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
Mitwirkung:

Wien, 1. Dezember 2024

| 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
Assistance:

Vienna, December 1, 2024

Dario Bogenreiter | Gábor Recski

# 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, 1. Dezember 2024

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 bei 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

Ihr Text hier.

# Abstract

In the contemporary digital communication landscape, the influence wielded by news agencies, users, and even bots is undeniable. Their ability to shape public opinion, drive agendas, and potentially sway elections has sparked interest in understanding the factors governing post popularity. While previous research on post popularity prediction has predominantly focused on platforms like Twitter, this thesis ventures into uncharted territory by analyzing the One Million Posts Corpus, sourced from the Austrian daily newspaper *Der Standard*. By delving into this dataset, which represents a unique demographic and content context, this study aims to unravel the details of post-engagement dynamics and showcase the underlying mechanisms that lead to popularity trends.

# Contents

CHAPTER 1

# Introduction

## 1.1 Motivation & Problem Statement

Even before the 2017 US presidential election between Trump and Clinton, it was evident that user posts on social media and news websites had a significant influence. They can shape opinions, drive agendas, and potentially 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). 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. Nevertheless, since they allow users to create posts, share personal stories, and interact with each other, these websites share major characteristics of standard social media websites. Although their influence on the political discussion can not be neglected, only a handful of researchers like Risch und Krestel (2020a) have explored predictive models for these kinds of websites. Additionally, most studies have focused on English texts and news, neglecting German use cases that may have different 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. 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 explainable Artificial Intelligence (AI), 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 *Der Standard*, Austria's leading platform for political discussions. The 'One Million Posts Corpus' (Schabus, Skowron & Trapp, 2017), which contains over one million user posts from *Der Standard*, 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 based on the same criteria.

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. 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?

R2. Do the explainable features created in this work differentiate significantly between the two classes of "engaging" and "regular posts"?

These research questions correspond to the following null hypothesis:

$H_0$1. There is no significant difference in feature importance.

$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) model) and the explainable models 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 applied 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 an in-depth discussion and analysis in Section 7. Finally, Section 8 concludes the thesis by summarizing key findings, discussing 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 *Der Standard*. 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 *Der Standard* 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.

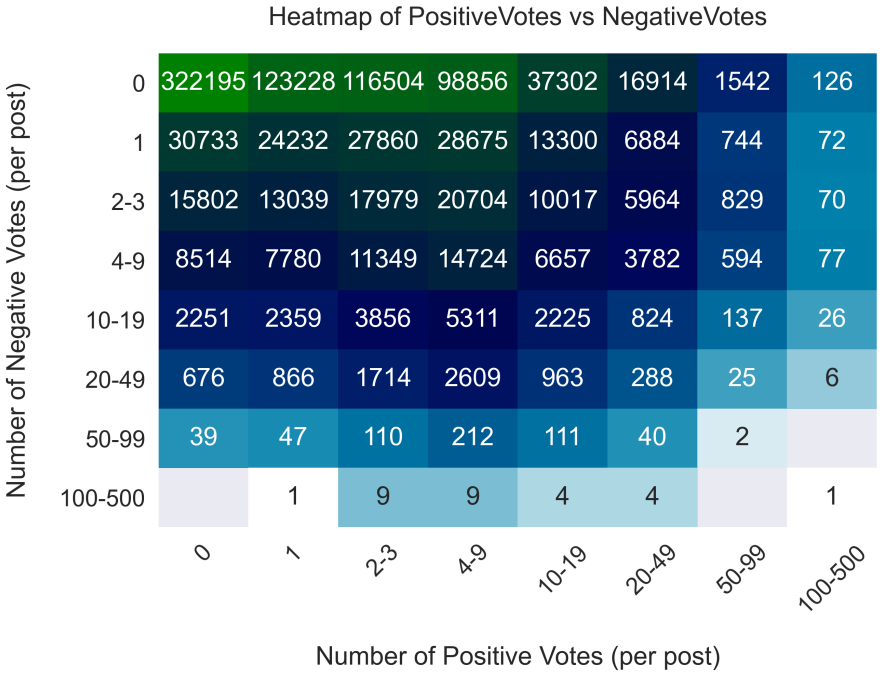


Figure 2.1: Heatmap of Posts, clustered by Up- and Downvotes

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 Top Posts**

One reason why only a few posts receive substantially more votes than others is that highly engaging posts are highlighted on the *Der Standard* website, as shown in Figure 2.2. These comments are listed above the text input field and all other posts. Comments are pinned if they receive more interaction than the remaining posts. Each article can have one or a few pinned comments (less than 10), and they remain in this prominent position until other posts receive even more attention.

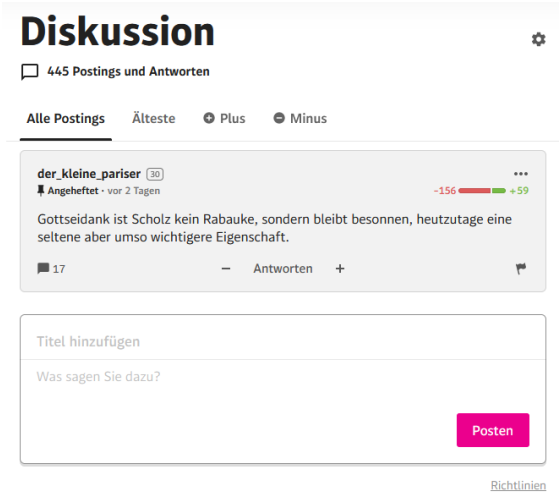


Figure 2.2: Example of a Top Comment

This layout, shown in Figure 2.2, boosts the visibility of engaging posts written shortly after the article is released. Later comments, even if more interesting, have little chance of becoming pinned because many users do not scroll further down to see them. The prominent position enables top posts to collect more and more votes, making it extremely difficult for later comments to accumulate a similar amount of up- and downvotes. This is why most other newspaper platforms have reverted to listing posts solely in chronological order (Risch & Krestel, 2020a). As of 2024, even *Der Standard* appears to have switched to a purely chronological ordering of posts. However, at the time the dataset was created, this bias still existed.

**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: Result Distortion through 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 Tree**

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

metrics for posts that do not add much meaningful content on their own, but instead derive their popularity from the shortcomings of others. Figure **??** illustrates such an example.

**Bias 5: Unique User Group**

The dataset reflects interactions from a specific demographic group of users. Approximately 46 percent of all *Der Standard* 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 baseline solutions, they employed a Bag of Words (BOW) model in conjunction with a Support Vector Machine (SVM). Additionally, they explored more sophisticated solutions, such as a doc2vec (D2V) document embedding combined with an SVM, and a neural network architecture using a LSTM model (Schabus et al., 2017). The performance of these models was evaluated using precision, recall, and F1 score (Schabus et al., 2017). Interestingly, the performance varied across the models and categories (Schabus et al., 2017). The BOW + SVM model provided robust results across most tasks, achieving the highest precision in two cases. In contrast, the more advanced LSTM model outperformed others in four task-metric combinations but notably failed to identify any posts with positive sentiment (Schabus et al., 2017). This highlights the complexity of text classification tasks and the varying effectiveness of different models depending on the specific label and evaluation metric.

In 2018, Wiedemann, Ruppert, Jindal und Biemann (2018) explored the 11,773 labeled posts from the dataset to investigate the potential of transfer learning in this domain. Their approach involved supervised pre-training of a neural model for 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). Despite this effort, the model

demonstrated only minor improvements (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.

In 2020, 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 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 15 Named Entity Recognition (NER) tasks and another classification task (Scheible et al., 2024).

Although GottBERT outperformed foreign and multilingual models in all NER tasks and the additional classification task, its performance on topic classification was unexpectedly weak. It was outperformed in terms of F1 score by other BERT models and even by the multilingual XLM-RoBERTa, contrary to expectations, as multilingual models typically perform worse than monolingual transformers (Scheible et al., 2024). The authors

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

suggested that this underperformance might be due to suboptimal hyperparameter configurations during training (Scheible et al., 2024).

## 3.2 Popularity Prediction

In 2020, Risch and Krestel studied predicting the number of upvotes and replies to posts on the newspaper *The Guardian* (Risch & Krestel, 2020a). This research is closely related to this study but differs in three ways: they did not consider downvotes, applied different algorithmic approaches, and the scenario is entirely different (English texts with posts ranked chronologically rather than by votes received). Their primary motivation was to demonstrate an automated method for identifying engaging posts without expensive manual annotation effords 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).

They defined popularity by the number of upvotes a post received and engagement as a combination of the adjusted number of upvotes and replies (Risch & Krestel, 2020a). This definition was justified by the assumption that users upvote posts that interest them the most and that users do not manipulate votes, as upvoting does not affect post ranking (posts are chronologically sorted on *The Guardian*) (Risch & Krestel, 2020a). This assumption does not hold for this research, as more upvotes on a post on *Der Standard*'s website result in a higher ranking. Their target variable, "top" and "flop" posts, were defined based on the relative share of upvotes each post received compared to all posts under one article (Risch & Krestel, 2020a).

For predicting "top" and "flop" comments, they employed four approaches: a baseline logistic regression trained on text length, logistic regression with features proposed by Park (Park et al., 2016), Convolutional Neural Network (CNN), and their newly designed recurrent neural network (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 by a few to a maximum of 10 percent in accuracy, depending on the percentage of comments identified as "top comments" (Risch & Krestel, 2020a).

Most interestingly, they identified challenges similar to those of the 'One Million' dataset, mentioned in Section 2 of this paper, such as earlier comments receiving more upvote, which they referred to as position bias, as comments are ranked chronologically (Risch & Krestel, 2020a). They corrected this bias by grouping comments into ranks according to when they were written and then calculating "top" and "flop" relative to these ranks (Risch & Krestel, 2020a). Another challenge they faced, which also exists in 'One Million' dataset, is that some articles have very few comments, making it difficult to correctly estimate popularity. They addressed this by removing such posts from their analysis (Risch & Krestel, 2020a).

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 anyways 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 a 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 - where used for further analysis. Furthermore, this study focused on finding the explainable features, more complex features such as embeddings, are only briefly mentioned, as this 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 in 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 19 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 5 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 6 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. |
| | AnglicismsFreq | Frequency of anglicisms in the text. |
| | 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 (`!`). |
| | 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)** | ShortWordsFreq | Frequency of short words (1-3 char.). |
| | LongWordsFreq | Frequency of long words (14+ char.). |
| | LongWordComboFreq | Sequences with two or more long words. |
| | ExaggerateFreq | Custom pattern to identify exaggeration |
| | RepeatedWordsFreq | Custom pattern to identify repetition |
| | OnlySymbolsFreq | Text that contains only symbols. |
| | OnlyWordsFreq | Text that contains only words. |
| | NoCapsShortFreq | Lowercase-only short texts. |
| | OnlyLinkFreq | Text that contains only links. |
| | FormalityFreq | Custom pattern to recognize formality |
| | DesireFreq | Custom pattern to identify desire |
| | AffectionFreq | Custom pattern to recognize formality |
| | RandomFeature0 | Random feature used for testing the validity of the approach. Randomly oscillating between 0 and 300. |
| | RandomFeature1 | Randomly oscillating between 0 and 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 new set of custom features tailored to the use case at hand. These features draw some inspiration from existing literature but are primarily derived from patterns observed in the data itself.

The approach aims to uncover recurring patterns in posts that consistently receive higher or lower votes than average, which could serve as valuable predictors.

**Pattern: "Only Symbols"**

This pattern serves as a negative predictor for popularity. It matches strings consisting only of symbols, which could indicate a post with little or no meaningful content, making it less likely to be desired by users.

Regex code:

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

Real World Example:

| Pattern | FullText | UpVotes | article_vote_median |
|---|---|---|---|
| OnlySymbols | 6666? | 0 | 1.000000 |
| OnlySymbols | 49,7% :) | 0 | 2.000000 |
| OnlySymbols | 4.500*14=63.000 | 0 | 1.000000 |

Figure 4.1: Example of a post with only symbols.

**Pattern: "ShortPhrase"**

This pattern serves as a positive predictor for popularity. It matches short phrases consisting of two words followed by punctuation, which could indicate concise and impactful content that is more likely to attract user attention.

Regex code:

```
\w{2,}\.{1} +\w{2,}(\.|\,)
```

Real World Example:

**Pattern: "LongWords"**

This pattern identifies long words that may represent technical jargon or deep, descriptive content, which could indicate a more in-depth, specialized post likely to appeal to certain user groups.

Regex code:

| Pattern | FullText | UpVotes | article_vote_median |
|---|---|---|---|
| ShortPhrase | Noch einmal: Irgendwelche Vorzeigeschulen, mit denen man alle blenden kann, interessieren mich nicht. Die Realität ist eine andere. Das, was Sie vorschlagen, ist ungefähr so realistisch wie wenn Sie eine Mindestsicherung von 2500 Euro vorschlagen würden. Außerdem müssen Sie sich vom Gedanken verabschieden, dass Ganztagsschulen Kinder bildungsferner Eltern "umpolen". Das funktioniert auch in anderen Ganztagsschulländern wie Frankreich, GB und den USA nicht. Auch nicht in Schweden. Das ist reine HH-Propaganda. | 1 | 0.000000 |
| ShortPhrase | Warum brauchen wir einander? Na eben, wir brauchen eben einander nicht (bis auf Microsoft und Apple) Endlich kommt man drauf, dass Europa die USA nicht braucht. Ein Geistesblitz in der Geschichte. Spät, aber doch noch! Warum braucht Europa Russland? Weil Flatulenzen im Winter nicht ausreichen, um die Wohnung zu beheizen! | 7 | 2.000000 |
| ShortPhrase | Also bitte, wenn, dann erinnert mich das magische Dreieck an goldene Sturm-Zeiten, aber sicher nicht an den Vfb. Tz. | 27 | 1.000000 |

Figure 4.2: Example of a short phrase.

```
\b\w{14,}\b
```

Real World Example:

| Pattern | FullText | UpVotes | article_vote_median |
|---|---|---|---|
| LongWords | Naja, ist es hier auf derstandard.at anders? Einige User posten aus irgendwelchen scheinheiligen Quellen Texte, damit Sie ihre "Erkenntnisse" penetrant den Benutzern präsentieren können (zwei Beispiele al s Persiflage :"Wissenschaftler haben herausgefunden, dass Leute alles glauben, wenn man behauptet, Wissenschaftler hätten es herausgefunden!"; oder ganz kompliziert erläuterte Sachverhalte, damit es glaubwürdig erscheint à la "Finanzmarkt- und Konzernmacht-Zeitalter der Plutokratie unterstützt von der Mediakratie in den Lobbykraturen...). Diese Meinungen werden hier, auf FB,... "geliked" (verflixtes Denglisch), weil es ja, wie bereits aus wissenschaftl. Texten bekannt, gut belegt ist und so verbreitet sich die geistige "Bullsh*t". | 5 | 2.000000 |
| LongWords | 25% an Uber? Für was? Ich bin der Meinung, dass Taxilenkerprüfungen seit es Navis gibt unnötig geworden sind und dass das gewerbe ein wenig liberalisiert gehört. Statt Uber sollte es eine Art staatliche App geben über welcher gleich die Steuern kassiert werden. Und alle Fahrer sollten natürlich versichert sein. Uber hatte ich am Anfang begrüßt, da es die teuren Taxivermittler quasi umging, aber die sind jetzt noch schlimmer als die Vermittler geworden. | 7 | 1.000000 |
| LongWords | Er hat einfach Recht. Ohne Reformwillen bringt es nichts sich zu Reformen zu verpflichten. Am besten die GR machen ihr eigenes Ding und treffen sich einmal die Woche unter den Olivenbäumen und singen die Internationale. Bis zur Einsicht dass mit linker Politik kein Staat zu machen ist, ist offenbar noch ein viel tieferer Abstieg nötig. | 5 | 1.000000 |

Figure 4.3: Example of long words.

### Pattern: "LongWordCombo"

This pattern detects posts with two long words in succession, which may suggest detailed, complex information that could either be highly engaging or overly technical for casual users.

Regex code:

```
\b\w{11,} +\w{11,}\b
```

Real World Example:

| Pattern | FullText | UpVotes | article_vote_median |
|---|---|---|---|
| LongWordCombo | Naja, ist es hier auf derstandard.at anders? Einige User posten aus irgendwelchen scheinheiligen Quellen Texte, damit Sie ihre "Erkenntnisse" penetrant den Benutzern präsentieren können (zwei Beispiele al s Persiflage :"Wissenschaftler haben herausgefunden, dass Leute alles glauben, wenn man behauptet, Wissenschaftler hätten es herausgefunden!"; oder ganz kompliziert erläuterte Sachverhalte, damit es glaubwürdig erscheint à la "Finanzmarkt- und Konzernmacht-Zeitalter der Plutokratie unterstützt von der Medialkratie in den Lobbykraturen...). Diese Meinungen werden hier, auf FB,... "geliked" (verflixtes Denglisch), weil es ja, wie bereits aus wissenschaftl. Texten bekannt, gut belegt ist und so verbreitet sich die geistige "Bullsh*t". | 5 | 2.000000 |
| LongWordCombo | Noch einmal: Irgendwelche Vorzeigeschulen, mit denen man alle blenden kann, interessieren mich nicht. Die Realität ist eine andere. Das, was Sie vorschlagen, ist ungefähr so realistisch wie wenn Sie eine Mindestsicherung von 2500 Euro vorschlagen würden. Außerdem müssen Sie sich vom Gedanken verabschieden, dass Ganztagsschulen Kinder bildungsferner Eltern "umpolen". Das funktioniert auch in anderen Ganztagsschulländern wie Frankreich, GB und den USA nicht. Auch nicht in Schweden. Das ist reine HH-Propaganda. | 1 | 0.000000 |
| LongWordCombo | ich verstehe nicht so recht, warum die eu so 'erpressbar' ist. wenn erdogan damit droht, flüchtlinge zu schicken, so kann man die grenzen zur türkei schließen bzw. scharfe kontrollen einführen - mit enormen, negativen wirtschaftlichen auswirkungen für die türkei. praktischerweise hat sich die türkei international ohnehin fast isoliert, würde dann auch noch den letzten ernsthaften partner auf diesem kontinent verlieren. außerdem könnte die eu auch einmal andeuten, dass man großes interesse an einem kurdenstaat hat, und ein solches bestreben auch unterstützt (sollte die eu sowieso). alleine das wäre drohung genung. | 63 | 3.000000 |

Figure 4.4: Example of two long words in succession.

### Pattern: "RepeatedWords"

This pattern identifies repeated words, which can be indicative of emphasis or redundancy, and may be a signal for posts that generate stronger reactions or engagement.

Regex code:

```
(\b\w{4,}\b).*\1
```

Real World Example:

| Pattern | FullText | UpVotes | article_vote_median |
|---|---|---|---|
| RepeatedWords | Naja, ist es hier auf derstandard.at anders? Einige User posten aus irgendwelchen scheinheiligen Quellen Texte, damit Sie ihre "Erkenntnisse" penetrant den Benutzern präsentieren können (zwei Beispiele al s Persiflage :"Wissenschaftler haben herausgefunden, dass Leute alles glauben, wenn man behauptet, Wissenschaftler hätten es herausgefunden!"; oder ganz kompliziert erläuterte Sachverhalte, damit es glaubwürdig erscheint à la "Finanzmarkt- und Konzernmacht-Zeitalter der Plutokratie unterstützt von der Medialkratie in den Lobbykraturen...). Diese Meinungen werden hier, auf FB,... "geliked" (verflixtes Denglisch), weil es ja, wie bereits aus wissenschaftl. Texten bekannt, gut belegt ist und so verbreitet sich die geistige "Bullsh*t". | 5 | 2.000000 |
| RepeatedWords | 25% an Uber? Für was? Ich bin der Meinung, dass Taxilenkerprüfungen seit es Navis gibt unnötig geworden sind und dass das gewerbe ein wenig liberalisiert gehört. Statt Uber sollte es eine Art staatliche App geben über welcher gleich die Steuern kassiert werden. Und alle Fahrer sollten natürlich versichert sein. Uber hatte ich am Anfang begrüßt, da es die teuren Taxivermittler quasi umging, aber die sind jetzt noch schlimmer als die Vermittler geworden. | 7 | 1.000000 |
| RepeatedWords | Er hat einfach Recht. Ohne Reformwillen bringt es nichts sich zu Reformen zu verpflichten. Am besten die GR machen ihr eigenes Ding und treffen sich einmal die Woche unter den Olivenbäumen und singen die Internationale. Bis zur Einsicht dass mit linker Politik kein Staat zu machen ist, ist offenbar noch ein viel tieferer Abstieg nötig. | 5 | 1.000000 |

Figure 4.5: Example of repeated words.

### Pattern: "FirstSing"

This pattern detects first-person singular pronouns, which may indicate personal opinions, making posts more relatable and potentially more engaging.

Regex code:

```
\b(ich|mich|mir|mein(e|r)*)\b
```

Real World Example:

| Pattern | FullText | UpVotes | article_vote_median |
|---|---|---|---|
| FirstSing | 25% an Uber? Für was? Ich bin der Meinung, dass Taxilenkerprüfungen seit es Navis gibt unnötig geworden sind und dass das gewerbe ein wenig liberalisiert gehört. Statt Uber sollte es eine Art staatliche App geben über welcher gleich die Steuern kassiert werden. Und alle Fahrer sollten natürlich versichert sein. Über hatte ich am Anfang begrüßt, da es die teuren Taxivermittler quasi umging, aber die sind jetzt noch schlimmer als die Vermittler geworden. | 7 | 1.000000 |
| FirstSing | meine top 5 1. "lawrence of arabia" - musik von maurice jarre. der film ist ein meisterwerk in jeder hinsicht. https://www.youtube.com/watch?v=ro7ustPLuc 2. "out of africa" - john barry - https://www.youtube.com/watch?v=XUXs8dGxWyo die flugszene... 3. "exodus" - ernest gold https://www.youtube.com/watch?v=jsmZeo1TcSA 4. "the sting" (dt. "der clou") - scott joplin 5. "breakfast at tiffany's" - henry mancini | 9 | 1.000000 |
| FirstSing | ich fliege im schnitt ca 20x im jahr.....mich kostet also das mitführen eines naßrasierers 300euro/jahr, sonstige dinge, die ich nicht mit dem handgepäck mitnehmen darf, sind auch dabei... dafür geht mein koffer bei umsteigeflügen 2-3x pro jahr nicht mit... dazu kommen die oftmaligen streiks bei lufthansa, und auch bei aua... getränk bei aua frei, bei zugehöriger brussels zu bezahlen ...also eine ganz schlechte nachricht für mich... | 17 | 1.000000 |

Figure 4.6: Example of first-person singular pronouns.

### Pattern: "Uncompassion"

This pattern detects impersonal or dismissive language, which could suggest a lack of empathy, potentially making posts feel cold or disengaging.

Regex code:

```
^(?=.{1,23}$).*\b(lol|wtf|ok|also|eh|ja|nein|schon|\.\.\.|hm|klar|toll|passt)\b
```

Real World Example:

| Pattern | FullText | UpVotes | article_vote_median |
|---|---|---|---|
| Uncompassion | ja, und? | 0 | 3.000000 |
| Uncompassion | Oh Gott, schon weider | 0 | 1.000000 |
| Uncompassion | ja, in ägypten | 0 | 2.000000 |

Figure 4.7: Example of uncompassionate language.

CHAPTER 5

# Experimental Setup: 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 (as outlined in section 2 under 'The Privilege of Top Posts'). Consequently, determining the exact number of votes as a regression task is not particularly insightful, as the primary reason posts get vast amounts of votes is the ranking system itself and only partially the actual content of a post. Posts with inherently engaging content, written after the pinned posts, might still receive fewer votes due to their later appearance.

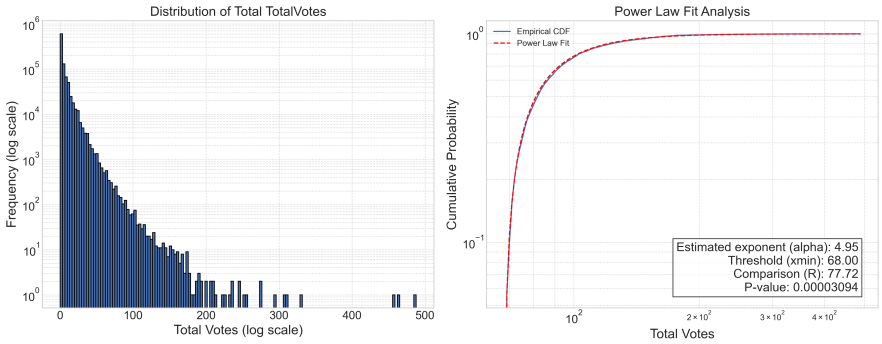


Figure 5.1: Distribution of Total Votes per Post

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 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 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.

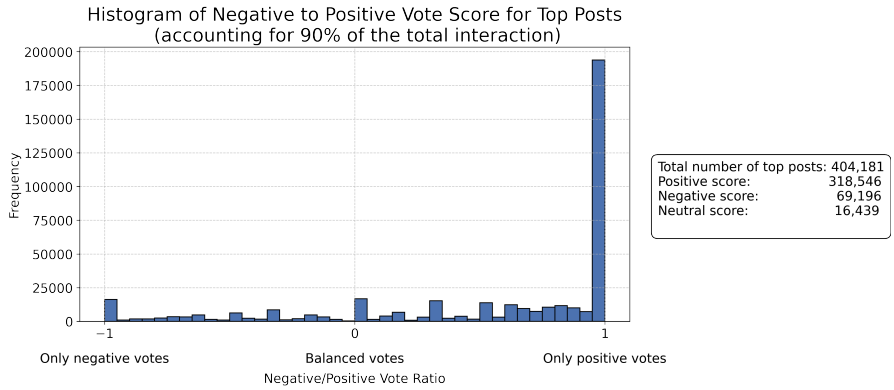


Figure 5.2: Frequency of Upvote-Downvote Scores for Engaging Posts

### 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.

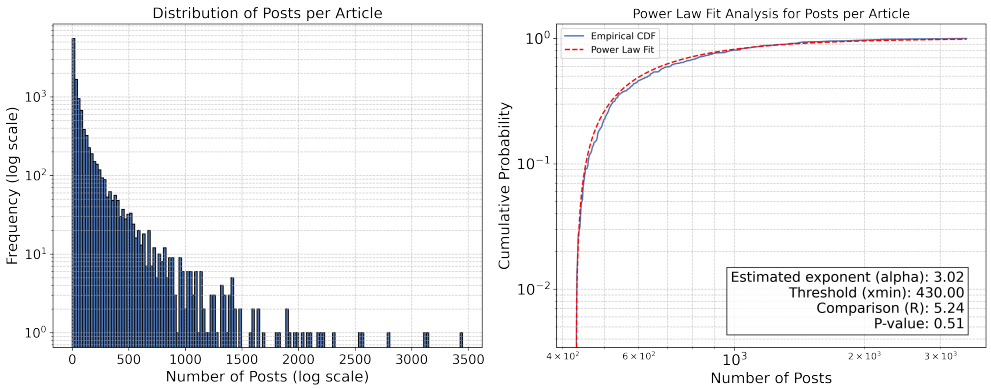


Figure 5.3: Distribution of Number of Posts per Article

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.

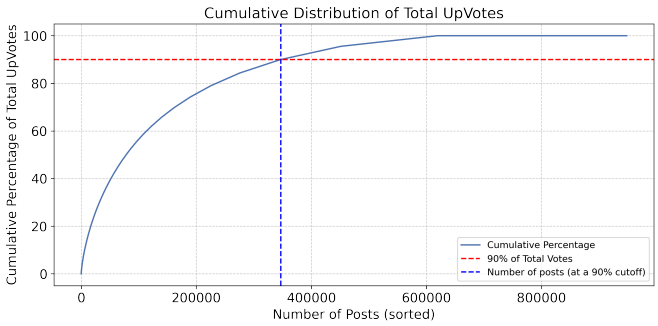


Figure 5.4: Total Votes Cumulative and 90% Interaction Threshold

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.

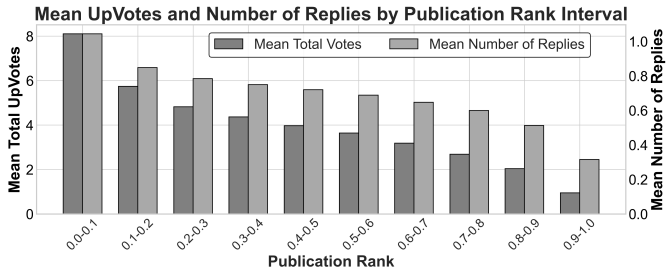


Figure 5.6: Relationship between Votes Received and Time Passed

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 50% of posts in each group as *top posts* and the bottom 50% as *flop posts* (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 we fit them onto a polynomial regression model.

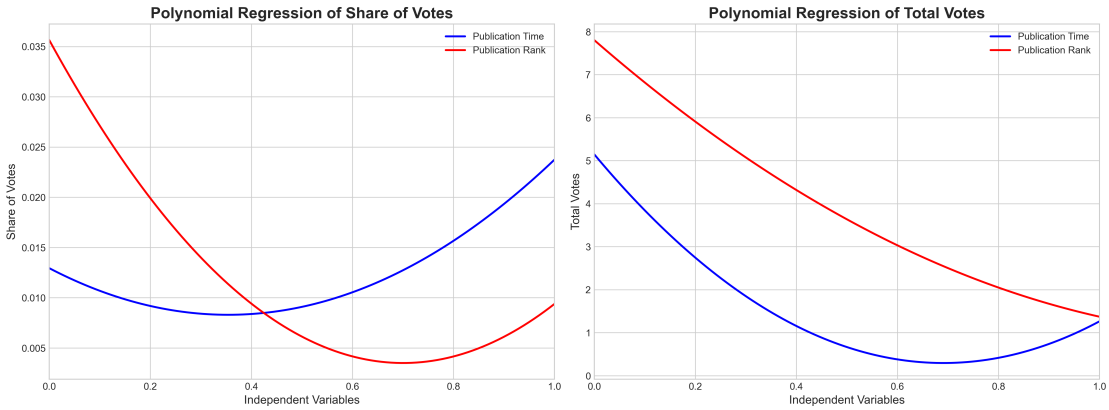


Figure 5.7: Influence of the time variables on ShareOfVotes and TotalVotes

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 *Der Standard* however work different than that.

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.

**Algorithm 5.1:** Post Filtering and Engagement Labeling

**Input:** Dataset **D** with posts, including publication time, rank, and votes.
**Output:** Labeled dataset $\mathbf{D}'$ with posts categorized as *engaging* or *regular*.

```
Compute combined time rank based on publication time and rank;
foreach article_ID do
    Normalize combined time rank to [0, 1];
    Group posts into percentile-based time intervals for each article;
end
foreach interval t do
    if t has fewer than 10 posts then
        Assign all posts in t to one group;
    else
        Order posts by combined time rank;
        while posts available do
            Fill up groups with 10 posts, and start a new group if full;
        end
    end
end
Compute total votes, post counts, and vote shares for each group;
Remove groups with fewer than 10 posts, median votes ≤ 0, or total votes ≤ 20;
Label posts with vote share > 6% as engaging, others as regular;
Summarize filtering and labeling statistics;
return D′;
```

## 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 initially assessed 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. Subsequently, the analysis explores the importance of these features within individual predictive models, with a particular focus on their role in random forests, as discussed in later sections.

The Mann-Whitney U test, as introduced by McKnight und Najab (2010), is applied as the primary statistical test because most features do not follow a normal distribution, as determined through exploratory data analysis. However, since some of the features, such as the stopword-to-text ratio, at least partially normal distributions, the independent t-test, as introduced by Kim (2015), is also utilized as additional metric. Both tests examine whether the observed differences in means between the two classes are statistically significant or could have arisen by chance (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 analysis 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 Overview** The process of feature importance testing is summarized in Algorithm 5.2 which 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 $\mathbf{F}$, engagement classes $\mathbf{C}$, significance threshold $\alpha = 0.05$
**Output:** Feature importance results $\mathbf{R}$

**foreach** *feature* $f \in \mathbf{F}$ **do**
  Compute means $\mu_{\text{class }0}, \mu_{\text{class }1}$;
  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 $p < \alpha$;
  Store results in $\mathbf{R}$;
**end**
Sort $\mathbf{R}$ by significance and composite ranking;
**return** $\mathbf{R}$;

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. It is particularly valuable for imbalanced datasets, where achieving a balance between precision and recall is crucial. The F1 score is defined as:

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

  By combining precision and recall, the F1 score ensures a comprehensive evaluation of both aspects of the model's predictive power.

**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:

- Examining individual predictions to understand the model's decision-making process, particularly for posts labeled as highly engaging or polarizing.
- Identifying trends and anomalies in the predicted outcomes to uncover patterns related to content features or user behaviors.
- Comparing findings with existing literature to situate the results within broader discussions on social media dynamics and content engagement.

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

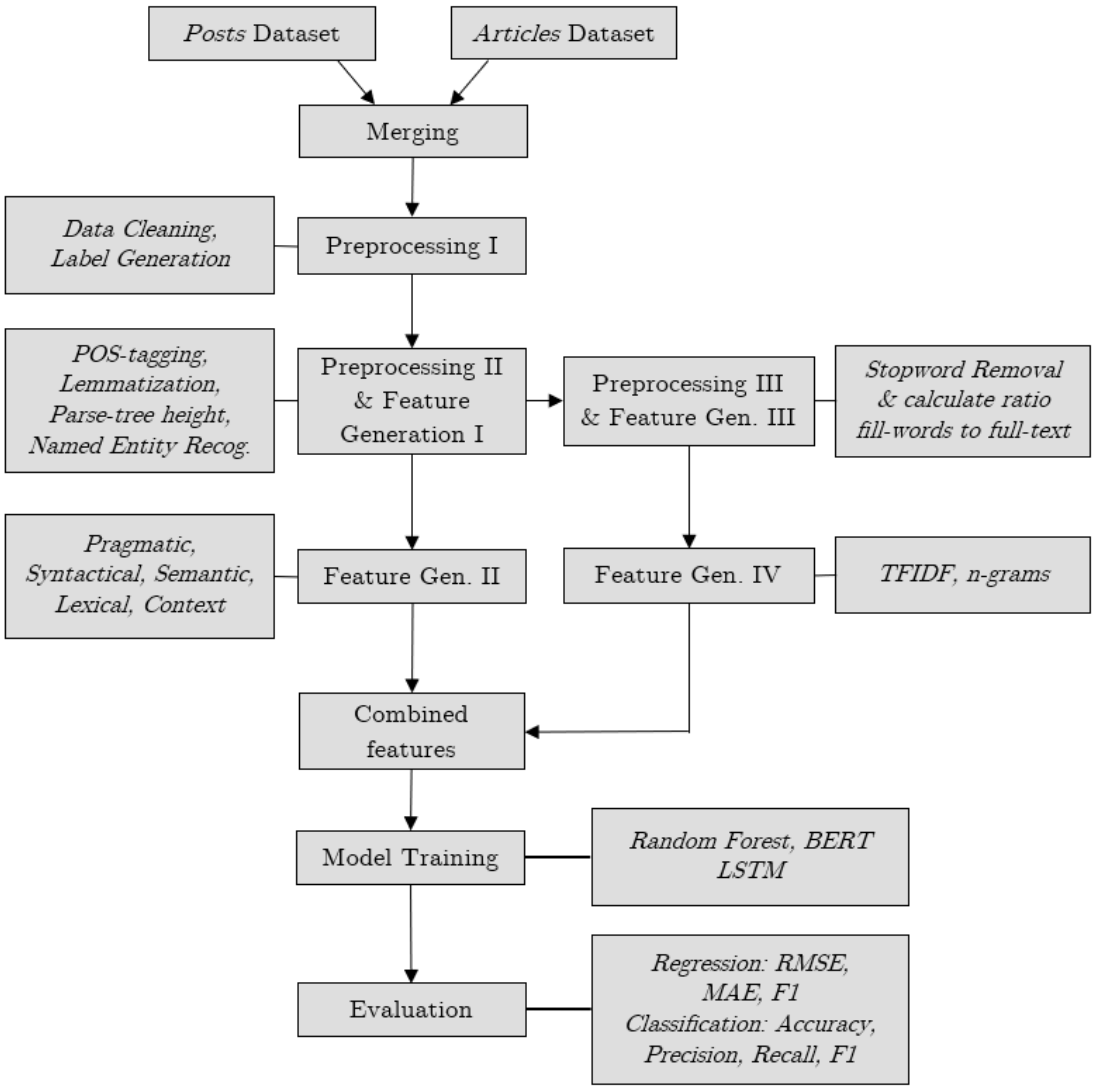


Figure 5.8: Prediction Pipeline: From Dataset Loading to Evaluation

**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**

Several basic features are created in this step. The key feature is the full text, generated by combining both the article body and heading. To ensure that 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. Summary statistics, including total votes for each post, are also computed in this step.

**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 5.9

| 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 5.9: Example of NER

### 5.3.1 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 5.10

**Algorithm 5.4:** Stop Word Removal and Ratio Calculation

**Input:** Text **T**, Custom stopword list **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* **T do**
    **if** *word is in* **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 5.10: Example of Stop Word Removal

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

### 5.3.2 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.

### 5.3.3 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.

### 5.3.4 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.

## 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**

Risch und Krestel (2020a) is incorporated—a logistic regression model using the text length.

**Explainable Model 1: k Nearest Neighbors (KNN)**

Additionally, KNN and

**Explainable Model 2: KNN**

decision trees are employed, and the baseline from the paper by

**Interpretable Model 1: RandomForest + Surface-Level Features**

Model 1 employs a small set of lightweight features, such as word count and punctuation count, combined with a shallow model (Random Forest), as shown in Figure 5.11.

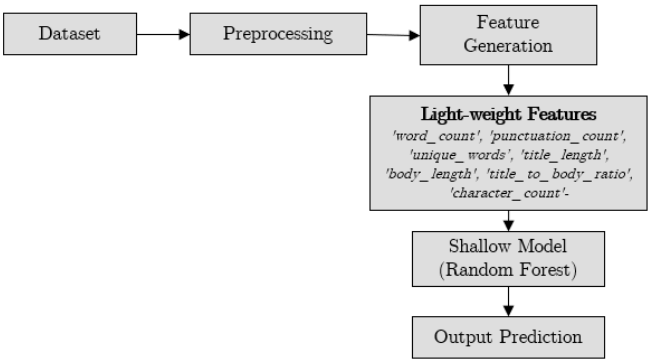


Figure 5.11: Model 1: Lightweight Features + Random Forest

**Interpretable Model 2: RandomForest + Complex Features**

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 5.12.

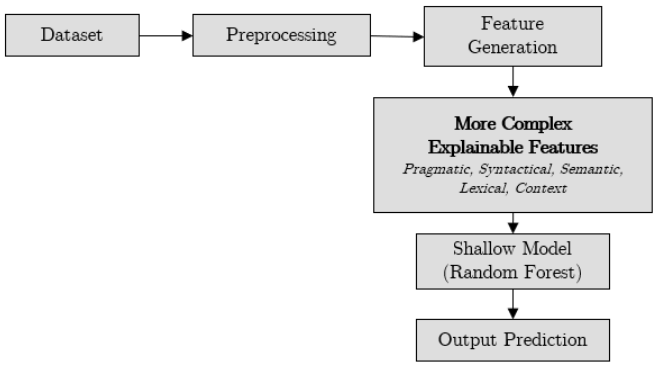


Figure 5.12: Model 2: Complex Features + Random Forest

### Interpretable Model 3: RandomForest + Complex and Sentiment Features

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 5.13.

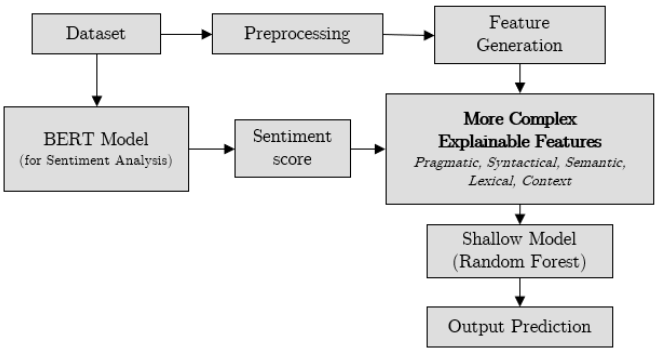


Figure 5.13: Model 3: Semi-Explainable Features + Random Forest

### Deep Learning Model 1: LSTM

Model 4 adopts a deep learning approach using a LSTM model. This model is provided with 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). See Figure 5.14 for the model architecture.

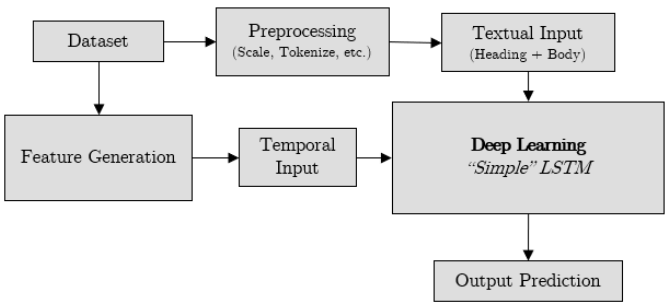


Figure 5.14: Model 4: LSTM with Textual and Temporal Inputs

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

Model 5 is similar to Model 4 but employs a BiLSTM, as shown in Figure 5.15.

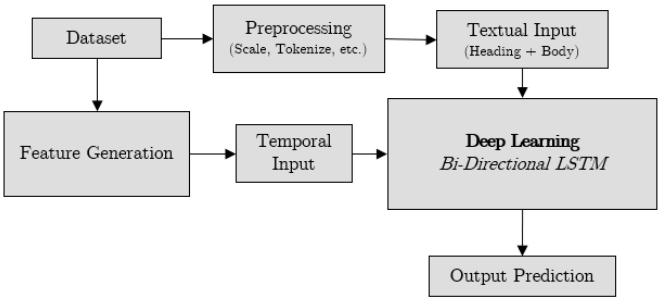


Figure 5.15: Model 5: Bidirectional LSTM

**Deep Learning Model 3: Gated Recurrent Unit (GRU)**

Model 6 employed a GRU (Cho et al., 2014), configured as described by Risch und Krestel (2020a). A GRU is a type of Recurrent Neural Network (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 70% 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.

[2] `https://dl.fbaipublicfiles.com/fasttext/vectors-crawl/cc.de.300.bin.gz`

**Deep Learning Model 4: BERT Standalone**

Model 6 uses DistillBERT [3], a distilled version of the BERT model tailored for the German language. Due to computational limitations, 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.

[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 TFIDF 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 TFIDF 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 interactions (e.g., votes) 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 presents the results obtained by applying algorithm Algorithm 5.2. The table shows that the differences between the two classes are significant for all of the generated features. The features are are ranked based on the magnitude of their absolute Cohen's $d$ value and absolute correlation strength, regardless of direction (positive or negative). 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 | Corr. | PValue MannW. | \|CohenD\| |
|---|---|---|---|---|---|
| SyllableCount | 43.542 | 73.643 | 0.113 | <0.001 | 0.667 |
| CharCount | 167.326 | 280.900 | 0.112 | <0.001 | 0.662 |
| UniqueWords | 22.486 | 36.303 | 0.110 | <0.001 | 0.654 |
| WordCount | 24.910 | 41.347 | 0.110 | <0.001 | 0.653 |
| QuestionFreq | 50.791 | 83.709 | 0.109 | <0.001 | 0.644 |
| PolysyllableCount | 4.672 | 8.344 | 0.111 | <0.001 | 0.636 |
| StopWordFreq | 10.962 | 18.024 | 0.099 | <0.001 | 0.594 |
| ShortWordsFreq | 9.097 | 14.803 | 0.100 | <0.001 | 0.584 |
| CapLetterFreq | 6.850 | 11.884 | 0.097 | <0.001 | 0.542 |
| AvgParseTreeHeight | 1.788 | 2.169 | 0.084 | <0.001 | 0.517 |
| BodyLength | 155.263 | 258.492 | 0.083 | <0.001 | 0.496 |
| PunctCount | 6.282 | 9.293 | 0.082 | <0.001 | 0.451 |
| SimilarityArticleBody | 0.020 | 0.041 | 0.089 | <0.001 | 0.438 |
| LongWordsFreq | 0.653 | 1.218 | 0.082 | <0.001 | 0.434 |
| DiversityRatio | 0.953 | 0.923 | -0.075 | <0.001 | 0.416 |
| AvgSentLength | 9.245 | 11.613 | 0.064 | <0.001 | 0.406 |
| SentCount | 3.140 | 4.350 | 0.073 | <0.001 | 0.400 |
| SimilarityArticleTitle | 0.015 | 0.034 | 0.080 | <0.001 | 0.372 |
| PeriodFreq | 2.485 | 3.557 | 0.067 | <0.001 | 0.369 |
| RepeatedWordsFreq | 0.394 | 0.674 | 0.065 | <0.001 | 0.361 |

Continued on next page

Table 6.1: Comparison of Rule Based Features Between Classes

| Feature | Mean Regular | Mean Engaging | Corr. | PValue MannW. | \|CohenD\| |
|---|---|---|---|---|---|
| TitleLength | 11.838 | 22.003 | 0.075 | <0.001 | 0.354 |
| LongWordComboFreq | 0.128 | 0.265 | 0.056 | <0.001 | 0.283 |
| ModelVerbsFreq | 0.339 | 0.563 | 0.053 | <0.001 | 0.287 |
| ARIScore | 16.813 | 16.030 | 0.041 | <0.001 | 0.227 |
| PunctToTextRatio | 0.052 | 0.037 | -0.050 | <0.001 | 0.315 |
| SMOGScore | 4.839 | 5.415 | 0.056 | <0.001 | 0.346 |
| ExaggerateFreq | 0.371 | 0.577 | 0.048 | <0.001 | 0.243 |
| NoCapsShortFreq | 0.067 | 0.015 | -0.040 | <0.001 | 0.261 |
| FirstPluralFreq | 0.117 | 0.243 | 0.046 | <0.001 | 0.221 |
| QuoteFreq | 0.409 | 0.699 | 0.045 | <0.001 | 0.224 |
| FullCapsFreq | 0.224 | 0.392 | 0.040 | <0.001 | 0.214 |
| ModalObligationFreq | 0.116 | 0.207 | 0.037 | <0.001 | 0.205 |
| FirstSingFreq | 0.430 | 0.635 | 0.039 | <0.001 | 0.181 |
| ModalPossibilityFreq | 0.160 | 0.257 | 0.034 | <0.001 | 0.195 |
| DigitsFreq | 0.991 | 1.468 | 0.037 | <0.001 | 0.183 |
| ExclaimFreq | 0.207 | 0.333 | 0.038 | <0.001 | 0.166 |
| ConjunctiveFreq | 0.208 | 0.305 | 0.026 | <0.001 | 0.162 |
| SecondPluralFreq | 0.027 | 0.057 | 0.024 | <0.001 | 0.121 |
| PastTenseFreq | 0.156 | 0.224 | 0.024 | <0.001 | 0.121 |
| WordsPer100Chars | 15.605 | 14.989 | -0.016 | <0.001 | 0.076 |
| TitleToBodyRatio | 0.144 | 0.153 | 0.016 | <0.001 | 0.052 |
| ShortPhraseFreq | 0.024 | 0.045 | 0.023 | <0.001 | 0.107 |
| UncompassionFreq | 0.009 | 0.001 | -0.016 | <0.001 | 0.116 |
| OnlyWordsFreq | 0.009 | <0.001 | -0.017 | <0.001 | 0.114 |
| FormalityFreq | 0.134 | 0.107 | -0.023 | <0.001 | 0.071 |
| EmojiSarcasticFreq | 0.038 | 0.022 | -0.017 | <0.001 | 0.088 |
| OnlyLinkFreq | 0.005 | <0.001 | -0.014 | <0.001 | 0.092 |
| DesireFreq | 0.030 | 0.048 | 0.016 | <0.001 | 0.087 |
| AffectionFreq | 0.012 | 0.023 | 0.017 | <0.001 | 0.080 |
| SentimentScore | 0.349 | 0.330 | -0.015 | <0.001 | 0.063 |
| AvgWordLength | 6.528 | 5.921 | 0.008 | <0.001 | 0.024 |
| EmojiNoseFreq | 0.034 | 0.022 | -0.014 | <0.001 | 0.074 |
| StopWordsRatio | 0.363 | 0.381 | 0.017 | <0.001 | 0.159 |
| StronglyNegative | 0.363 | 0.386 | 0.014 | <0.001 | 0.048 |
| StronglyPositive | 0.070 | 0.054 | -0.006 | <0.001 | 0.070 |
| ProfanityFreq | 0.005 | 0.008 | 0.010 | <0.001 | 0.041 |
| EmojiNegativeFreq | 0.057 | 0.048 | -0.007 | <0.001 | 0.042 |

Continued on next page

Table 6.1: Comparison of Rule Based Features Between Classes

| Feature | Mean Regular | Mean Engaging | Corr. | PValue MannW. | \|CohenD\| |
|---|---|---|---|---|---|
| OnlySymbolsFreq | 0.002 | <0.001 | -0.006 | <0.001 | 0.045 |
| NoWordsFreq | 0.002 | <0.001 | -0.006 | <0.001 | 0.045 |
| FleschEaseScore | 32.796 | 40.338 | 0.005 | <0.001 | 0.049 |
| CapToLowerRatio | 0.066 | 0.063 | -0.002 | <0.001 | 0.037 |
| AnglicismsFreq | 0.008 | 0.005 | -0.004 | <0.001 | 0.042 |
| RandomFeature0 | 150.045 | 148.806 | -0.004 | 0.020 | 0.014 |
| EmojiPositiveFreq | 0.040 | 0.037 | -0.002 | 0.004 | 0.016 |
| EmojiSurpriseFreq | 0.001 | 0.002 | 0.003 | 0.240 | 0.005 |
| RandomFeature1 | 0.501 | 0.500 | -<0.001 | 0.397 | 0.002 |

### 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 Cohen's $d$ values and absolute correlation strengths. 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. The p-value from the Mann-Whitney test clearly indicates that the differences between the two classes are statistically significant. In general, the presence of named entities in a post serves as a strong indicator of its relevance or interest. However, the infrequent occurrence of many named entities limits their effectiveness in distinguishing between the two classes, as reflected in the low Cohen's d score.

Table 6.2: Comparison of Named Entity Features Between Classes

| Feature | Mean RegularPost | Mean EngagingPost | Corr. | PValue MannW. | CohenD |
|---|---|---|---|---|---|
| Österreich | 0.036 | 0.074 | 0.034 | <0.001 | 0.159 |
| Europa | 0.010 | 0.027 | 0.026 | <0.001 | 0.120 |
| FPÖ | 0.013 | 0.026 | 0.020 | <0.001 | 0.097 |
| SPÖ | 0.007 | 0.017 | 0.021 | <0.001 | 0.094 |
| ÖVP | 0.006 | 0.015 | 0.018 | <0.001 | 0.089 |
| EU | 0.014 | 0.026 | 0.017 | <0.001 | 0.088 |
| Türkei | 0.006 | 0.013 | 0.020 | <0.001 | 0.082 |
| deutsche | 0.018 | 0.032 | 0.016 | <0.001 | 0.080 |
| Merkel | 0.004 | 0.010 | 0.017 | <0.001 | 0.075 |
| Zeit | 0.008 | 0.015 | 0.014 | <0.001 | 0.064 |
| USA | 0.009 | 0.017 | 0.012 | <0.001 | 0.070 |
| Wien | 0.012 | 0.020 | 0.014 | <0.001 | 0.063 |
| Griechen | 0.009 | 0.016 | 0.012 | <0.001 | 0.060 |
| Wiener | 0.002 | 0.006 | 0.013 | <0.001 | 0.054 |
| IS | 0.003 | 0.008 | 0.011 | <0.001 | 0.064 |
| ... | ... | ... | ... | ... | ... |
| Assad | 0.002 | 0.003 | 0.003 | <0.001 | 0.022 |
| Iran | 0.001 | 0.002 | 0.003 | 0.003 | 0.019 |
| Salzburg | 0.002 | 0.002 | 0.002 | 0.037 | 0.012 |
| Schweiz | 0.001 | 0.002 | 0.001 | 0.022 | 0.014 |
| China | 0.002 | 0.002 | -<0.001 | 0.074 | 0.010 |

### 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. However, the effect does not appear to be strong. In fact, the differences between the generated features seem even smaller than those observed for the named entities.

Table 6.3: Comparison of TFIDF Features Between Classes

| Feature | Mean RegularPost | Mean EngagingPost | Corr. | PValue MannW. | CohenD |
|---|---|---|---|---|---|
| herr | 0.005 | 0.012 | 0.027 | <0.001 | 0.113 |
| jahr | 0.014 | 0.024 | 0.023 | <0.001 | 0.112 |
| endlich | 0.003 | 0.010 | 0.026 | <0.001 | 0.113 |
| österreich | 0.011 | 0.020 | 0.023 | <0.001 | 0.108 |
| land | 0.010 | 0.018 | 0.020 | <0.001 | 0.111 |
| europa | 0.005 | 0.011 | 0.023 | <0.001 | 0.102 |
| frau | 0.008 | 0.016 | 0.022 | <0.001 | 0.099 |
| mensch | 0.011 | 0.019 | 0.018 | <0.001 | 0.108 |
| alle | 0.015 | 0.024 | 0.017 | <0.001 | 0.100 |
| muss | 0.016 | 0.024 | 0.017 | <0.001 | 0.094 |
| mann | 0.005 | 0.011 | 0.021 | <0.001 | 0.090 |
| fpö | 0.007 | 0.014 | 0.020 | <0.001 | 0.096 |
| gegen | 0.011 | 0.019 | 0.017 | <0.001 | 0.092 |
| flüchtling | 0.007 | 0.013 | 0.018 | <0.001 | 0.094 |
| övp | 0.005 | 0.011 | 0.019 | <0.001 | 0.087 |
| ... | ... | ... | ... | ... | ... |
| rot | 0.005 | 0.004 | <0.001 | 0.008 | 0.002 |
| genauso | 0.004 | 0.004 | -0.001 | 0.038 | 0.002 |
| schreiben | 0.008 | 0.007 | <0.001 | 0.012 | 0.005 |
| vergessen | 0.005 | 0.004 | -0.001 | 0.083 | 0.008 |
| danke | 0.012 | 0.009 | -0.001 | 0.438 | 0.035 |

### 6.1.4 Evaluation of N-gram Features

As for the n-grams, the situation further deteriorates, as all of the combinations are extremely rare for these features, as shown in Table 6.4. Additionally, it becomes evident that defining certain bigrams and trigrams can lead to the repetition of words, such as in "mad my" and "made my day," where the first combination refers most likely to the same concept.

Table 6.4: Comparison of NGRAM Features Between Classes

| Feature | Mean RegularPost | Mean EngagingPost | Corr. | PValue MannW. | CohenD |
|---|---|---|---|---|---|
| einfach nicht | 0.003 | 0.006 | 0.012 | <0.001 | 0.057 |
| van bellen | 0.001 | 0.003 | 0.012 | <0.001 | 0.053 |
| letzten jahr | 0.002 | 0.005 | 0.011 | <0.001 | 0.046 |
| fpö wähler | <0.001 | 0.002 | 0.010 | <0.001 | 0.045 |
| seit jahr | 0.003 | 0.006 | 0.008 | <0.001 | 0.047 |
| saudi arabien | <0.001 | 0.003 | 0.009 | <0.001 | 0.044 |
| österreich nicht | 0.001 | 0.003 | 0.010 | <0.001 | 0.040 |
| spö övp | <0.001 | 0.002 | 0.010 | <0.001 | 0.039 |
| steuer zahlen | <0.001 | 0.002 | 0.008 | <0.001 | 0.041 |
| jeden tag | 0.001 | 0.002 | 0.008 | <0.001 | 0.036 |
| kein wund | <0.001 | 0.002 | 0.007 | <0.001 | 0.037 |
| frau merkel | <0.001 | 0.002 | 0.008 | <0.001 | 0.035 |
| schwarz blau | <0.001 | 0.002 | 0.008 | <0.001 | 0.032 |
| jedes jahr | <0.001 | 0.002 | 0.007 | <0.001 | 0.034 |
| sorry grün | 0.001 | <0.001 | -0.005 | <0.001 | 0.045 |
| ... | ... | ... | ... | ... | ... |
| stimmen nicht | 0.001 | 0.001 | -0.001 | 0.393 | 0.002 |
| jedenfalls nicht | <0.001 | <0.001 | <0.001 | 0.371 | 0.002 |
| geben viele | 0.001 | 0.001 | -0.001 | 0.429 | 0.002 |
| nicht wissen | 0.001 | 0.002 | -<0.001 | 0.251 | 0.004 |
| nicht lesen | 0.001 | 0.001 | -0.001 | 0.381 | 0.003 |

## 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 BERT and BiLSTM models achieve the highest overall performance. However, it is noteworthy that the Random Forest model trained with explainable features and the sentiment score (computed using the BERT model) achieves competitive results, coming close to BERT's performance. The Random Forest models display a clear improvement when trained on more complex features. While these models are less interpretable than Decision Trees, they effectively balance performance and explainability.

The simpler models, such as Decision Trees and k-Nearest Neighbors (k-NN), while inherently interpretable, perform poorly overall. Both models tend to overfit the training data, as evidenced by their substantially better training performance, that are shown in Table 6.5 on the next page, compared to their generalization capabilities on the test set. The Decision Tree performs only marginally better than the baseline model, which simply predicts the majority class and achieves minimal performance metrics.

The BiLSTM clearly outperforms its unidirectional counterpart, suggesting the benefit of incorporating bidirectional context in the classification task. Furthermore, it is worth to mention that, deep learning models, including LSTMs, would likely have overfitted the data if regularization techniques such as dropout and early stopping were not applied.

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 | 57.62 |
| 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 |

**Random Forest: Feature Importance**

Figure 6.1 shows the feature importance ranking in the Random Forest model, where the values indicate how much each feature contributes to minimizing the Gini impurity. Notably, there is no single feature that stands out significantly in terms of predictive power. Instead, most of the top-ranking features are those that can be calculated for all rows in the dataset, such as the average parse tree height or the character count.

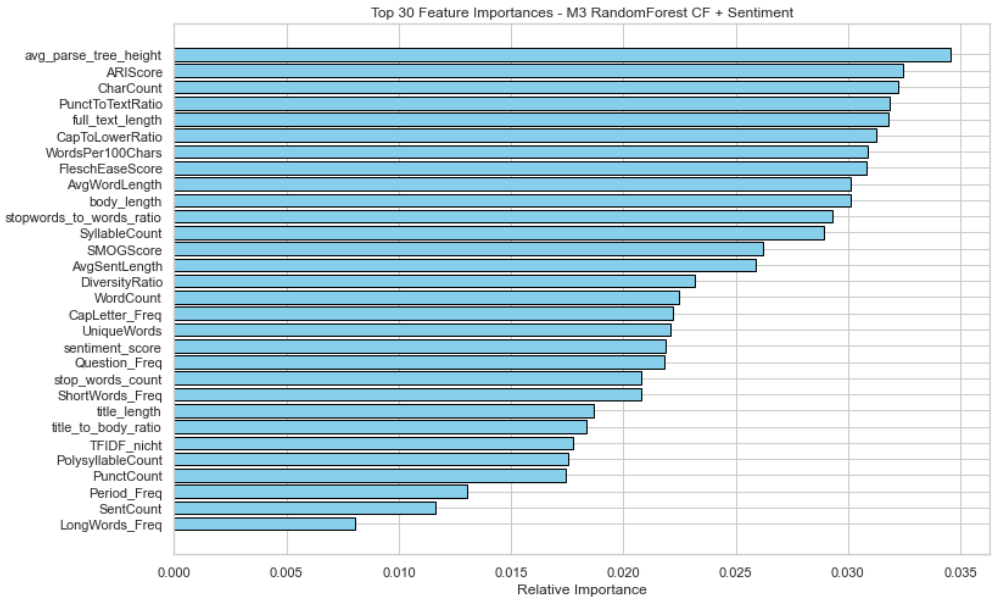


Figure 6.1: Feature Importance in Random Forest

### 6.2.2 Regression Task Performance

| Model | Evaluation | MSE | MAE | R2 |
|---|---|---|---|---|
| Baseline_Regression | Test | 1.35618 | 0.80925 | -0.000293192 |
| ComplexRegressionModel_RF | Test | 1.29268 | 0.76988 | 0.0465441 |
| RF_BERT | Test | 1.29240 | 0.76979 | 0.0467443 |
| LSTM | Test | 1.34257 | 0.78501 | 0.00974544 |
| BiLSTM | Test | 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 interpretability and predictive power. However, their performance remains slightly behind more advanced models like BERT.

Some highly specialized features, such as 'text consisting only of special symbols and spaces,' can be useful for distinguishing between classes, but their rarity limits their effectiveness as strong predictors. Conversely, features that apply to all data instances, such as 'text length,' tend to reveal general trends but frequently lead to misclassifications due to the presence of counterexamples.

Overall, models like random forests tend to favor features that are consistently available across all data points, such as text length or ARI score. While these features may not always be the most decisive, their consistent availability provides a reliable input, making them advantageous in general, even if they lack informativeness in specific cases.

This suggests that generating highly specific features may not be the most effective strategy when aiming to produce features suitable for integration into interpretable or explainable models. Instead, greater focus should perhaps be placed on generating features that are applicable to almost all rows in a dataset.

Another noteworthy insight from the experiments was that the baseline from Risch und Krestel (2020a) performed remarkably well, coming close to the performance of a random forest trained on hundreds of features. However, it performed significantly worse than reported in their paper. One possible explanation for this discrepancy could be the smaller dataset used in this study compared to theirs, as their use case incorporated

substantially more data. Additionally, differences in the layout and functionality of the websites might have influenced the 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 biases or the broader question of whether the number of votes is truly a reliable indicator of what people want to see—or if it simply reflects a platform for exchanging opinions.

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 üerformance 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.

It was shown 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 slightly lower accuracy than more complex 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 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

**ABSA** Aspect Based Sentiment Analysis. 18

**AI** Artificial Intelligence. 2, 19

**ARI** Automated Readability Index. 27, 30, 69

**BERT** Bidirectional Encoder Representations from Transformers. 3, 14, 53, 55, 57, 66, 69, 81, 82

**BiLSTM** Bidirectional Long Term Short Memory. 56, 66

**BOW** Bag of Words. 13, 23

**CNN** Convolutional Neural Network. 16

**GottBERT** German OSCAR text trained BERT. 14

**GRU** Gated Recurrent Unit. 56

**KNN** k Nearest Neighbors. 53, 69

**LDA** Latent Dirichlet Allocation. 28

**LSTM** Long Short Term Memory. 3, 13, 55, 56, 66, 68, 75

**MAE** Mean Absolute Error. 47

**NER** Named Entity Recognition. 14, 18, 23, 50, 63

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

**NLTK** Natural Language Processing Tool Kit. 51

**OFAI** Austrian Research Institute for Artificial Intelligence. 5

**OSCAR** Super-large Crawled Aggregated Corpus. 14

**POS** Part of Speech. 26, 28, 50

**RMSE** Root Mean Squared Error. 47

**RNN** Recurrent Neural Network. 56

**RoBERTa** A Robustly Optimized BERT Pretraining Approach. 14

**SMOG** Simple Measurement of Gobbledygook. 30

**SMPD** Social Media Popularity Prediction. 1

**SVM** Support Vector Machine. 13

**TF-IDF** Term frequency–inverse document frequency. 23, 29, 52, 64

# 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).*

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.

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).

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.

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*.

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.

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*.

McKnight, P. E. & Najab, J. (2010). Mann-whitney u test. *The Corsini encyclopedia of psychology*, 1–1.

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.

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.

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

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.

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

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).

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.*

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