Programming techniques: Regular expression search algorithm. *Communications of the ACM*, *11* (6), 419–422.

Trujillo, A. & Cresci, S. (2022). Make reddit great again: assessing community effects of moderation interventions on r/the_donald. *Proceedings of the ACM on Human-computer Interaction*, *6* (CSCW2), 1–28.

Tsagkias, M., Weerkamp, W. & De Rijke, M. (2009). Predicting the volume of comments on online news stories. In *Proceedings of the 18th acm conference on information and knowledge management* (S. 1765–1768).

Wehrli, S., Arnrich, B. & Irrgang, C. (2023, September). German text embedding clustering benchmark. In M. Georges, A. Herygers, A. Friedrich & B. Roth (Hrsg.), *Proceedings of the 19th conference on natural language processing (konvens 2023)* (S. 187–201). Ingolstadt, Germany: Association for Computational Lingustics. Zugriff auf https://aclanthology.org/2023.konvens-main.20/

Wiedemann, G., Ruppert, E., Jindal, R. & Biemann, C. (2018). Germeval-2018 task 14: Transfer learning from lda to bilstm-cnn for offensive language detection in twitter. In *14th conference on natural language processing konvens 2018*.

Zhang, S., Zheng, D., Hu, X. & Yang, M. (2015). Bidirectional long short-term memory networks for relation classification. In *Proceedings of the 29th pacific asia conference on language, information and computation* (S. 73–78).

Zhuang, L., Wayne, L., Ya, S. & Jun, Z. (2021, August). A robustly optimized BERT pre-training approach with post-training. In S. Li et al. (Hrsg.), *Proceedings of the 20th chinese national conference on computational linguistics* (S. 1218–1227). Huhhot, China: Chinese Information Processing Society of China. Zugriff auf https://aclanthology.org/2021.ccl-1.108

Zosa, E., Shekhar, R., Karan, M. & Purver, M. (2021, September). Not all comments are equal: Insights into comment moderation from a topic-aware model. In R. Mitkov &

G. Angelova (Hrsg.), *Proceedings of the international conference on recent advances in natural language processing (ranlp 2021)* (S. 1652–1662). Held Online: INCOMA Ltd. Zugriff auf https://aclanthology.org/2021.ranlp-1.185