Master Thesis

Controversy Detection in Dutch News

Author: Hannah van Goor

1st supervisor:	dr. Anil Yaman
2nd reader:	prof. dr. Sandjai Bhulai
daily supervisor:	Niya Stoimenova
daily supervisor:	Joao Reis

July 7, 2023

Abstract

Navigating controversial topics in news plays a vital role in fostering social awareness, promoting civil discourse and combating online polarization. The ability to anticipate whether a particular news post will be controversial can, in its function in a bigger system, assist in achieving significant and positive outcomes for reducing polarization, by for example exhibiting opposing views on controversial posts. The benefits of being exposed to a wide range of viewpoints have been widely proven and technology could be utilized to expand people's perspectives. In this research, we investigate which model(s) prove insightful in predicting controversial Dutch news posts. We propose a variety of content-based generalizable modelling approaches to predict controversy in Dutch news posts. Furthermore, we developed the first sizeable data set regarding controversy detection in the Netherlands, of 10k news posts obtained from the 10 largest Dutch news sources, annotated with an entropy measure over the Facebook reactions serving as proxy for controversy. Three different vectorization techniques have been tested; tf-idf, Word2Vec and BERT embeddings. Moreover, a range of traditional machine learning regressors as well as a language model approach have been implemented. As baseline, a dummy regressor which always predicts the mean entropy of the text per source is used. All experiments are set up in a pipeline with hyperparameter tuning and 10-fold cross validation to evaluate the models. The language model yields the best mean squared error; 0.099 (baseline mse: 0.11). Most models outperform the baseline and are thus reasonably successful in predicting controversial news posts. Nevertheless, our work is grounded in the understanding that its effectiveness and ethical implications are deeply intertwined with the socio-technical ecosystem and the actual environment in which it will operate.

Contents

List of Figures **iii**

List of Tables **v**

1 Introduction **1**

2 Related Work **5**

2.1 What causes polarization and controversy? 5

2.2 Prior work on controversy detection 8

2.3 Related domains 12

3 Methodology **14**

3.1 Data 14

3.1.1 Descriptive analysis of the data 15

3.2 Controversy indicator 20

3.3 Text Vectorization 21

3.4 Cross-validation 22

3.5 Models 22

3.5.1 Linear Regression 23

3.5.2 Random Forest 24

3.5.3 XGBoost 25

3.5.4 Support Vector Regressor 25

3.5.5 Hyperparameter tuning for regressors 26

3.6 Language Model Approach 27

3.7 Evaluation Metrics 28

4 Experimental Design **29**
4.1 Preprocessing . . . 30
4.2 Experimental setup approach 1 . . . 31
4.3 Experimental setup approach 2 . . . 32

5 Results **34**
5.1 Baseline . . . 34
5.2 Approach 1: Traditional ML Regressors . . . 35
5.2.1 Linear Regression . . . 35
5.2.1.1 Tf-Idf . . . 35
5.2.1.2 Word2Vec . . . 35
5.2.1.3 BERT . . . 36
5.2.2 XGBoost . . . 36
5.2.2.1 Tf-Idf . . . 36
5.2.2.2 Word2Vec . . . 36
5.2.2.3 BERT . . . 36
5.2.3 Random Forest . . . 37
5.2.3.1 Tf-Idf . . . 37
5.2.3.2 Word2Vec . . . 37
5.2.3.3 BERT . . . 37
5.2.4 Support Vector Regressor . . . 37
5.2.4.1 Tf-Idf . . . 37
5.2.4.2 Word2Vec . . . 37
5.2.4.3 BERT . . . 38
5.3 Approach 2: Language Model . . . 38
5.4 Evaluation of results . . . 42

6 Discussion **47**

7 Conclusion **50**

References **51**

8 Appendix **58**
8.1 Data . . . 58
8.2 Results . . . 63

List of Figures

3.1 Wordclouds related to entropy scores. 19

4.1 Experimental flow. 29

5.1 Baseline: distribution of errors in test set. 35
5.2 Language model: distribution of errors in test set. 38
5.3 Baseline. 39
5.4 Linear regression lasso (TF-IDF). 39
5.5 Linear regression ridge (TF-IDF) 39
5.6 XGBoost (TF-IDF). 39
5.7 Random forest (TF-IDF). 39
5.8 SVR (TF-IDF). 39
5.9 Distribution of errors of test set of all traditional regressors (TF-IDF). . . . 39
5.10 Baseline. 40
5.11 Linear regression lasso (W2V). 40
5.12 Linear regression ridge (W2V) 40
5.13 XGBoost (W2V). 40
5.14 Random forest (W2V). 40
5.15 SVR (W2V). 40
5.16 Distribution of errors of test set of all traditional regressors (W2V). 40
5.17 Baseline. 41
5.18 Linear regression lasso (BERT). 41
5.19 Linear regression ridge (BERT) 41
5.20 XGBoost (BERT). 41
5.21 Random forest (BERT). 41
5.22 SVR (BERT). 41
5.23 Distribution of errors of test set of all traditional regressors (BERT). 41

5.24 MSE scores of all methods and embedding techniques. 43

8.2 Histograms of entropy scores for AD (left) and Metro (right). 58
8.3 Histograms of entropy scores for NOS (left) and NRC (right). 58
8.1 Histograms of the entropy score, reactions count and word count. 59
8.4 Histograms of entropy scores for NU (left) and Parool (right). 60
8.5 Histograms of entropy scores for RTL (left) and Telegraaf (right). 60
8.6 Histograms of entropy scores for Trouw (left) and Volkskrant (right). 60
8.7 Histograms of entropy scores in linear regression - lasso train set (left) and test set (right). 63
8.8 Histograms of entropy scores in XGBoost train set (left) and test set (right). 63
8.9 Histograms of entropy scores in random forest train set (left) and test set (right). 64
8.10 Histograms of entropy scores in SVR train set (left) and test set (right). . . 64
8.11 Histograms of entropy scores in language model train set (left) and test set (right). 64

List of Tables

3.1 Sample rows from dataset. 16
3.2 Reaction distribution. 16
3.3 Basic information dataset and average entropy score per source. 17
3.4 Hyperparameters per model. 27

5.1 Overview of final results. The results in green are the lowest MSE scores for approach 1. The result in red is the MSE score for approach 2. 45

8.1 Part 1: Examples of some of the text instances from the dataset and their entropy scores. 61
8.2 Part 2: Examples of some of the text instances from the dataset and their entropy scores. 62

1

Introduction

Search engines, social media, news aggregators and the Web in general are praised for being effective sources of information retrieval and have alleviated the public from the pains of gaining knowledge. Nonetheless, in particular over the past years, these systems have been facing harsh criticism for spreading potentially damaging misinformation and have impacted our daily lives to the extent that some say they harm our democracies. The internet and social media have never had such an influence on public opinion and people's decisions as they do now. The AI and systems designed to generate revenue from the public-facing internet have exposed the consequences of algorithm-based social media platforms that prioritize constant user engagement at the expense of considering the intricate impacts on society, politics, and the global community (1). AI-driven content moderation on these platforms have resulted in unprecedented levels of political instability, division, distrust in institutions and polarization of public discourse on controversial topics. (2)

The role of independent media in democratic societies and democratic debate has always been understood as essential. For a functioning democracy, it is crucial for users to be exposed to diverse opinions, discussions and concepts, even to those that they may not agree with or even like (3). Yet the majority of consumer-facing online systems make use of recommender systems, which recommend items to users in their digital environments based on their preferences - and an underlying advertisement business model - but not based on social and ethical metrics. The nature of these recommender systems encourages users to consume information that is in line with their own beliefs, and engage with people and channels that share similar views. Such selective exposure to information has been attributed to the narrowing of political viewpoints and fragmentation of political discourse

in the United States by Garret and Resnick (4) .

This has sparked conversation about topics such as transparancy, diversity and autonomy in the space of AI ethics principles, especially concerning recommender systems. Effective and responsible AI is highly researched in laboratory settings, however it proves difficult to reliably deploy these systems. For example, a product-based study carried out in 2017 suggests to add an 'information nutritional label' for online documents. As stated in the article, "Such a label describes, along a range of agreed-upon dimensions, the contents of the product (an information object, in our case) in order to help the consumer (reader) in deciding about the consumption of the object." The agreed-upon dimensions are amongst others virality, opininion, credibility and controversy (5).

As stated by researchers such as (6) and (7), a crucial factor in encouraging social awareness, supporting civil discourse, and promoting critical literacy is navigating controversial topics on the Web. Research into the automatic detection of controversial topics emerged around 2007, when Kittur et al. (8) designed the first classifier for controversy in Wikipedia articles. Yet the broader AI community seems to have deep dived into controversy detection as an NLP application around 2015, triggered by the wrongdoings during the US presidential elections. It has since seen a rapid uptake by researchers, who have experimented with a wide range of indicators, data sets and methods. The detection of controversial topics is a challenging task, first and foremost because the ambiguity around what constitutes a controversy and/or a controversial topic sets us on shaky ground. Generally speaking, Cambridge Dictionary defines a controversy as 'a disagreement, often a public one, that involves different ideas or opinions about something'.

Existing literature highlights various issues associated with controversy, such as the splitting of communities, biased information, hate speech and violence among groups. (9). This means that modeling and understanding controversies can be useful in many situations and for different stakeholders, for example for journalists, news agencies, government and the general public. Research has demonstrated that being exposed to different viewpoints can bring social advantages in various ways. Firstly, studies have shown that when individuals solely discuss a topic with others who share their opinion (known as an echo chamber), they tend to adopt more extreme and polarized views on the topic. This selective exposure may also impact the political engagement process, causing voters to make decisions earlier and potentially affecting their level of participation over time. Secondly, exposure to diverse

perspectives can increase tolerance towards individuals with differing opinions (4). Recognizing the benefits of being exposed to a wide range of viewpoints, Garrett and Resnick (4) have suggested that technology could be utilized to expand people's perspectives, such as by adjusting the presentation of information to encourage individuals to become more open-minded and deliberate in their thinking (7)

The automatic detection of controversial news posts in the dutch media poses challenges. First of all, we need to factor in subjectivity, when is a news post perceived as controversial and when is it not? Controversies are often implicit and thus not explicitly mentioned in text. Additionally, controversies can cover a wide range of topics with varying vocabulary and can change over time with some topics and actors becoming controversial while others stop being so. This has led to multiple ways of quantifying controversy as well as a range of by researchers proposed proxies. Secondly, there is currently no benchmark data set for the task of controversy detection, and data is used from different platforms and in different languages. Especially when also working with self-formulated indicators of controversy, this complicates comparison between research. To the best of our knowledge, there has only been one research in Dutch, which has not been made publicly available (10). Lastly, researchers have experimented with a variety of methods, some of them platform specific and some more generic. Oftentimes, these approaches lack generalizability. It seems critical to reduce dependency on platform-specific features.

Our research addresses the aforementioned challenges by focusing on a central question: What model(s) prove insightful in predicting the entropy score of Facebook reactions of Dutch news posts? We will be operating under the overarching premise that controversial news can be detected.

Main RQ: What model(s) prove insightful in predicting the entropy score of Facebook reactions of Dutch news posts?

In order to answer the main research question, we investigate whether a Linear Regression, XGBoost, Random Forest, Support Vector Regressor or a Language Model approach provides better insight in predicting the entropy score of Facbook reactions for Dutch news posts. To evaluate the "insightfulness" of a model, we consider the mean squared error as a quantitative measure of its predictive performance. A lower mean squared error suggests higher accuracy and closer predictions to the true entropy scores. By comparing the mean

squared errors of different models, the relative performance can be assessed and the best performing models can be identified. The model approach that obtains the lowest mean squared error proves to be most insightful in predicting the entroy of Facebook reactions for Dutch news posts.

Contributions With the insights gained from our findings, we hope to contribute to existing literature in three key areas:

- This research proposes a variety of content-based topic-agnostic regression models for the task of controversy detection, using a relatively novel approach in designing the controversy score. As opposed to the majority of earlier work, the entropy score used in this research takes into account people's views by utilizing their reactions to posts.
- Moreover, this research presents the first sizeable data set of Dutch news posts gathered on Facebook annotated with a controversy measure, which can be used for the task of controversy detection.
- Lastly, the models proposed are independent of platform-specific features and can thus be generalized to Dutch text in general. In fact, when using specific pre-trained embeddings, the approach can be utilized in different languages.

The remainder of this thesis is structured as follows: Section 2 provides an overview of related work. Section 3 explains the methods used to collect the data and provides an analysis of the data. Furthermore, it discusses the design of the entropy score and the models used in this research. Section 4 outlines the experimental design. Section 5 reports on the experiments and results. Section 6 discusses the results and puts the research into context. It also provides suggestions for future work. Finally, section 7 draws conclusions. Code is made available at https://github.com/hannahvangoor/Dutch-Controversy-Detection.

2

Related Work

This section will give an overview of the empirical literature on controversy detection that has been published over the years, and will provide the necessary context for the remainder of the research. Broadly speaking, research into controversy detection can be explained through the data that is used and the methodology that is applied. Furthermore, knowledge can be obtained from related fields of work, such as sentiment analysis, stance detection and political ideology prediction. However, the section will begin with an outline of what causes polarization and controversy.

2.1 What causes polarization and controversy?

The shaping of people's opinions is influenced by two main psychological factors: confirmation bias and social influence. Firstly, confirmation bias is the tendency for people to accept claims that are in line with one's belief and ignore disagreeing claims. (11). Generally speaking, confirmation bias favors communication with like-minded people. Next to confirmation bias, on an individual level, several other biases impact people's views; cognitive dissonance (12), homophily (13), selective exposure (14) and information overload. Secondly, social influence refers to one's opinion being affected by the people you interact with. On a group level, in-group favoritism and group polarization (15) are also known to play a role in the forming of one's opinion. Baumann et al. (16) state that the isolation following from these above-mentioned aspects, alongside other system-level information filters imposed by for example recommendation systems, are considered to be contributing to the emergence of echo chambers and filter bubbles. Likewise, Vicario et al. (17) suggest that the polarization observed in communities, both offline and online, could be the result of the combined effect of these two factors. The terms polarization and controversy

have oftentimes been treated as synonymous throughout prior research (18). However, a first step towards finding an algorithmic approach that works in a domain- and language-independent manner for solving polarization, is defining and detecting controversial topics in a systematic and fair way. Importantly, we need to gain insight into how controversies emerge on the web - and how they can be quantified.

Controversies are a type of public debate that revolve around issues that divide large segments of society. They often arise from the interaction between core-campaigners and broader sections of the public. As they touch upon deeply rooted ideological divisions or opposing value systems, controversies tend to be persistent and difficult to solve over time. Moreover, the exchange of opinions often extends beyond factual matters and can evoke strong emotions (19). The increasing divergence of opposing views between groups is known to the public as the phenomenon of polarization. Although the term "polarization" is frequently used by both the public as well as in research, it is not a singular concept as it is often assumed to be. The *occurrence* of polarization is regularly discussed in literature, but the specific ways in which the concept is defined or quantified are not clearly distinguished. While some articles do provide a formal measure of polarization, these measures may be specific to the analyzed data set or topic of interest and not fully evaluated or compared to other measures. However, in 'Disambiguation of social polarization concepts and measures' by Bramson et al. (20), nine distinct mathematical concepts of polarization are discussed and are exemplified with formal measures. The most basic notion of polarization is the spread of a distribution. In this sense, the greater the difference between the most extreme views, the more polarized the populations' ideas are. Another way of looking at polarization, as argued by Bramson et al. (20), is distinctness. The degree of polarization in terms of distinctness is determined by the ability to differentiate between groups. The greater the clarity of separation between groups, the more polarized the overall population, irrespective of the distance between groups, their size, or the level of internal agreement within the group.

Interian et al. (18) present an annotated review of the most used network polarization measures and the strategies to handle the issue around polarization. In their work they define network polarization as 'the phenomenon in which the underlying network connecting the members of a society or community is composed of highly connected groups with weak inter-group connectivity', which is a working definition similar to the definition used throughout this research. One of the five approaches outlined in their study is content

qualification methods, where the content published or read by the users is investigated in order to measure its polarity. Each publication determined its own method in how polarity was calculated from the content, yet all of them exploited the user content in the calculation. For example, Bozdag et al. (21) examined the concept of pluralism on Twitter by evaluating information diversity in the Netherlands and Turkey. They described the situation of segregation of the internet as small communities with shared interests, leading to polarization. They utilized, amongst others, the metrics of source diversity, output diversity and input-output correlation, with entropy being a key metric used to measure the diversity of information produced or consumed by a user. Flaxman et al. (22) analyzed the web browsing behaviors of 50,000 internet users in the United States who regularly read online news. Their findings suggest that the usage of social networks and search engines is linked to a rise in the average ideological gap between people. Surprisingly, these very channels also contribute to individuals being exposed to content from the opposite side of their political spectrum. Additionally, most of the online news consumption involves people visiting the homepage of their preferred mainstream news sources. Badami et al. (23) explore polarization within the context of users' interactions with a set of items and its impact on recommender systems. They define polarization based on item ratings and examine its correlation with item reviews.

In light of the detrimental effects of polarization on society, it is imperative to explore ways in which filter bubbles and echo chambers can be avoided. One approach suggested in prior work, is to encourage individuals to engage with opposing viewpoints (24). Liao et al. (25) aimed to mitigate the echo-chamber effect by informing users of other users' opinions on a particular issue, as well as the extremity of their position and their knowledge on the subject. They find that participants who strive to know accurate information on a topic are generally exposed to a broader spectrum of views and tend to align with users who express moderately-mixed attitudes towards that topic. Vydiswaran et al. (26) discovered that users do not actively search for contrasting perspectives on their own. However, when presented with contrasting evidence, they are more likely to develop a comprehensive understanding of the topic. In their research they outline the most effective approaches in which to inform users about controversial topics in a way that may influence opinions. They found that 'showing the credibility of a source, or the expertise of a user, increases the chances of other users believing in the content.' In similar fashion, Munson, Lee and Resnick (27) show that presenting users with information about their own biases motivates them to read articles with opposing views. Based on the research discussed above, we can

observe that various ways of analyzing polarization have been proposed as well as that several studies have focused on decreasing online polarization.

2.2 Prior work on controversy detection

Analysis of controversy has its origins in Wikipedia, exploiting the rich user-generated content base and it's associated metadata such as the discussion page length and the presence of edits and reversions (23). For example, Yasseri et al. (28) investigated the "dynamics of conflict" behind the encyclopedia, focusing on "editorial wars", whereas Rad and Barbosa (29) analyzed mutual reverts and the presence of bi-polarity in the collaboration network to detect controversial articles. Jang (6) investigated probabilistic models for automatic controversy detection and introduced a *controversy language model*, enhancing predictions by leveraging probabilites related to wikipedia controversy features. Most of this early work classified each wikipedia page in isolation. Expanding on this line of research, Dori-Hacohen et al. (30) detected controversy in wikipedia pages using collective classification, taking into account topically neighboring set of pages and hence not classifying pages in isolation anymore. In a follow-up study, Dori-Hacohen et al. (31) generalized and extended their abovementioned approach, aiming to classify controversial webpages by finding Wikipedia pages that discuss the same topic; if that Wikipedia page was deemed controversial, then the webpage was classified as controversial. Given the diverse and ever-evolving nature of controversies, semantic approaches were considered more effective in detecting them. Linmans et al. (32) utilized neural networks, extracting semantic information from texts using weak signals. Exploiting the semantic properties of word embeddings allowed them to significantly improve upon existing controversy detection methods. Their results demonstrated that weak-signal-based neural approaches were more aligned with human estimates of controversy and more resilient to the inherent variability of controversies.

Furthermore, other domains and platforms have been mined for controversial topics, mainly but not limited to social media platforms and news. To begin with, many prior studies have focused on measuring polarization and controversy in social media. Zarate et al. (9) draw from the premise that we can gauge the level of controversy by determining whether one or two main jargons are being used in a discussion. They test their methods on Twitter datasets. In similar manner, Garimella et al. (24) conducted a comprehensive evaluation of a variety of methods for measuring controversy in twitter data as well as evaluated various approaches to constructing graphs. They achieved the highest level of

performance compared to other studies. Additionally, they developed unique metrics for quantifying polarization on Twitter and apply the structure of the endorsement graph to accurately identify whether a set of tweets is controversial or not, independent of context and without requiring any domain expertise. In their thesis, Jang and Allan (33) present controversy as the result of two conflicting viewpoints that form the basis of the debate. They demonstrate that a particular subset of tweets can represent the opposing positions in a polarized discussion. Twitter has become a popular tool for analyzing discussions and polarization, since it serves as a primary platform for public debate in online social media. In similar fashion, researchers have used Reddit data to train controversy detection models. Benslimane et al. (34) combines structural and content information of a post on Reddit as input for their GNNs, exploiting user's interaction graphs. In a study by Hessel and Lee (35), the controversiality of a piece of content is evaluated within the context of the community in which it is shared, essentially stating that controversial topics may be community-specific.

The final main source of data used in controversy detection research is news articles or posts. Choi et al. (36) was among the first researchers to detect controversial topics in online news items, in which he explores the frequency of sentiment words with respect to controversial topics. In a relatively early study by Mejova et al. (37) in 2014, a data-driven methodology was adopted to examine the interplay between controversy, emotional expression, and biased language in news articles. Through this approach, they observed that in the context of controversial issues, the usage of negative affect and biased language is widespread, whereas the expression of intense emotion is tempered. Interestingly, they found that highly emotional terms are less likely to be used in the context of controversial issues, potentially indicating self-moderation on the part of news sources. Furthermore, they identified notable dissimilarities in how controversial topics where treated between different news outlets. Kim and Allan (38) propose a method for identifying controversial topics in a news article which involves an unsupervised training approach essentially producing a disagreement signal within comments on an article and afterwards generating a topic phrase that describes the controversy of the article. Their approach has been shown to be effective through experiments that utilized an expectation-maximization algorithm for training. Zhou et al. (39) tested three semi-supervised learning techniques that reproduce categorizations of political news articles and users as either conservative or liberal, assuming that liberal users will most likely predominantly vote for liberal articles and vice versa for conservative users. The algorithms were initiated with a small number of labeled

articles and users, and subsequently propagated political leaning labels throughout the entire network.

Basile et al. (40) introduced a somewhat straightforward regression model that leveraged the Facebook reactions feature to predict the entropy of a post's reactions. They considered this metric as an approximation to predict the controversy level of news, where a higher entropy value (denoting highly mixed reactions) indicates a greater level of controversy. Furthermore, experiments were performed within and across various communities, specifically focusing on the Facebook pages of individual newspapers. Daphne Groot and Tommaso Caselli (10) roughly followed the previously mentioned research for a poster presentation at the Conference of Computational Linguistics in the Netherlands, focused on the problem of controversy detection in Dutch social media. Groot and Caselli obtained 1859 news posts from 6 Dutch news outlets through the Facebook Graph API. They ran the title, summary and message of each post through a LDA model to predict the topic (considering 100 topics). Then the title, summary, message (all tf-idf vectorized) and predicted topic were input to the first regressor model to predict the reactions volume. The predicted reactions volume served as input to the second regressor to predict the reactions volume per class after which in a third regressor model the entropy score is predicted, reflecting the controversial value. The average entropy score in the dataset equals 0.477. Whereas the results of the topic modeling were not satisfactory, the linear SVR model predicted the entropy scores with 0.272 mean squared error, in the eyes of the researchers a positive result.

Table 2.2 gives a succinct overview of the related work on controversy detection.

Study	Data source	Approach
Kittur et al. (8)	Wikipedia	Edit history of a wikipedia article
Yasseri et al. (28)	Wikipedia	"Editorial wars"
Rad and Barbosa (29)	Wikipedia	"Mutual reverts"
Jang (6)	Wikipedia	"Probabilistic models"
Sznajder et al. (41)	Wikipedia	Textual context
Dori-Hacohen et al. (30)	Wikipedia	Collective classification
Dori-Hacohen et al. (31)	Wikipedia	Webpages in collective classification
Linmans et al. (32)	Wikipedia	Webpages using NN
Zarate et al. (9)	Twitter	Jargon detection
Garimella et al. (24)	Twitter	Endorsement graphs
Jang and Allan (33)	Twitter	Analysis of subset of tweets
Benslimane et al. (34)	Reddit	GNN on users interaction graphs
Hessel and Lee (35)	Reddit	Controversy within communities
Choi et al. (36)	News	Frequency of sentiment words
Mejova et al. (37)	News	Emotional expression and biased language
Kim and Allan (38)	News	Disagreement signal in comments
Zhou et al. (39)	News	Political leaning labels
Basile et al. (40)	News	Regression on Facebook reactions
Groot and Caselli (10)	Dutch News	Regression on Facebook reactions

2.3 Related domains

A great amount of papers investigate controversy within the political domain, often focusing on case studies centered around long-lasting major political events such as presidential elections. Closely related to controversy detection is the prediction of political ideology and political leaning in text. As early as in 2003, Laver et al. (42) introduce in their paper 'Extracting policy positions from political texts using words as data' a methodology on extracting policy positions of political parties in Britain and Ireland as well as providing accompanying uncertainty measures for their estimates. Ever since, research into political ideology detection has gained traction. Yu et al. (43) classified party affiliation from US Congressional speech data, examining generalizability to Senate speeches as well as analyzing their time-dependency. They made use of simple text representations and SVM and naïve Bayes algorithms to train the classifier. Awadallah et al. (44) proposed a system named OpinioNetIt, built to gain insight into the various viewpoints surrounding political controversies, using information on the positions taken by different politicians and stakeholders. The proposed network extended far beyond the then current state-of-the-art in sentiment analysis and opinion mining as it specifically addressed the complexity of political controversies, taking into account the oftentimes nuanced and subtly expressed opinions. Kulkarni et al. (45) suggested a novel approach for political ideology detection of news articles, leveraging not just the textual content but other cues that could be insightful, such as the selection of the title, which is what readers see in snippet views and the presence of hyperlinks in the text. In 2019, the purpose of the International Workshop on Semantic Evaluation was to provide insight into the current state of the art on hyperpartisan news detection. Hyperpartisan news is a type of news that presents an extreme left-wing or right-wing perspective. The effectiveness of automating hyperpartisan news detection remains an open question, but the best team participating in the workshop achieved an accuracy of over 0.8 on a balanced yes/no dataset (46).

Hosseinia et al. (47) examined if the implementation of sentiment and emotion information in pre-trained bidirectional transformers enhances stance detection accuracy in lengthy conversations about contemporary topics. Their experimental findings indicate that a shallow recurrent neural network with sentiment or emotion information can achieve comparable outcomes to fine-tuned BERT, but with 20 times fewer parameters. He et al. (48) deployed a language model that has undergone fine-tuning to identify the partisanship of news articles. By utilizing corpus-contextualized topic embeddings, they were able to

represent the ideology of a news corpus on a specific topic and measure polarization. Jang and Alan (33) aimed with their paper to examine techniques for producing a stance-aware summary that sheds light on a specific controversial topic, by compiling arguments from two opposing perspectives. They worked with Twitter data and approached stance summarization/detection as a task of prioritizing a certain amount of tweets that effectively explained the two conflicting positions of a contentious issue. Stance detection refers to the identification of whether an expressed viewpoint supports or opposes a particular idea. It is closely related to sentiment analysis, however it primarily explores the dual-sided relationship between an opinion and a query. Both sentiment analysis and stance detection are linked to controversy detection and learnings in these fields can be leveraged when building a model or product around controversy detection.

Aker et al. (49) proposed a new human-annoted dataset of news articles with sentiment labels, acting on the observed gap between machines and human judges in determining the sentiment scores of longer texts, such as news articles. According to their study, people may view the entire article as highly sentimental even if only one or two sentences have strong sentiment. Kim and Hovy (50) present a system for detecting opinions at the sentence level, introducing a reliable approach for identifying words that express opinions and those that do not. Subsequently, they detail the process of identifying opinion-bearing sentences using these words.

To conclude, three important insights from the above-described related work and related domains can be taken. Firstly, ambiguity around what constitutes a controversy results in a range of platform-dependent indicators, demonstrating the need for a well-defined indicator and the focus on generalizability. Secondly, utilizing semantic properties of word embeddings can significantly improve controversy detection, especially when using neural approaches. Thirdly, emotion, stance and opinion mining can potentially enhance explainability in controversy detection results.

3

Methodology

The aim of this research is to propose a content-based topic-agnostic method for detecting controversial news posts in Dutch. To demonstrate our method, we consider the content as well as the Facebook reactions associated with Dutch news posts posted on their respective Facebook page. Section 3.1 presents a detailed overview and analysis of the data used in this study. The current state-of-the-art in controversy detection lacks general and flexible approaches, especially in Dutch. The majority of previous studies focus on detecting controversy regarding political issues, often identified in a single carefully-curated dataset and make use of domain-specific knowledge or platform-specific features (24). Section 3.2 provides a comprehensive overview of the design of the controversy indicator used in this study. We employ a diverse range of models to address our research objectives. Section 3.5 elaborates on all the models used in this work.

3.1 Data

The analysis in this research is designed to increase understanding in a system with the ability to detect controversial news posts in Dutch news. Such a system requires as input data both news texts and a proxy for a controversy measure, where design choices come into play as described in section 2. As previously mentioned, there are very few publicly available datasets for controversy detection, in particular in Dutch language. To the best of our knowledge, this is the second data set which contains Dutch news posts and a corresponding controversy score, after the research by Groot and Caselli (10). In order to capture user's opinions and build a proxy upon those, we chose Facebook data. Facebook is the only social media that allows for a range of user reactions instead of just 'like, giving the option for measuring opposing views to capture controversy. Section 3.2 elaborates on

the methodology followed to build the proxy.

On April 12th 2023, 10.000 news posts were scraped from the Facebook pages of the 10 largest and general news outlets in The Netherlands; *NOS, NU.nl, Algemeen Dagblad, RTL Nieuws, de Telegraaf, NRC, Volkskrant, Metro, Trouw, Parool.* Intuitively, what is perceived as controversial is often location-sensitive, therefore only Dutch news outlets were considered. This allows the model to learn how controversies are presented in Dutch news. 1000 news posts per source were scraped, starting from April 12th and going back in time until 1000th post was reached. How far back in time was scraped thus differs per source depending on the posting frequency. The earliest post in the dataset dates back to 2021-11-26. For the Facebook scraping, a facebook page scraper library [1] was modified and implemented. The scraper made use of Selenium and Geckodriver and ran locally on my machine. Selenium is a popular web automation framework that allows interaction with web browsers programmatically. GeckoDriver is the Firefox-specific WebDriver that enables Selenium to control Firefox browsers. With the help of Selenium and GeckoDriver, the scraper navigates to the desired Facebook page, emulates user actions such as scrolling and clicking, and extracts the desired information from the rendered HTML. It simulates a user's browsing behavior to access the content that is typically loaded dynamically or requires user interaction (51) (52). In this way, the Facebook scraper we utilized offers an alternative approach to collect data from Facebook pages by automating the browsing experience and extracting information directly from the web page's HTML structure.

3.1.1 Descriptive analysis of the data

The following data was collected per post: i.) Source name; ii.) number of shares and comments; iii.) date and time posted on; iv.) content of the news post; v.) the full list of users' reactions. For each data point (post), the text of the Facebook post was taken into account as well as the source name along with a breakdown of the reactions (including likes) and its overall entropy calculated based on reaction counts. In order to capture polarity in the entropy measure, the 'likes' and 'loves' were aggregated into one 'like' class. Table 3.1 shows several sample rows from the dataset, illustrating how the entropy measure relates to the user reactions.

[1] https://github.com/shaikhsajid1111/facebook_page_scraper

Table 3.1: Sample rows from dataset.

Id	Content	Like	Wow	Care	Sad	Angry	Haha	Entropy
1.)	'Medewerkers van een..'	8	17	0	0	0	14	1.05
2.)	'Andy en het team..'	18	0	11	0	0	38	0.97
3.)	'Onder de slachtoffers..'	0	4	7	19	0	0	0.90
4.)	'Alleen al de afgelopen..'	25	12	0	0	0	21	1.06
5.)	'De ploeg van bondscoach..'	48	0	0	0	0	0	0.0

Table 3.2 outlines the distribution of reactions per news source. From this table we can infer that some news sources such as NOS and Algemeen Dagblad receive considerably more reactions than other news sources. On the other hand, every news source obtains substantially more likes than other reactions.

Table 3.2: Reaction distribution.

Source	Like	Wow	Cares	Sad	Angry	Haha	Total
NOS	206.237	31.201	35.235	31.431	30.127	74.068	408.299
NU.nl	120.890	9253	15.320	10.610	8.835	32.899	197.807
Algemeen Dagblad	167.785	22.190	33.648	28.916	25.676	44.030	322.245
RTL Nieuws	137.118	23.881	30.695	28.150	21.931	49.888	291.663
de Telegraaf	89.240	13.428	17.738	13.691	13.286	37.967	185.350
NRC	86.857	2078	7890	4285	1334	16.023	118.467
Volkskrant	42.239	1350	6145	4227	2244	8316	64.521
Metro	74.857	6690	9720	5604	7630	26.653	131.154
Trouw	10.476	300	1951	1265	876	3454	18.322
Parool	24.399	941	3118	2649	983	3296	35.386
Total	960.098	111.312	161.460	130.828	112.922	296.594	1.773.214
Percentage	54 %	6 %	9 %	7.5 %	6.5 %	17 %	100 %

Table 3.3 presents an overview of the collected data, showing for each news source and the

total dataset; the average number of reactions, the average number of words in the post, the average number of sentences in the post, minimum entropy, maximum entropy and average entropy. From this information it can be observed that there are rather large disparities in the average number of reactions as well as in the maximum number of entropy. However, the average entropy per news source is relatively close to the total average entropy thus indicating that despite these differences user's seem to react similarly to the news. Figure 8.1 in the appendix depicts a histogram of the entropy score of the dataset, a histogram of the reactions count of the dataset and a histogram of the word count of the posts in the dataset. A kernel density estimate curve is added to the plots, which illustrates a smooth representation of the distribution of the data. As one can see, most of the posts have a word count between 0 and 100 words, signifying that we are dealing with rather short texts. Moreover, the majority of news posts obtained between 0 and 1000 reactions, with a few very large outliers. At the top of Figure 8.1 the histogram of the entropy scores is displayed, in which an organic distribution in the text seems present. Three peaks can be observed, one close to zero, one centered around 0.6 and one centered around 1.1, with a significant drop around 0.7. This distribution is most likely due to the design of the entropy score. It seems important to keep into mind that this could impact the predictions to tend to 0 or 0.7.

Table 3.3: Basic information dataset and average entropy score per source.

Source	avg Reactions	avg Words	avg Sentences	min Entropy	max Entropy	avg Entropy
Total	177	22	3	0.0	1.79	0.6
NOS	408	27	3	0.0	1.01	0.7
NU.nl	197	11	2	0.0	1.10	0.6
Algemeen Dagblad	322	12	2	0.0	1.01	0.7
RTL Nieuws	291	14	2	0.0	1.10	0.7
de Telegraaf	185	16	2	0.0	1.01	0.7
NRC	118	44	4	0.0	1.79	0.5
Volkskrant	64	27	3	0.0	1.79	0.6
Metro	131	17	3	0.0	1.79	0.6
Trouw	18	26	3	0.0	1.79	0.5
Parool	35	26	3	0.0	1.79	0.5

The following lists show the 5 news posts with the highest and lowest entropy scores in the dataset. All the highest-scored news posts have a corresponding entropy measure of 1.79, whereas the lowest-scored posts have a measure of 0. An interesting observation is that the posts from the high entropy scored list come from two sources, Volkskrant and Trouw, whereas the posts from the low entropy scored list come from three different sources, Algemeen Dagblad, de Telegraaf and RTL Nieuws. Figures 8.2, 8.3, 8.4, 8.5 and 8.6 in the appendix show the distributions of entropy score per news source. One can see that some distributions are centered between 0 and 1 whereas others are centered more towards the maximum 1.8. This could lead to some inbalance in the dataset.

List of news posts with highest entropy score:

1. "De Cubanen stemmen zondag voor een nieuw parlement. De keuze is beperkt: op de lijst staan enkel kandidaten die kunnen rekenen op goedkeuring van de staat. Waarom organiseert een socialistisch land met één regerende partij dit toneelstuk?" de Volkskrant (1.79) - Topic: Cuban elections.

2. "Supermarktketen Jumbo overweegt zijn sponsoring in het schaatsen en wielrennen af te bouwen na 2024. Groot alarm voor de sport, of niet?" Trouw (1.79) - Topic: Sponsorship of cycling.

3. "PSV staat voor het tweede jaar op rij in de finale van de KNVB-beker. De Eindhovenaren maakten op een met 6300 toeschouwers volgepakt sportpark De Westmaat een einde aan het indrukwekkende bekeravontuur van de amateurs van Spakenburg." Trouw (1.79) - Topic: Football.

4. "Sahil's keuze om in 'Fight of Flight' de levens van twee (jonge) mensen centraal te stellen en met elkaar te vergelijken, werkt ontzettend goed, schrijft Yasmina Aboutaleb. 'Zo wordt pijnlijk duidelijk wordt hoe bepalend (on)geluk en toeval zijn voor een mensenleven.'" de Volkskrant (1.79) - Topic: Book about refugees.

5. "Wouter Kolff wordt de nieuwe Hubert Bruls, spreekbuis en voorzitter van het gewichtige Veiligheidsberaad." de Volkskrant (1.79) - Topic: New director of security group.

List of news posts with lowest entropy score:

1. "Heel goed nieuws voor de familie Hoekstra" AD.nl (0) - Topic: family.

2. "Nu weet ik dat ik de rest van mijn leven zelfstandig kan blijven" RTL Nieuws (0) - Topic: quoting good news.

3. "Hij mag een kwartier gratis winkelen, waarna hij de boodschappen wil doneren aan de Voedselbank." AD.nl (0) - Topic: free groceries.

4. "Goeie actie? AD.nl" (0) - Topic: Discount.

5. "Vrolijk nieuws uit de dierentuin van Houston. De 90-jarige stralenschildpad Mr. Pickles is voor het eerst vader geworden." De Telegraaf (0) - Topic: birth of turtle.

In the appendix, Tables 8.1 and 8.2 present a random selection of news posts chosen, along with their associated entropy scores, to give some more insight into what texts obtain a certain controversy score. Figure 3.1 shows two word clouds, the left word cloud constructed from the news posts with an entropy score between 0.1 and 0.7 reflecting the second peak and the right word cloud constructed from news posts with an entropy score between 0.7 and 1.1 reflecting the third peak. In these word clouds, the bigger and bolder the word appears, the more often it is observed in the news post. The few largest and boldest words in both clouds are equal, such as 'Nederland' (translated: Netherlands), 'Ik' (translated: me) and 'jaar' (translated: year), but also some differences can be observed. In the right cloud, corresponding to news posts with a higher entropy score, more countries are named such as 'China', 'Russische' and 'Oekraïnse' as well as words such as 'euro' (currency), 'oorlog' (translated: war) and 'corona' (translated: covid), which indeed indicate more controversial topics: Brexit, Russian invasion of Ukraine and the covid pandemic.

Figure 3.1: Wordclouds related to entropy scores.

3.2 Controversy indicator

Following the work by Timmermans et al. (2017) and Basile et al. (2017) (40), I define

Definition 1 *Controversy: a situation where, even after lengthy interactions, opinions of the involved participants tend to remain unchanged and become more and more polarized towards extreme values.*

Intuitively, controversy represents a certain level of disagreement on a topic. This requires particular data to build a proxy indicator in order to measure that level of disagreement. Comparing social media, Facebook posts typically receive more “likes” whereas on Twitter longer comments seem to be more common. However, since February 2016, Facebook users have been able to express specific emotions in response to a post using the reaction feature. This means that a post can now be wordlessly marked with an expression of, for example, “love” or “sad” rather than a generic “like”. This new feature helps Facebook to obtain more information about their users however these reactions are also reasonably safe emotion signals that can be used as proxies. Facebook allows users to not just 'like' a post, but choose from a set of 7 different emotions: angry, like, cares, haha, wow, sad and love. Facebook reactions have been used before as a proxy for controversy annotations (53), allowing a model to be trained for predicting the degree of controversy associated to news. Our hypothesis is based on definition 1, suggesting that if users' reactions fall in two or more emotion classes with high frequencies, the news item is more controversial. To capture polarity in the reactions, 'likes' and 'loves' were added together in one class. We propose that entropy can be used to model news controversy, with higher entropy indicating greater controversy. This is in line with prior research by Basile et al. (40) who took on a similar approach on Italian news Facebook posts.

Entropy is a key concept in Information Theory and in simple terms entropy is just a measure of uncertainty. It stems from the problem coined in the paper 'A Mathematical Theory of Communication' by Claude Shannon in 1948 (54), as the development of information entropy provided a way to gauge the information contained in a message and thus quantified the amount of uncertainty that is eliminated through that message. Along these lines, in the field of information theory, entropy refers to the average amount of "information," "surprise," or "uncertainty" that is inherent in the possible outcomes of a variable (55). Mathematically, entropy is defined as follows:

Definition 2 $E(S) = \Sigma_{i=1}^{c} - p_i * log_2 * p_i$

Entropy is lowest at the 'extremes', when there are either no positive instances or only positive instances. In our case for example, when there is only one reaction present, the uncertainty is 0. Entropy is highest when the data is evenly split between positive and negative instances. This means extreme disorder because there is no majority. The mathematical definition of entropy matches the intuition of the concept, since when users express opposing reactions to a news article, it is indicative that a piece of text may be controversial.

3.3 Text Vectorization

Vectorization is a term used to describe a common technique in machine learning where input data, typically in its raw format such as text, is transformed into numerical vectors. This conversion is necessary since machine learning models operate on numerical data. By representing the input data as vectors of real numbers, it becomes compatible with ML models, enabling them to process and analyze the data effectively. In the context of machine learning, vectorization plays a crucial role in the process of feature extraction (56). There are different techniques to derive information from raw text data, ranging from relatively old and simple methods such as *Bag-of-Words* and *Tf-Idf* to word embeddings (57) such as *Word2Vec* and *Glove* to finally revolutionary language models such as *BERT* and *GPT*. These embeddings act as latent vector representations designed to capture the inherent meaning of words, enabling them to convey semantic connections even when presented in various surface forms (32).

Tf-idf (Term Frequency-Inverse Document Frequency) is a numerical representation technique widely used in natural language processing. By counting both the frequency of a term in a document and its rarity across the entire corpus, tf-idf assigns higher weights to terms that are more informative and discriminative (58). Language models use transfer learning from attention-based transformers (59). Such models can either be finetuned with new data for a specific downstream task or can be used to extract pretrained embeddings which can then serve as input for simple machine learning algorithms or deep learning neural networks. Word2Vec (or other word embeddings) and language models (e.g. BERT) differ in terms of the type of embeddings they provide. While Word2Vec offers static embeddings for individual words, BERT offers contextual embeddings that take into account the surrounding words, resulting in word representations that are influenced by their context (60). We utilized the pre-trained 160-dimensional Word2Vec Dutch combined word

embeddings from the repository available at https://github.com/clips/dutchembeddings. These embeddings were created by combining four large corpora, ensuring a diverse and representative training data allowing the embeddings to capture the nuances and intricacies of the Dutch language. Additionally, using the same architecture and parameters as the transformer-based pre-trained language model BERT, de Vries et al. (61) developed and evaluated a monolingual Dutch BERT model named BERTje. As opposed to the multilingual BERT model that includes Dutch but relies solely on Wikipedia text, BERTje is trained on an extensive and varied dataset consisting of 2.4 billion tokens. The pretrained BERTje embeddings are one of the embedding options used throughout this research because of their outstanding performance.

3.4 Cross-validation

One of the main challenges in machine learning is the uncertainty of how well the model will perform on unseen data, meaning how well the model generalizes to new data and provides an accurate assessment of its performance. To overcome this, a technique called cross-validation can be utilized, which involves splitting the dataset into separate training and test subsets. Such an approach helps to gain insight into how well the model learns. There are various types of cross-validation techniques: k-fold cross-validation, stratified k-fold cross-validation, and leave-one-out cross-validation. Despite their differences, the underlying concept remains the same: iteratively training and testing the model on different subsets of the data to obtain a robust evaluation. We used the most common type of cross validation (CV) in the model building, called k-fold CV. In k-fold CV, the training set of the data is split into k number of subsets, called folds. The model is then repeatedly fit k times, each time training the data on k-1 of the folds and evaluating on the kth fold (called the validation data) (62). Cross-validation is applied twice in the experimental setup, namely for the Grid Search in hyperparameter tuning, where we use 10-Fold CV, and for model evaluation, where we use 5-Fold CV. During the 10 iterations of the 10-fold CV, the model uses different parameter settings to select the best performing parameters (63).

3.5 Models

A wide range of models was selected to study controversy detection in Dutch news. By design, the data is independent and this can be constructed as a regression problem. As

baseline, a dummy regressor which always predicts the mean entropy of the text per source is used. Given the fact that the entropy scores ranges between the values of 0 and 1.8, predicting the mean entropy per source already performs reasonably well with a mean squred error of 0.13, a *variance of errors* of 0.12 and a *standard deviation of errors* of 0.33. Furthermore, next to a language model approach, several traditional machine learning models were implemented; Linear Regression was applied as well as the ensemble methods Random Forest, XGBoost and a Support Vector Regressor.

Linear regression is a simple and straightforward model and it is computationally efficient. It assumes a linear relationship between the input features and the target variable. XGBoost is a powerful ensemble learning algorithm known for its high predictive performance. It combines multiple decision trees to create a robust and accurate model (62). XGBoost is particularly effective in capturing complex nonlinear relationships between the input features and the target variable. Additionally, it incorporates regularization techniques to prevent overfitting and can handle a large number of features. SVR also works well in scenarios where the relationship between the features and the target variable is not necessarily linear. SVR can capture complex patterns in the data and handle high-dimensional feature spaces effectively. It offers the flexibility to incorporate different kernel functions, allowing for nonlinear mapping of the input features. SVR's robustness to outliers and ability to handle large feature sets make it a suitable option for entropy score prediction. Random Forest is also an ensemble learning algorithm that combines multiple decision trees to form a predictive model. Random Forest's versatility, robustness, and ability to capture complex interactions make it a promising choice for entropy score prediction.

For the experimental setup, we chose for increasing model complexity, starting with a baseline and linear regression, moving towards XGBoost, Random Forest and SVR up until a language model approach. By using a diverse set of models; Linear Regression, XGBoost Regressor, Support Vector Regressor, Random Forest Regressor and a language model, we can leverage the strengths of each model to explore performance in predicting controversy of news articles. The remainder of this section will provide a more detailed explanation of each model employed in this study.

3.5.1 Linear Regression

Linear Regression is a widely employed method to understand and quantify the linear association between variables. The line of best fit, which represents the relationship between

the data points, is determined by minimizing the squared distance between the points and the line. This approach is known as minimizing the squared error (64). In linear regression, we estimate the unknown variable (represented by y, the model's output) by calculating a weighted sum of the known variables (represented by x_i, the inputs), and adding a bias term to the sum (65).

$$y = b + \Sigma_{i=1}^{n} x_i * w_i \tag{3.1}$$

In addition to traditional linear regression, there are regularization techniques such as ridge and lasso regularization that can be applied to improve the model's performance. Ridge regularization introduces a penalty term to the squared error objective function, which helps to reduce the impact of multicollinearity among the predictor variables. This penalty term, controlled by a hyperparameter, shrinks the regression coefficients towards zero while still allowing all variables to contribute to the model. On the other hand, Lasso regularization applies a different penalty term that encourages sparsity in the coefficient estimates. It not only reduces multicollinearity but also performs variable selection by forcing some coefficients to exactly zero, effectively removing irrelevant features from the model. By incorporating ridge or lasso regularization techniques into linear regression, we can address the issue of overfitting and improve generalization (66).

3.5.2 Random Forest

The Random Forest Regressor [1] is a technique that can perform regression tasks with the use of multiple decision trees. Decision trees are a type of machine learning algorithm that aim to divide a dataset into smaller subsets for accurate target value prediction. The process involves creating nodes to represent conditions and branches to depict possible outcomes. This splitting procedure persists until no further improvement can be achieved or a predetermined rule is satisfied, such as reaching the maximum depth of the tree (67). CART (Classification and Regression Trees) is a widely used algorithm for decision trees (also used by *scikit-learn*). It constructs a binary tree, meaning each node has two edges, and finds the best feature to split on using an appropriate impurity criterion. For CART least squares (mean squared error) is used. Random forests create a set of decision trees from a randomly selected subset of the training set. It then aggregates the votes from different decision trees to predict the final outcome. By employing a 'majority wins' approach, it mitigates the potential errors that could arise from an individual tree (64). They are named ensemble techniques for this very reason.

[1] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

3.5.3 XGBoost

XGBoost Regressor [1] is a widely used and efficient implementation of the Gradient Boosted Trees algorithm. This supervised learning approach focuses on function approximation by optimizing loss functions and employing various regularization techniques. Its popularity stems from its ability to deliver powerful predictive models with enhanced accuracy and interpretability. However, the success of XGBoost can be attributed to its scalability across various scenarios (68). XGBoost works with the following objective function (loss function and regularization). At iteration **t**, we need to minimize:

$$L^{(t)} = \Sigma_{i=1}^{n} l(y_i, \hat{y_i}^{(}t-1) + f_t(x_i)) + \Omega(f_t) \tag{3.2}$$

Subsequently, in order to enable the use of conventional optimization techniques, the original objective function has to be transformed into a function in the Euclidean domain with Taylor's Theorem. The objective function must be differentiable for this (69). The next step is to build an optimal next learner that achieves maximum possible reduction of loss. In practice, the process of building the learner involves the following steps (known as the "Exact Greedy Algorithm"):

1. Begin with a single root node that includes all the training examples.
2. Iterate over each feature and evaluate all possible splits based on the values of that feature.
3. Calculate the loss reduction for each potential split using the formula: gain = loss(parent node) - (loss(left child node) + loss(right child node)).
4. Only continue growing the branch if the gain for the best split is positive (and greater than the min split gain parameter).

3.5.4 Support Vector Regressor

Support Vector Regression (SVR) [2] is a supervised learning algorithm used for predicting continuous values. It builds upon the same underlying principle as Support Vector Machines (SVMs). SVR aims to find the best-fit line, which is a hyperplane that maximizes the number of points within its margin. Unlike traditional regression models that focus on minimizing the error between the predicted and actual values, SVR aims to find the

[1] https://github.com/dmlc/xgboost
[2] https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html

best line within a specified threshold. This threshold, known as the margin ϵ, represents the distance between the hyperplane and the boundary line (70). The error is quantified by calculating the distance between a point and the boundary line. When the data is not linearly separable, SVR algorithms can identify and determine the appropriate placement of the decision boundary. The objective is to find a decision boundary that effectively separates the data points, which can be done using a kernel function. The kernel trick offers a more efficient and computationally cheaper method to transform data into higher dimensions. Essentially, the kernel is calculating the dot product of two vectors, x and y, in a feature space that may be of very high dimensionality. It allows us to perform computations as if we were working in that high-dimensional space without explicitly transforming the data. To find the linear function that explains the training data, ensuring it is as flat as possible and introducing slack variables, the following objective function - also called primal function - needs to be optimized: (71)

$$J(\beta) = \frac{1}{2}\beta'\beta + C\Sigma_{n=1}^{N}(\xi_n + \xi_n^*) \tag{3.3}$$

subject to several constraints.

3.5.5 Hyperparameter tuning for regressors

Tuning the hyper parameters in a model is very valuable as the performance of the model is highly dependent on these values. Grid search is a powerful tuning technique used to find the optimal values for hyperparameters in a machine learning model. It involves an exhaustive search over a predefined grid of parameter values for the model, hence the name "grid" search. The model, often referred to as an estimator, is evaluated for each combination of hyperparameters in the grid to determine the best configuration. This systematic approach saves time and effort by automating the process of hyperparameter tuning, allowing for an efficient and thorough search for the best parameter values (72). GridSearch uses k-fold cross-validation. The *sklearn* library GridSearchCV is used in this research, which is a python implementation of the grid search process. GridSearchCV requires as input the model that will be used, as well as a list of parameters and the range of values for each parameter of the specified model. Please see table 3.4 for a clarification on the hyper parameters that were tuned per model.

Table 3.4: Hyperparameters per model.

Model	Parameter	Clarification
Random Forest	*Maximum Depth*	The maximum depth of the tree.
	Number of Estimators	The number of trees in the forest.
	Minimum Samples Split	Minimum required number of observations in node to split it (73).
	Minimum Samples Leaf	Minimum number of samples that should be present in the leaf node after splitting a node.
Support Vector Regressor	*C*	Adds a penalty for each misclassified data point (74).
	Epsilon	Defines a margin of tolerance where no penalty is given to errors.
XGBoost Regressor	*Maximum Depth*	The maximum depth of a tree, same as GBM (75).
	Number of Estimators	The number of trees in the forest.
	Learning Rate	It is the step size shrinkage used in update to prevent overfitting.

In the random forest model, the maximum depth was set at 15, the minimum samples leaf at 1, the minimum samples split at 2 and the number of estimators at 150. In the support vector regressor, C was set to 0.1 and epsilon set to 0.4. For the XGBoost regressor, the maximum depth was set at 3, the learning rate at 0.1 and the number of estimators at 200.

3.6 Language Model Approach

BERT (Bidirectional Encoder Representations from Transformers) is a paper published by researchers at Google AI (76), revolutionizing language modelling. One of the fundamental advancements introduced by BERT is its utilization of bidirectional training within the transformer architecture, an attention model, for natural language tasks. Earlier approaches focused on processing a text sequence either in left-to-right manner or through a combination of left-to-right and right-to-left training. BERT demonstrates that a language model trained bidirectionally obtains a more rounded understanding of language context as opposed to models trained in a single direction. The authors presented a technique called Masked LM, where certain words are randomly masked (around 15 % of the input text) after which the model is trained to predict the masked words based on the sur-

rounding context. This mechanism allows the model to effectively capture dependencies and relationships between words in both directions, thereby enhancing its ability to grasp complicated linguistic nuances. BERT can be used for a wide variety of language tasks with only adding a small layer to the model. During the fine-tuning process of BERT, the majority of hyperparameters remain unchanged from the original model. Fine-tuning involves adapting the pre-trained BERT model to the specific downstream task, by training it on task-specific labeled data (77). BERT was trained on similar data as other large language models, namely on BooksCorpus (800M words) and English Wikipedia (2500M words) (76). Due to English input data, the success of BERT was mostly limited to English language. De Vries et al. (61) developed a Dutch BERT model called BERTje, which is architecturally equivalent to the BERT base model with 12 transformer blocks. The pre-training data was similar to the original in terms of size and diversity, consisting of high quality texts of books, news, web news and Wikipedia.

3.7 Evaluation Metrics

This task is treated as a regression problem and the most common way to evaluate regression tasks is mean squared error (MSE). In this case, error represents the difference between the observed values and the predicted values. The difference is squared so that negative and positive values do not cancel each other out. Then the average is taken. In general for a higher error value the performance of the system is considered to be lower (78). The MSE is calculated as follows:

$$MSE = \frac{1}{n}\Sigma(y_{test} - y_{pred})^2 \tag{3.4}$$

Additionally, confidence intervals provide valuable insights into the uncertainty associated with the predictions. By estimating the range within which the true values are likely to fall, confidence intervals offer a measure of the model's reliability. A narrower interval signifies a more precise and confident estimation, while a wider interval implies greater uncertainty in the predictions. A is confidence interval is calculated with the following formula:

$$CI = \hat{x} + / - z(\frac{s}{\sqrt{n}}) \tag{3.5}$$

Lastly, we look at the standard deviation of errors, which captures the dispersion or spread of the errors around the predicted values. A lower standard deviation indicates that the errors are closely clustered around the predicted values, suggesting a more accurate model.

4

Experimental Design

The central question throughout this research 'What model(s) prove insightful in predicting the entropy score of Facebook reactions of Dutch news posts?' is investigated using Facebook data as described in section 3. The data is preprocessed using standard methodologies justified by prior research. Two distinct modeling approaches have been explored and evaluated. The first approach involves text vectorization as a separate step, after which four traditional machine learning models have been employed on the transformed data. The second approach utilizes a language model that combines text vectorization and controversy detection within a single network. Figure 4.1 illustrates the experimental flow in this research. Subsection 4.2 explains the experimental setup of approach 1. Subsection 4.3 describes the experimental setup of approach 2.

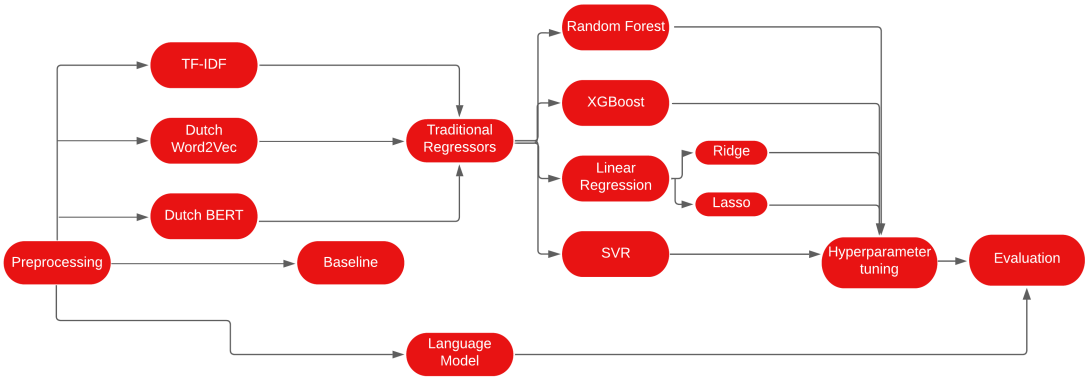

Figure 4.1: Experimental flow.

4.1 Preprocessing

In the preprocessing phase, several steps were taken to ensure the quality and suitability of the dataset for further analysis. Firstly, rows containing missing values in the entropy and content fields were removed from the dataset. This decision was motivated by the necessity to work with complete and reliable data, as it's impossible to fill in textual data, and thus missing values would introduce inaccuracies in subsequent analyses. Additionally, rows with a total number of reactions below the threshold of 30 were removed, aligning again with research from Basile et al. (40) and Groot et al. (10). By doing so, we assume that potential noise and inconsistencies stem from insufficient reaction data and thus this would need to be minimized, allowing for a more robust analysis. Furthermore, rows with an entropy score of 1.3 or higher were excluded from the dataset. This decision was driven by the fact that such values seem to be outliers, which could disproportionately influence the model's learning process. By eliminating these extreme cases, we hope that the model's ability to generalize and capture meaningful patterns from the remaining data is enhanced. The final dataset to be used in the models then contains 6316 rows.
Furthermore, the source name was added to the end of each respective news post to be able to include it in text vectorization. A preprocessing step for text data was included if the text was vectorized using tf-idf. In that case, it applies lowercase conversion, tokenization, stop word removal, stemming, lemmatization, and afterwards joins tokens back into text.

Tokenization is the process of dividing a text into smaller units called tokens. Stemming is a technique that reduces words to their base or root form by removing affixes, which can help reduce the vocabulary size and improve computational efficiency. Lemmatization, on the other hand, goes beyond stemming and analyzes the word's morphological structure to map it to its lemma or dictionary form. It helps preserve the semantic integrity of words, making it beneficial for tasks that require a deeper understanding of the text. By first applying stemming to remove common affixes and then applying lemmatization to handle the remaining variations based on linguistic rules and context, we can achieve a more comprehensive normalization of the text (79).

4.2 Experimental setup approach 1

The experimental setup of approach 1 allows for data loading, preprocessing, embedding generation, baseline analysis and a regression model, providing a comprehensive framework for analyzing and predicting the the entropy score. The chosen embedding option determines the specific embedding technique used in the pipeline.

First of all, the pipeline begins with the CSVReader transformer, which reads a CSV file and loads it into a Pandas DataFrame. The GensimEmbeddings transformer then generates word embeddings using the 160 dimension Dutch Word2Vec model, as described in section 4. It takes a model path and embedding size as inputs, calculates the embeddings for each word in the text, and adds them as new columns to the DataFrame. The TFIDFEmbeddings transformer calculates tf-idf embeddings for the input text using scikit-learn's TfidfVectorizer. Similarly, if tf-idf is the chosen embedding option, the resulting embeddings are added as new columns to the dataframe. Furthermore, the BERTEmbeddings transformer generates BERT embeddings for the text using the "GroNLP/bert-base-dutch-cased" model, also further explained in section 4, and corresponding tokenizer. It tokenizes the text, encodes it, and passes it through the BERT model. The maximum hidden state output is extracted and stored as the embedding for each text. Similarly to the other embedding approaches, BERT embeddings are added as new columns to the dataframe.

The BaselineAnalysis transformer conducts a baseline analysis by performing a dummy regression which always predicts the mean entropy of the text per source. Moreover, it calculates various error metrics for the entropy score and plots a histogram showing the distribution of errors.

The chosen embedding option determines which embedding technique to apply and thus determines the input features. Subsequently, the data is split into training, validation, and test sets. Based on the selected model, the subsequent step is executed within the pipeline:

- The LinearRegressionAnalysis class is designed for conducting linear regression analysis on input features. The option is included to perform lasso or ridge regularization, with alpha as an input variable, set to 1. After the data preparation, the class fits a linear regression model to the training set. 5-Fold cross validation is used to evaluate the model. Furthermore, performance evaluation takes place on the test set, with metrics like mean squared error (MSE).

- The XGBoostTransformer is used for training a XGBoost model on the input features. It fits a XGBoost model on the training set. Hyperparameter tuning is conducted through grid search and cross-validation, enabling the selection of the best model based on the best performance. 5-Fold cross validation is used to evaluate the model. The chosen model is utilized to generate predictions on the test set.

- The SVRTransformer class is utilized to train an SVR model on the input features. The SVR model is fitted to the training set, and hyperparameter optimization is performed using grid search and cross-validation. The best model is selected based on its performance. The model is evaluated using 5-fold cross-validation. Finally, the chosen model is applied to the test set to generate predictions.

- The RFTransformer class is responsible for training a random forest regression model on the input features. Once the data is prepared, the random forest model is fitted to the training set. Grid search and cross-validation are employed to optimize hyperparameters, the best model is selected based on the scoring metric of negative mean squared error. 5-Fold cross validation is used to evaluate the model. Furthermore, performance evaluation occurs on the test set, calculating the mean squared error (MSE).

After making predictions, every model calculates the mean squared error, variance, standard deviation of errors and plots the distribution of errors. In summary, the experimental setup of approach 1 encompasses various stages, including preprocessing, embedding generation, baseline analysis, and the option to select one of four traditional regressors.

4.3 Experimental setup approach 2

The experimental setup of approach 2 allows for training a language model using the BERT architecture. Firstly, the data is loaded and the input and target columns are assigned, subsequently the data is split into train, validation, and test sets. Initially, it uses the *train test split* function from scikit-learn to split the data into train and temporary sets, with a test size of 30%. Then, the temporary set is further split into the validation and test sets, with a test size of 50%. After the data splitting, the BERT tokenizer is loaded using the pre-trained "GroNLP/bert-base-dutch-cased" model. The tokenizer is used to tokenize and encode the text data from the train, validation, and test sets. The *batch encode plus* method is utilized, applying options such as truncation, padding, and a maximum length

of 128 tokens. The tokenized input is returned as TensorFlow tensors and stored in variables. Additionally, to create the input datasets for the model, the tokenized input and target variables together are converted into TensorFlow datasets. The next step involves preparing the model for training.

The code defines the input layers for the BERT model, namely *input ids* and *attention mask*, using *tf.keras.layers.Input*. *Input ids* are used as unique identifiers for the tokens within a sentence, whereas the attention mask is employed to batch the input sequence and indicate which tokens should be attended to by the model during processing. Tokens associated with an attention mask value of 0 are considered irrelevant and will be ignored by the model (80). Then, the BERT model (*TFAutoModel*) is loaded with the pre-trained weights from "GroNLP/bert-base-dutch-cased". The loaded BERT layers are frozen.

In terms of the model architecture, the code retrieves the BERT embeddings by passing the input layers ('input ids' and 'attention mask') to the BERT model. Average pooling is performed on the embeddings, which results in a fixed-size output. Dropout regularization is applied to prevent overfitting, and a dense layer with a ReLU activation function is added. Another dropout layer is included before the final dense layer, which has a linear activation function and serves as the regression output. After defining the model architecture, the model is compiled with an Adam optimizer. The Adam optimizer plays a crucial role in enhancing the accuracy and performance of neural networks by effectively adjusting the learnable parameters of the model. The loss function is set to mean squared error. The model is then trained using the 'fit' method, with the train dataset as input. The training is performed for 10 epochs, using a batch size of 256. The validation dataset is used to evaluate the model's performance during training. In the best performing model, the learning rate was set to $5e-4$, the drop out rate was set to 0.1 and there were 10 epochs. Afterwards, the best model's performance is evaluated on the test set by calculating the mean squared error and analyzing the errors. It then visualizes the distribution of errors through a histogram plot.

5

Results

In this section, the results are presented that have emerged from research into the following central question: What model(s) prove insightful in predicting the entropy score of Facebook reactions of Dutch news posts? A simple baseline and five different modelling approaches have been tried and tested. Concerning the four traditional regressors, results are reported for three distinct vectorization techniques; tf-idf, Dutch word2vec and Dutch BERT. For every approach and vectorization technique the following evaluation metrics are discussed: mean squared error, standard deviation of errors and the (10 %, 90 %) confidence interval of MSE. This means that there is 80 % confidence that the mean squared error falls within those bounds, which we considered reasonable due to the inherent challenges with quantifying controversy. Additionally, the distribution of errors in the test set is shown, to give insights in the variance of errors. Furthermore, figures 8.7,8.8, 8.9, 8.10 and 8.11 in the appendix display the distribution of entropy scores (target variable) in the train set and test set for all the modeling approaches, with word2vec embeddings. This is checked to make sure that there are not any significant imbalances in entropy score between the train and test set. Section 5.4 provides a final overview and analysis of all results.

5.1 Baseline

As mentioned before, the dummy regressor baseline always predicts the mean entropy of the text per source. The baseline obtains a mean squared error of **0.114**, a standard deviation of error of **0.133** and a (10 %, 90 %) confidence interval of **[0.112, 0.116]** when evaluated on the test set. Figure 5.1 shows the distribution of errors in the test set.

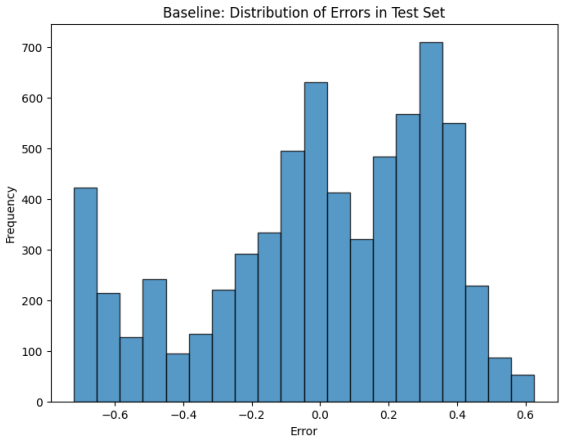

Figure 5.1: Baseline: distribution of errors in test set.

5.2 Approach 1: Traditional ML Regressors

5.2.1 Linear Regression

5.2.1.1 Tf-Idf

Linear Regression with as input tfidf embeddings and lasso regularization ($\alpha = 1$) yields a mean squared error of **0.121**, a standard deviation of errors of **0.341** and a (10 %, 90 %) confidence interval of **[0.112, 0.121]** when evaluated on the test set. Figure 5.9 shows the distribution of errors obtained by linear regression with lasso regularization.

Linear Regression with as input tfidf embeddings and ridge regularization ($\alpha = 1$) yields a mean squared error of **0.109**, a standard deviation of errors of **0.326** and a (10 %, 90 %) confidence interval of **[0.102, 0.111]** when evaluated on the test set. Figure 5.9 shows the distribution of errors obtained by linear regression with ridge regularization.

5.2.1.2 Word2Vec

Linear Regression with as input the 160 dimension Dutch Word2Vec embeddings and lasso regularization ($\alpha = 1$) yields a mean squared error of **0.121**, a standard deviation of errors of **0.341** and a (10 %, 90 %) confidence interval of **[0.112, 0.120]** when evaluated on the test set. Figure 5.16 shows the distribution of errors obtained by this model.

Linear Regression with as input the 160 dimension Dutch Word2Vec embeddings and ridge regularization ($\alpha = 1$) yields a mean squared error of **0.124**, a standard deviation of

errors of **0.347** and a (10 %, 90 %) confidence interval of **[0.116, 0.125]** when evaluated on the test set. Figure 5.16 shows the distribution of errors obtained by this model.

5.2.1.3 BERT

Linear Regression with as input Dutch BERT embeddings and lasso regularization (α = 1) yields a mean squared error of **0.121**, a standard deviation of errors of **0.341** and a (10 %, 90 %) confidence interval of **[0.112, 0.121]** when evaluated on the test set. Figure 5.23 shows the distribution of errors obtained by linear regression with lasso regularization.

Linear Regression with as input Dutch BERT embeddings and ridge regularization (α = 1) yields a mean squared error of **0.115**, a standard deviation of errors of **0.335** and a (10 %, 90 %) confidence interval of **[0.107, 0.117]** when evaluated on the test set. Figure 5.23 shows the distribution of errors obtained by linear regression with ridge regularization.

5.2.2 XGBoost

5.2.2.1 Tf-Idf

XGBoost regression with input vectorized with a tf-idf approach yields a mean squared error of **0.113**, a standard deviation of errors of **0.336** and a (10 %, 90 %) confidence interval of **[0.106, 0.120]** when evaluated on the test set. Figure 5.9 shows the distribution of errors in the test set.

5.2.2.2 Word2Vec

XGBoost regression is performed with the 160-dim Dutch Word2Vec embeddings as input, which yields a MSE of **0.117**, a standard deviation of errors of **0.345** and a a (10 %, 90 %) confidence interval of **[0.111, 0.123]** when evaluated on the test set. Figure 5.16 shows the distribution of errors in the test set.

5.2.2.3 BERT

XGBoost regression with Dutch BERT embeddings obtains a mean squared error of **0.109** and a standard deviation of errors of **0.329** when evaluated on the test set. The (10 %, 90 %) confidence interval equals **[0.102, 0.115]**. Figure 5.23 shows the distribution of errors obtained by the RF regressor.

5.2.3 Random Forest

5.2.3.1 Tf-Idf

A Random Forest Regressor with input vectorized with tf-idf approach yields a mean squared error of **0.113**, a standard deviation of errors of **0.336** and a (10 %, 90 %) confidence interval of **[0.105, 0.120]** when evaluated on the test set. Figure 5.9 shows the distribution of errors in the test set.

5.2.3.2 Word2Vec

A Random Forest Regressor using 160 dimension Dutch Word2Vec embeddings as input obtains a mean squared error of **0.127** and a standard deviation of errors of **0.356** when evaluated on the test set. The (10 %, 90 %) confidence interval equals **[0.120, 0.134]**. Figure 5.16 shows the distribution of errors obtained by the RF regressor.

5.2.3.3 BERT

A Random Forest Regressor with Dutch BERT embeddings obtains a mean squared error of **0.109** and a standard deviation of errors of **0.331** when evaluated on the test set. The (10 %, 90 %) confidence interval equals **[0.103, 0.115]**. Figure 5.23 shows the distribution of errors obtained by the RF regressor.

5.2.4 Support Vector Regressor

5.2.4.1 Tf-Idf

A Support Vector Regressor with input vectorized with tf-idf approach yields a mean squared error of **0.110**, a standard deviation of errors of **0.332** and a (10 %, 90 %) confidence interval of **[0.104, 0.116]** when evaluated on the test set. Figure 5.9 shows the distribution of errors in the test set.

5.2.4.2 Word2Vec

A Support Vector Regressor using 160 dimension Dutch Word2Vec embeddings as input obtains a mean squared error of **0.129** and a standard deviation of errors of **0.360** when evaluated on the test set. The (10 %, 90 %) confidence interval equals **[0.122, 0.137]**. Figure 5.16 shows the distribution of errors obtained by the SVR regressor.

5.2.4.3 BERT

A Support Vector Regressor with Dutch BERT embeddings obtains a mean squared error of **0.120** and a standard deviation of errors of **0.338** when evaluated on the test set. The (10 %, 90 %) confidence interval equals **[0.113, 0.127]**. Figure 5.23 shows the distribution of errors obtained by the SVR regressor.

5.3 Approach 2: Language Model

The language model approach using the Dutch BERT architecture obtains an average mean squared error across all folds of **0.0999** and an average standard deviation of errors across all folds of **0.312** when evaluated on the test set. Figure 5.2 shows the distribution of errors obtained by the language model.

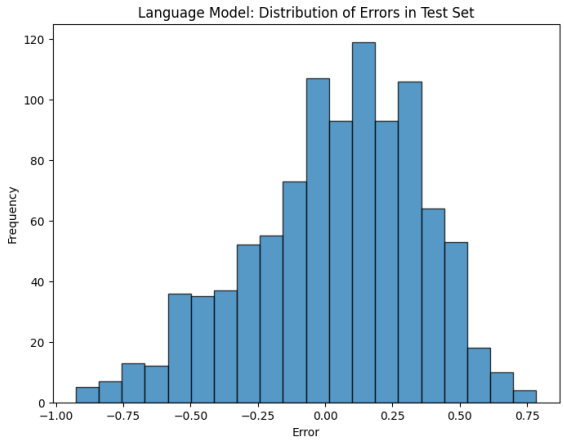

Figure 5.2: Language model: distribution of errors in test set.

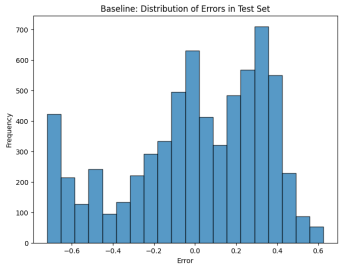

Figure 5.3: Baseline.

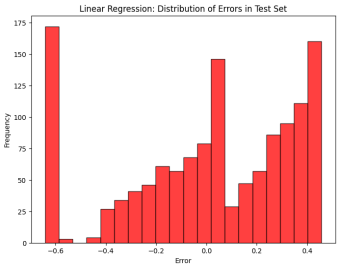

Figure 5.4: Linear regression lasso (TF-IDF).

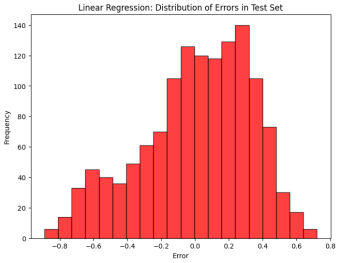

Figure 5.5: Linear regression ridge (TF-IDF)

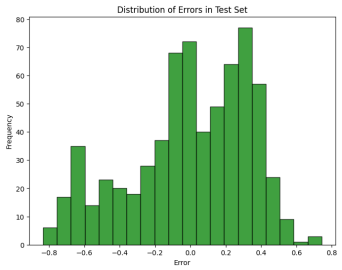

Figure 5.6: XGBoost (TF-IDF).

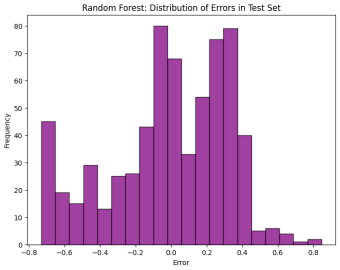

Figure 5.7: Random forest (TF-IDF).

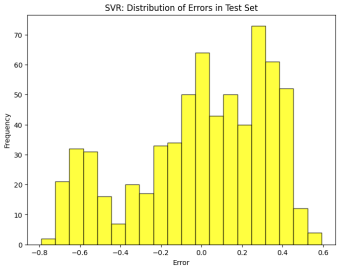

Figure 5.8: SVR (TF-IDF).

Figure 5.9: Distribution of errors of test set of all traditional regressors (TF-IDF).

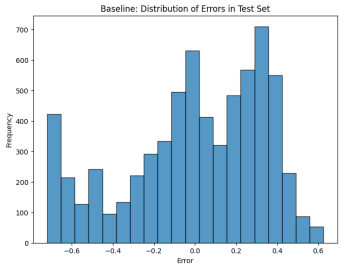

Figure 5.10: Baseline.

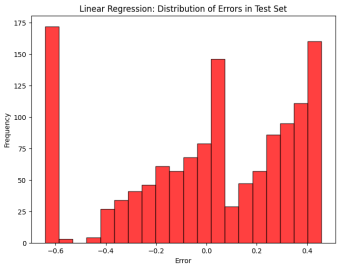

Figure 5.11: Linear regression lasso (W2V).

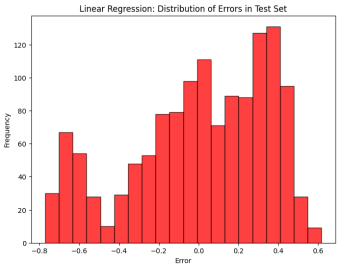

Figure 5.12: Linear regression ridge (W2V)

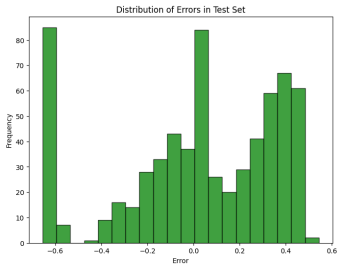

Figure 5.13: XGBoost (W2V).

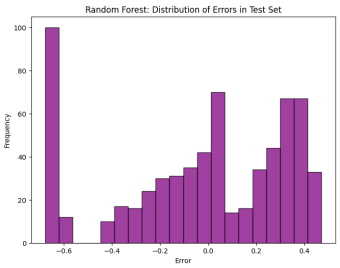

Figure 5.14: Random forest (W2V).

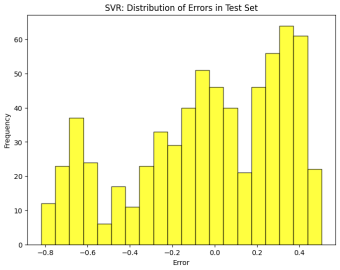

Figure 5.15: SVR (W2V).

Figure 5.16: Distribution of errors of test set of all traditional regressors (W2V).

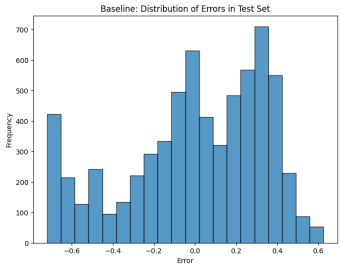

Figure 5.17: Baseline.

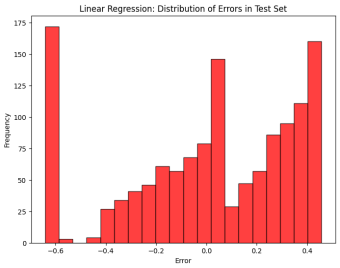

Figure 5.18: Linear regression lasso (BERT).

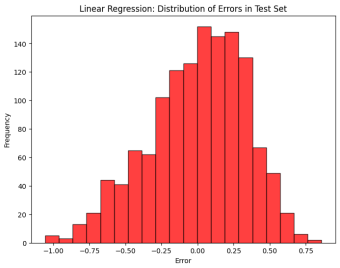

Figure 5.19: Linear regression ridge (BERT)

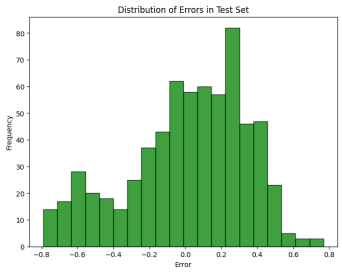

Figure 5.20: XGBoost (BERT).

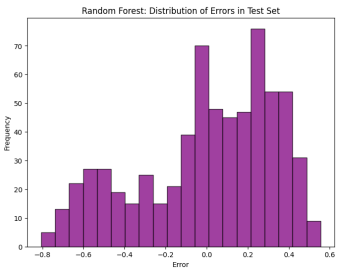

Figure 5.21: Random forest (BERT).

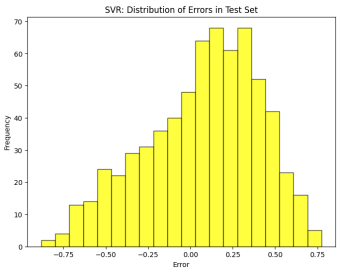

Figure 5.22: SVR (BERT).

Figure 5.23: Distribution of errors of test set of all traditional regressors (BERT).

5.4 Evaluation of results

The mean squared error (MSE) is a statistical measure that quantifies the average squared distance between the true and predicted values. Since it uses squared units rather than the natural data units, the interpretation is less intuitive. We can take the square root of the MSE which equals the average difference between a statistical model's predicted values and the actual values however in natural data units and thus is more interpretable. Root mean squared error is analogous to the standard deviation. In table 5.1 we can see the mean squared error and standard deviation of errors for all models and embedding techniques tested in this research. We observe that all models perform relatively similar, which might be explained by the challenging nature of the problem at hand. Standard deviation of the errors of every model except the baseline roughly range between 0.33 and 0.36, meaning that the typical difference between our model's predictions and the actual entropy score is approximately 0.33. Since the entropy scores range from 0 to 1.3, this means on average an error of 23 %. Furthermore, we can observe that from approach 1, the models that obtain the lowest MSE to be linear regression with ridge regularization and tf-idf vectorized input (**MSE: 0.109**) and XGBoost with BERT embeddings (**MSE: 0.109**). Confidence intervals of the mean squared error are rather tight for all the models, meaning that with 80 % confidence we can say that the mean squared error will fall within that range (and therefore also root mean squared error). Figure **??** shows all the distribution of errors of the traditional regressors in approach 1 (all using BERT embeddings), which emphasizes the difference in error distributions accross the different models. In the appendix, figures 5.9 and 5.16 illustrate the distribution of errors of the traditional regressors in approach 1 with tf-idf and Word2Vec embedding techniques. Approach 2, the language model approach, obtains superior results in terms of mean squared error (**0.0999**) and in standard deviation (**0.312**). This means that our alternative hypothesis of **??** is true, the language model approach exhibits a lower mean squared error compared to the other models investigated in this research, indicating better performance in predicting the entropy of Facebook reactions for Dutch news posts.

That said, we observe that the baseline performs reasonably well compared to the other models, which sparks the debate around simpler and more complex models, especially in terms of computational efficiency, which is explained in more detail in section 6. We have to keep in mind that the baseline is not a content-based method as it relies on platform-specific features (average entropy of source), which could impact deployment 'in the wild'.

Additionally what is interesting to notice is that there is no embedding technique that consistently outperforms across all models. Figure 5.24 plots the MSE scores of all models and different embedding techniques.

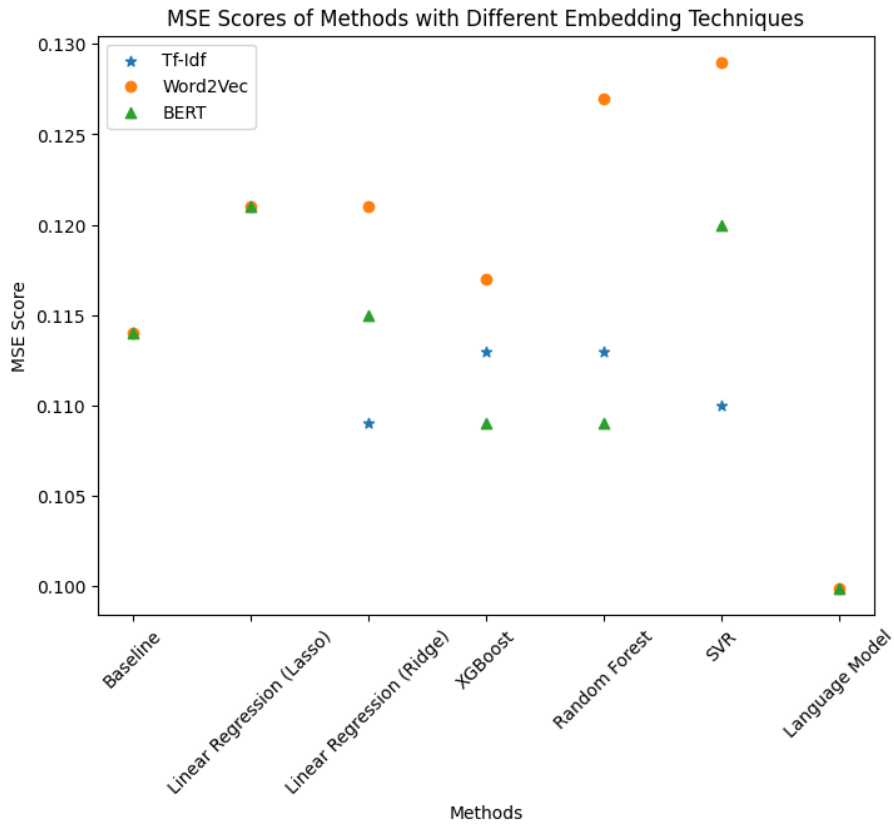

Figure 5.24: MSE scores of all methods and embedding techniques.

When we scrape a new news post from for example NOS, we can demonstrate what it means to predict an entropy score. The following news text was posted on the 6th of June:

"Vanochtend viel een man met een mes kinderen en volwassenen aan op een speelplaats in de Franse stad Annecy. Daarbij zijn zes mensen gewond geraakt, onder wie vier jonge kinderen. Onder de gewonden is een Nederlands kind. Alle vier de kinderen verkeren in levensgevaar. NOS"

Translation: *"This morning, a man attacked children and adults with a knife at a playground in the French city of Annecy. In the attack, six people were injured, including four*

young children. Among the injured is a Dutch child. All four children are in life danger. NOS"

When we use our baseline model, we obtain a predicted entropy score of **0.721** as opposed to a true entropy score of **0.891**, meaning an error of **0.170**. When we use the language model, we obtain a predicted entropy score of **0.561**, meaning an error of **0.330**.

Another example news post from NOS, 6th of June:

"Nederlanders betalen maandelijks gemiddeld een lager voorschotbedrag voor energie dan een half jaar geleden. NOS"

Translation: *"Dutch pay a lower monthly advance for energy on average than six months ago. NOS"*

When we use our baseline model, we again obtain a predicted entropy score of **0.721** (same source as before, so same prediction) as opposed to a true entropy score of **0.685**, meaning an error of **0.036**. When we use the language model, we obtain a predicted entropy score of **0.684**, meaning an error of **0.001**. In its function in a larger system, it can be argued that an overprediction, where the predicted value is higher than the true value, is relatively less tricky than an underprediction, where the predicted value is lower than the true value.

Table 5.1: Overview of final results. The results in green are the lowest MSE scores for approach 1. The result in red is the MSE score for approach 2.

Model	Embedding	Performance metric	Result
Baseline	-	*MSE*	**0.114**
		Std of errors	**0.337**
		CI of MSE	**[0.112, 0.116]**
Linear Regression (lasso)	Tf-Idf	*MSE*	**0.121**
		Std of errors	**0.341**
		CI of MSE	**[0.112, 0.121]**
	Word2Vec	*MSE*	**0.121**
		Std of errors	**0.341**
		CI of MSE	**[0.112, 0.120]**
	BERT	*MSE*	**0.121**
		Std of errors	**0.341**
		CI of MSE	**[0.112, 0.121]**
Linear Regression (ridge)	Tf-Idf	*MSE*	**0.109**
		Std of errors	**0.326**
		CI of MSE	**[0.102, 0.111]**
	Word2Vec	*MSE*	**0.124**
		Std of errors	**0.347**
		CI of MSE	**[0.116, 0.125]**
	BERT	*MSE*	**0.115**
		Std of errors	**0.335**
		CI of MSE	**[0.107, 0.117]**
XGBoost	Tf-Idf	*MSE*	**0.113**
		Std of errors	**0.336**
		CI of MSE	**[0.106, 0.120]**
	Word2Vec	*MSE*	**0.117**
		Std of errors	**0.345**
		CI of MSE	**[0.111, 0.123]**
	BERT	*MSE*	**0.109**
		Std of errors	**0.329**

		CI of MSE	**0.102, 0.115**
Random Forest	Tf-Idf	*MSE*	**0.113**
		Std of errors	**0.336**
		CI of MSE	**[0.105, 0.120]**
	Word2Vec	*MSE*	**0.127**
		Std of errors	**0.356**
		CI of MSE	**[0.120, 0.134]**
	BERT	*MSE*	**0.109**
		Std of errors	**0.331**
		CI of MSE	**[0.103, 0.115]**
SVR	Tf-Idf	*MSE*	**0.110**
		Std of errors	**0.332**
		CI of MSE	**[0.104, 0.116]**
	Word2Vec	*MSE*	**0.129**
		Std of errors	**0.360**
		CI of MSE	**[0.122, 0.137]**
	BERT	*MSE*	**0.120**
		Std of errors	**0.338**
		CI of MSE	**[0.113, 0.127]**
Language Model	BERT	*MSE*	**0.0999**
		Std of errors	**0.312**

6

Discussion

To finalize our research, we would like to offer some considerations in order for the reader to obtain a more comprehensive understanding of our data, our methods, our results and our conclusions. Some factors may be unchangeable, but are worth mentioning. Although we have made all choices with thorough consideration, it is important to offer a critical view on this research, given the challenging nature of the study. When moving beyond the performative towards the operational side of this work, our perspective is grounded in the understanding that its effectiveness and ethical implications are deeply intertwined with the socio-technical ecosystem and the actual environment in which it will operate. Rather than viewing a controversy detector as a cure-all solution, we perceive it as a tool that can assist in achieving significant and positive outcomes for individuals. In our research, we are limited in the sense that "controversy" is a qualitative concept that can at best be reductively translated into narrow quantitative terms. One has to keep in mind that these concepts only make sense in the specific context of their use. It can be stated that all data is essentially biased and it is up to the researchers to make choices from the start onwards on what to include and what not. We are unaware of the meta data of the Facebook posts and reactions and thus unwillingly bias could have been introduced in the data. Furthermore, the vast majority of previously published work uses different data and entropy scores, which limits the ability of readers to directly compare our study with other similar research.

Data A key pillar in this research is the entropy score, computed from Facebook user reactions to news posts. As mentioned above, we are unaware of the meta data, meaning user reactions could unknowingly be biased in terms of gender, age, geography and more. Furthermore, we are highly dependent on the user's own interpretation of the news post.

Since news posts were scraped from 10 different news sources, user reactions can logically vary across communities and geographical areas. In addition, regarding the news posts that serve as input to the models, it is challenging to comment on the quality of the data. Apart from removing missing texts and preproccessing properly, some texts only contain one sentence or are referencing other posts, which could affect predictions. It seems likely that only with an increase in the quality of the data, major exploratory data analysis and curation of the data set, it would be possible to further lower the mean squared error.

Design In this study, we faced countless decisions that had to be made in our research setup, model design as well as in the construction of the entropy metric. Our decisions in this study were informed by a comprehensive review of existing literature. That said, it is important to note that alternative choices could have led to slightly different outcomes. One of the first choices that had to be made was to design this problem as a regression task instead of a classification task. We reasoned that in its function in a bigger system, an absolute numerical prediction would be insightful, especially when the controversy score would serve as input to e.g. a recommender system. Moreover, constructing this as a classification task would mean choosing a threshold, which we figured introduces more bias and potentially errors in the predictions.

Controversy measure One of the most impactful choices in this study, is the choice of controversy measure. While the selected proxy provides a reasonable approximation of controversy indicator (53), it's important to acknowledge that Facebook reactions may not directly originate from the users as explicit cues of controversies. Furthermore, we can argue whether our proposed metric is 'good' in terms of whether it actually captures controversy. The decision to aggregate 'love' and 'like' reactions is one example of steering the metric to capture opposing views, yet this also heavily impacts the design of the entropy score. Referring back to table 3.2, we see that like and love together represent 54 % of all reactions, more than half of the total reactions. The reactions 'sad' and 'angry' are plausibly most opposed to 'like' and 'love', yet together account for just 14 % of total reactions. This means that we deal with a considerable unbalance in the data and it can be questioned whether the entropy score sufficiently captures controversy.

Models The chosen modeling approaches built upon related work and take into account their future work recommendations. Furthermore, it leverages available data and statistical techniques to capture patterns and predict entropy scores. It's crucial to interpret the

results within the context of the models' assumptions and limitations, recognizing that controversy itself may be a complex and multifaceted concept that extends beyond the features and algorithms employed. As in the majority of data science and machine learning research, also in this work the fundamental trade-off exists between simpler and more complex models. Simpler models, such as linear regression, tend to have fewer parameters and less flexibility in capturing complex patterns in the data. On the other hand, more complex models, such as ensemble methods and especially a language model, possess a larger number of parameters and greater capacity to capture intricate relationships in the data. This trade-off has important implications for training times and costs once the models are deployed (81). In this research, all traditional machine learning models take roughly 1/10 of the time to the language model to train, with linear regression training time running even 1/20 of the total training time of the language model. For just a slight increase in performance as demonstrated in section 5, one might consider opting for simpler models in this case instead of the language model, however in the end the choice of the appropriate model complexity depends on the specific requirements of the user.

Future work A natural extension of this study would involve expanding the model to account for perspective bias in different communities, meaning to include more sources and more events (a longer time period). At the reaction-level, future work can focus on exploring and experimenting with clusters of reactions, such as positive, negative, or ambiguous, instead of treating the reactions as distinct indicators, potentially achieving a more nuanced understanding of user responses. Another evident next step would be to expand this study to incorporate other social media data, such as from Twitter or Reddit, in order to obtain a more comprehensive understanding of controversies across various online platforms. Additionally, as innovations in natural language processing and language models are quickly progressing, using these latest advancements for controversy detection could improve results. Lastly, the problem could be designed as a classification task instead of a regression task. That said however, ultimately the desirable next step in our opinion would be to include human feedback into the design. Human annotation of the data for the controversy score could unveil intricacies in what is perceived controversial. A technical approach to controversy detection will grow in importance over time with more usage of social media around the world and due to the prominent place of news recommender systems in our everyday life. The automated detection of controversial news posts will be one of the tools needed to support decision making and opinion forming for tomorrow's leaders.

7

Conclusion

The automatic detection of controversial news has been increasing in importance over the past few years, with the overload of information on the internet influencing people's every day life and decisions. This research marks the pioneering advancement in its field within the Netherlands, setting the stage for further extensive work yet to come. We've build upon learnings from prior work by utilizing semantic properties of word embeddings, predicting controversy with a broad range of regression models and focusing on a well-defined indicator. Furthermore, we've used a substantially larger dataset than has been done in prior research to test our methods, and have set up the ability to generalize our approach. On top of that, our methods obtain better results in predicting controversial news than has ever been achieved with similar approaches.

We present a variety of content-based regression models using an original approach in designing the controversy score. Our models are independent of platform specific features and can be generalized to any Dutch text as input. Furthermore, this research presents the first sizeable data set of Dutch news posts gathered on Facebook annotated with a controversy measure, which can be used for the task of controversy detection in the Netherlands.

The social impact of polarization has been widely felt and has affected democracies and societies around the world. In a world of polarity, it is important to stay balanced, to not always pick a side. Although it is harder to make room for two truths rather than one, connection lives in the space between the two.

References

[1] Courtney Bowman. **The Efficacy and Ethics of AI Must Move Beyond the Performative to the Operational**, 3 2023. 1

[2] Blas Kolic, Fabián Aguirre-López, Sergio Hernández-Williams, and Guillermo Garduño-Hernández. **Quantifying the structure of controversial discussions with unsupervised methods: a look into the Twitter climate change conversation**. 6 2022. 1

[3] Myrthe Reuver, Nicolas Mattis, Marijn Sax, Suzan Verberne, Nava Tintarev, Natali Helberger, Judith Moeller, Sanne Vrijenhoek, Antske Fokkens, and Wouter Van Atteveldt. **Are we human, or are we users? The role of natural language processing in human-centric news recommenders that nudge users to diverse content**, 2021. 1

[4] R Kelly Garrett and Paul Resnick. **Resisting Political Fragmentation on the Internet**, 2011. 2, 3

[5] Norbert Fuhr, Anastasia Giachanou, Gregory Grefenstette, Iryna Gurevych, Andreas Hanselowski, Kalervo Jarvelin, Rosie Jones, YiquN Liu, Josiane Mothe, Wolfgang Nejdl, Isabella Peters, and Benno Stein. **An Information Nutritional Label for Online Documents**. *ACM SIGIR Forum*, **51**:46–66, 2 2018. 2

[6] Myungha Jang, John Foley, Shiri Dori-Hacohen, and James Allan. **Probabilistic approaches to controversy detection**. **24-28-October-2016**, pages 2069–2072. Association for Computing Machinery, 10 2016. 2, 8, 11

[7] Elad Yom-Tov, Susan Dumais, and Qi Guo. **Promoting Civil Discourse Through Search Engine Diversity**. *Social Science Computer Review*, **32**:145–154, 2014. 2, 3

[8] Mary Beth. Rosson, ACM Digital Library., and ACM Special Interest Group on Computer-Human Interaction. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* ACM, 2007. 2, 11

[9] Juan Manuel Ortiz de Zarate and Esteban Feuerstein. **Vocabulary-based Method for Quantifying Controversy in Social Media**. 1 2020. 2, 8, 11

[10] Daphne Groot and Tommaso Caselli. **Controversy Detection in Dutch Social Media**. 1 2019. 3, 10, 11, 14, 30

[11] Longzhao Liu, Xin Wang, Xuyang Chen, Shaoting Tang, and Zhiming Zheng. **Modeling Confirmation Bias and Peer Pressure in Opinion Dynamics**. *Frontiers in Physics*, **9**, 3 2021. 5

[12] R. Kelly Garrett. **Echo chambers online?: Politically motivated selective exposure among Internet news users**. *Journal of Computer-Mediated Communication*, **14**:265–285, 2009. 5

[13] Garimella. **Garimella Polarization on Social Media Thesis**. 5

[14] Natalie Jomini Stroud. **Media use and political predispositions: Revisiting the concept of selective exposure**. *Political Behavior*, **30**:341–366, 9 2008. 5

[15] Cass R Sunstein. **Chicago Unbound Chicago Unbound The Law of Group Polarization The Law of Group Polarization**. 5

[16] Fabian Baumann, Philipp Lorenz-Spreen, Igor M. Sokolov, and Michele Starnini. **Modeling echo chambers and polarization dynamics in social networks**. 6 2019. 5

[17] Michela Del Vicario, Antonio Scala, Guido Caldarelli, H. Eugene Stanley, and Walter Quattrociocchi. **Modeling confirmation bias and polarization**. *Scientific Reports*, **7**, 1 2017. 5

[18] Ruben Interian, Ruslán G. Marzo, Isela Mendoza, and Celso C. Ribeiro. **Network polarization, filter bubbles, and echo chambers: an annotated review of measures and reduction methods**. *International Transactions in Operational Research*, 2022. 6

[19] Kaspar Beelen, Evangelos Kanoulas, and Bob Van De Velde. **Detecting controversies in online news media**. pages 1069–1072. Association for Computing Machinery, Inc, 8 2017. 6

[20] Aaron Bramsona, Patrick Grim, Daniel J. Singer, Steven Fisher, William Berger, Graham Sack, and Carissa Flocken. **Disambiguation of social polarization concepts and measures**. *Journal of Mathematical Sociology*, **40**:80–111, 2016. 6

[21] Engin Bozdag, Qi Gao, Geert-Jan Houben, and Martijn Warnier. **Does Offline Political Segregation Affect the Filter Bubble? An Empirical Analysis of Information Diversity for Dutch and Turkish Twitter Users**. 6 2014. 7

[22] Seth Flaxman, Sharad Goel, and Justin M. Rao. **Filter bubbles, echo chambers, and online news consumption**. *Public Opinion Quarterly*, **80**:298–320, 2016. 7

[23] Mahsa Badami, Institute of Electrical, Electronics Engineers, and IEEE Computer Society. *Detecting Polarization in Ratings: An Automated Pipeline and a PreliminaryQuantification on Several Benchmark Data Sets*. 7, 8

[24] Kiran Garimella, Gianmarco De Francisci Morales, Aristides Gionis, and Michael Mathioudakis. **Quantifying Controversy on Social Media**. *ACM Transactions on Social Computing*, **1**:1–27, 3 2018. 7, 8, 11, 14

[25] Qingzi Liao Dissertation. **DESIGNING TECHNOLOGIES FOR EXPOSURE TO DIVERSE OPINIONS**. 7

[26] Vinod Vydiswaran, ChengXiang Zhai, Dan Roth, and Peter Pirolli. *BiasTrust: Teaching Biased Users About ControversialTopics*. 7

[27] Sean A Munson, Stephanie Y Lee, and Paul Resnick. **Encouraging Reading of Diverse Political Viewpoints with a Browser Widget**, 2013. 7

[28] Taha Yasseri, Robert Sumi, András Rung, András Kornai, and János Kertész. **Dynamics of conflicts in wikipedia**. *PLoS ONE*, **7**, 6 2012. 8, 11

[29] Hoda Sepehri-Rad and Denilson Barbosa. **Identifying Controversial Wikipedia Articles using EditorCollaboration Networks**. 2017. 8, 11

[30] Shiri Dori-Hacohen, David Jensen, and James Allan. **Controversy detection in wikipedia using collective classification**. pages 797–800. Association for Computing Machinery, Inc, 7 2016. 8, 11

[31] Dori-Hacohen Shiri and James Allan. **Automated Controversy Detection on the Web**. 8, 11

[32] Jasper Linmans, Bob Van De Velde, and Evangelos Kanoulas. **Improved and robust controversy detection in general web pages using semantic approaches under large scale conditions**. pages 1647–1650. Association for Computing Machinery, 10 2018. 8, 11, 21

[33] Myungha Jang and James Allan. **Explaining controversy on social media via stance summarization**. pages 1221–1224. Association for Computing Machinery, Inc, 6 2018. 9, 11, 13

[34] Samy Benslimane, Jérôme Azé, Sandra Bringay, Maximilien Servajean, and Caroline Mollevi. **World Wide Web, 2023, Special Issue on Web Information Systems Engineering**. **26**, 2021. 9, 11

[35] Jack Hessel and Lillian Lee. **Something's Brewing! Early Prediction of Controversy-causing Posts from Discussion Features**. 4 2019. 9, 11

[36] Yoonjung Choi, Yuchul Jung, and Sung-Hyon Myaeng. **Identifying Controversial Issues and Their Sub-topics in News Articles**, 2010. 9, 11

[37] Yelena Mejova, Amy X. Zhang, Nicholas Diakopoulos, and Carlos Castillo. **Controversy and Sentiment in Online News**. 9 2014. 9, 11

[38] Youngwoo Kim and James Allan. **Unsupervised Explainable Controversy Detection from Online News**. 9, 11

[39] Daniel Xiaodan Zhou, Paul Resnick, and Qiaozhu Mei. **Classifying the Political Leaning of News Articles and Users from User Votes**, 2011. 9, 11

[40] Angelo Basile, Tommaso Caselli, and Malvina Nissim. **Predicting controversial news using facebook reactions**. **2006**. CEUR-WS, 2017. 10, 11, 20, 30

[41] Benjamin Sznajder, Ariel Gera, Yonatan Bilu, Dafna Sheinwald, Ella Rabinovich, Ranit Aharonov, David Konopnicki, and Noam Slonim. **Controversy in Context**. 8 2019. 11

[42] Michael Laver, Kenneth Benoit, and John Garry. **Extracting Policy Positions from Political Texts Using Words as Data**, 2003. 12

[43] Bei Yu, Stefan Kaufmann, and Daniel Diermeier. **Classifying party affiliation from political speech**. *Journal of Information Technology and Politics*, **5**:33–48, 2008. 12

[44] Rawia Awadallah, Maya Ramanath, and Gerhard Weikum. **Harmony and dissonance: Organizing the people's voices on political controversies**. pages 523–532, 2012. 12

[45] Vivek Kulkarni, Junting Ye, Steven Skiena, and William Yang Wang. **Multi-view Models for Political Ideology Detection of News Articles**. 9 2018. 12

[46] Johannes Kiesel, Maria Mestre, Rishabh Shukla, Emmanuel Vincent, Payam Adineh, David Corney, Benno Stein, and Martin Potthast. **SemEval-2019 Task 4: Hyperpartisan News Detection**. pages 829–839, 2019. 12

[47] Marjan Hosseinia, Eduard Dragut, and Arjun Mukherjee. **Stance Prediction for Contemporary Issues: Data and Experiments**. 5 2020. 12

[48] Zihao He, Negar Mokhberian, Antonio Camara, Andres Abeliuk, and Kristina Lerman. **Detecting Polarized Topics Using Partisanship-aware Contextualized Topic Embeddings**. 4 2021. 12

[49] Ahmed Aker, Sakib Haque, Zachary Eberhart, Aakash Bansal, and Collin McMillan. **Corpus of news articles annotated with article-level sentiment**. **2022-March**, pages 36–47. IEEE Computer Society, 2022. 13

[50] Soo-Min Kim and Eduard Hovy. **Automatic Detection of Opinion Bearing Words and Sentences**. 13

[51] Sophia Z. **Collecting Public Data from Facebook Using Selenium and Beautiful Soup**, 10 2021. 15

[52] Data Scraping Services and Data Extraction. **How to Use Web Scraping with Selenium and BeautifulSoup for Dynamic Pages?**, 2 2022. 15

[53] Chris Pool and Malvina Nissim. **Distant supervision for emotion detection using Facebook reactions**. 11 2016. 20, 48

[54] C E Shannon. **A Mathematical Theory of Communication**, 1948. 20

[55] Sam T. **Entropy: How Decision Trees Make Decisions**, 1 2019. 20

[56] Abhishek Jha. **Vectorization Techniques in NLP [Guide]**, 4 2023. 21

[57] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. **Efficient Estimation of Word Representations in Vector Space**. 1 2013. 21

[58] Rohit Madan. **TF-IDF/Term Frequency Technique: Easiest explanation for Text classification in NLP using Python**, 5 2019. 21

[59] Ashish Vaswani, Google Brain, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. **Attention Is All You Need**. 21

[60] Mauro Di Pietro Towards Data Science Mauro Di Pietro. **Text Classification with NLP: Tf-Idf vs Word2Vec vs BERT**, 7 2020. 21

[61] Wietse de Vries, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim. **BERTje: A Dutch BERT Model**. 12 2019. 22, 28

[62] Will Koehrsen. **Hyperparameter Tuning the Random Forest in Python**, 1 2018. 22, 23

[63] Rahil Shaikh. **Cross Validation Explained: Evaluating estimator performance.**, 11 2018. 22

[64] Terence Shin. **All Machine Learning Models Explained in 6 Minutes**, 1 2020. 24

[65] Dorian Lazar. **Understanding Linear Regression**, 8 2020. 24

[66] Saptashwa Bhattacharyya. **Ridge and Lasso Regression: L1 and L2 Regularization**, 9 2018. 24

[67] Stacay Ronaghan. **The Mathematics of Decision Trees, Random Forest and Feature Importance in Scikit-learn and Spark**, 5 2018. 24

[68] Tianqi Chen and Carlos Guestrin. **XGBoost: A Scalable Tree Boosting System**. 3 2016. 25

[69] Dimitris Leventis. **XGBoost Mathematics Explained**, 11 2018. 25

[70] Ashwin Raj. **Unlocking the True Power of Support Vector Regression**, 10 2020. 26

[71] Vladimir N Vapnik. **An Overview of Statistical Learning Theory**, 1999. 26

[72] Farhad Malik. **What Is Grid Search?**, 2 2020. 26

[73] Ankit Chauhan. **Random Forest Classifier and its Hyperparameters**, 2 2021. 27

[74] Soner Yildirim. **Hyperparameter Tuning for Support Vector Machines — C and Gamma Parameters**, 5 2020. 27

[75] Aman Gupta. **XGBoost Hyperparameters — Explained**, 4 2021. 27

[76] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. **BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding**. 10 2018. 27, 28

[77] Rani Horev. **BERT Explained: State of the art language model for NLP**, 10 2018. 28

[78] XuanKhanh Nguyen. **Understanding the Mean Squared Error**, 6 2020. 28

[79] Hunter Heidenreich. **Stemming? Lemmatization? What?**, 12 2018. 30

[80] Natasha Sharma. **Hugging Face Pre-trained Models: Find the Best One for Your Task**, 4 2023. 33

[81] Thomas Dorfer. **Why Simple Models Are Often Better**, 1 2023. 49

8

Appendix

8.1 Data

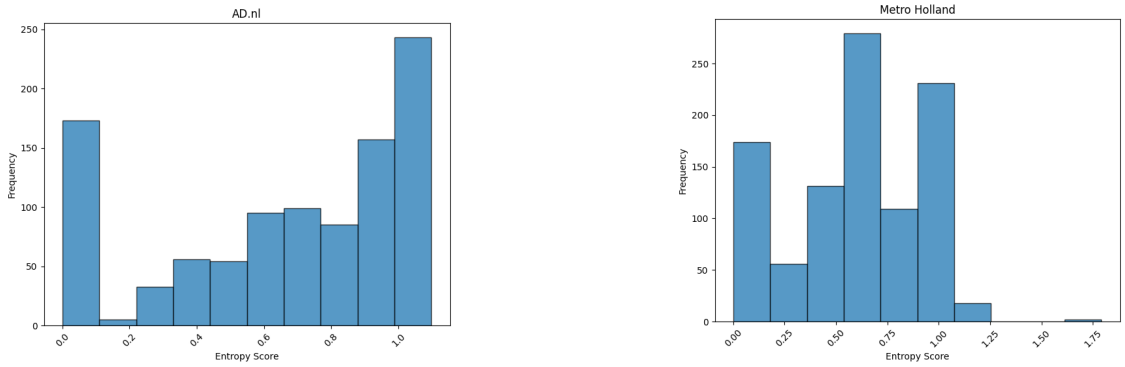

Figure 8.2: Histograms of entropy scores for AD (left) and Metro (right).

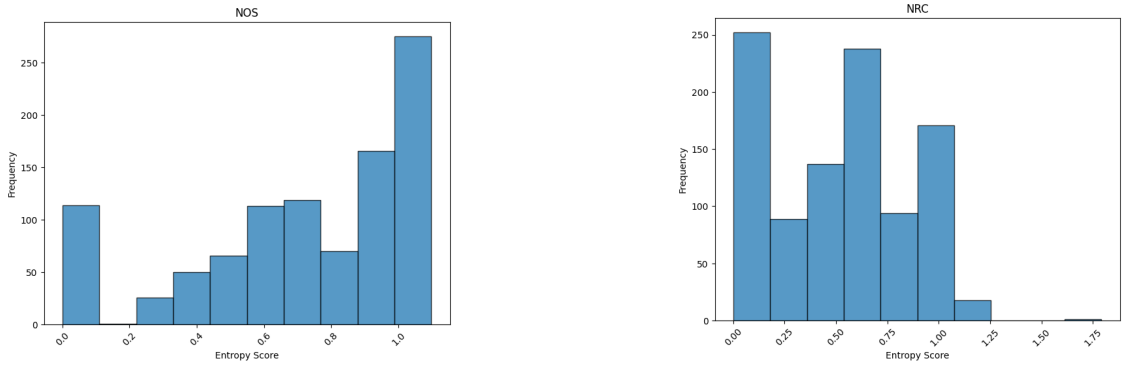

Figure 8.3: Histograms of entropy scores for NOS (left) and NRC (right).

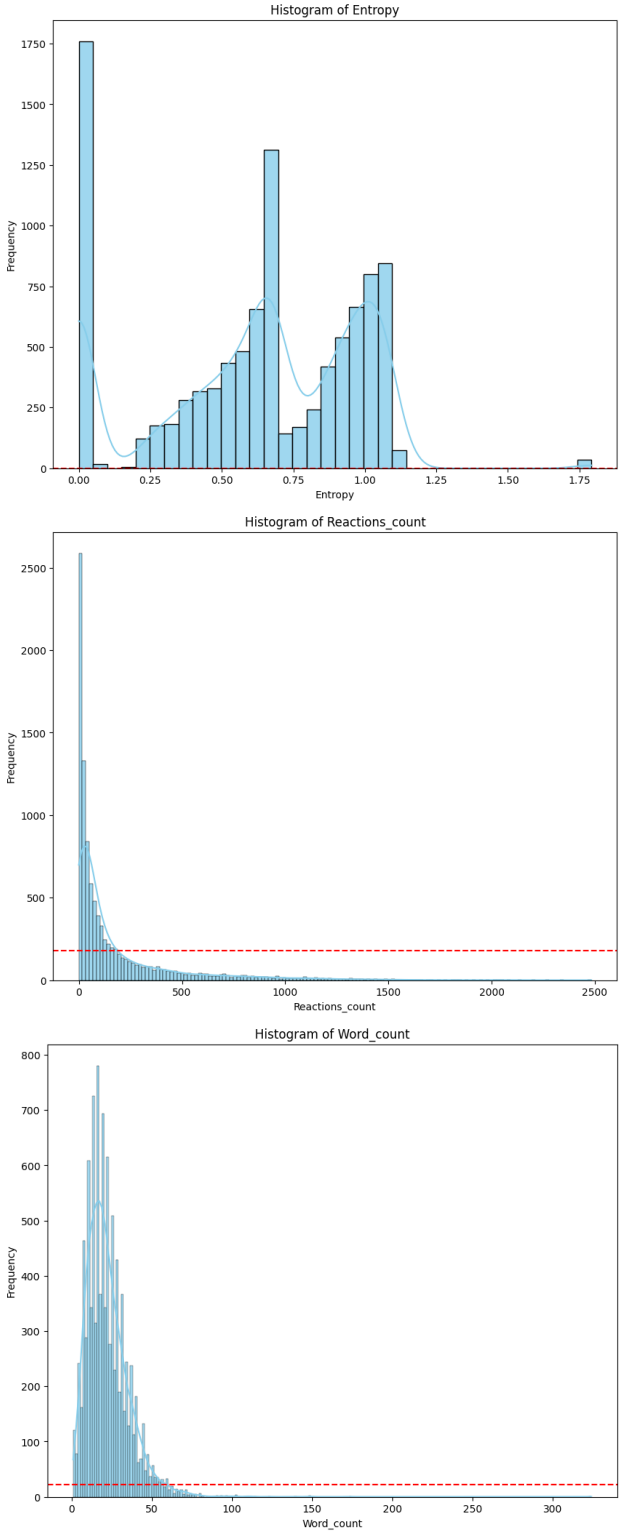

Figure 8.1: Histograms of the entropy score, reactions count and word count.

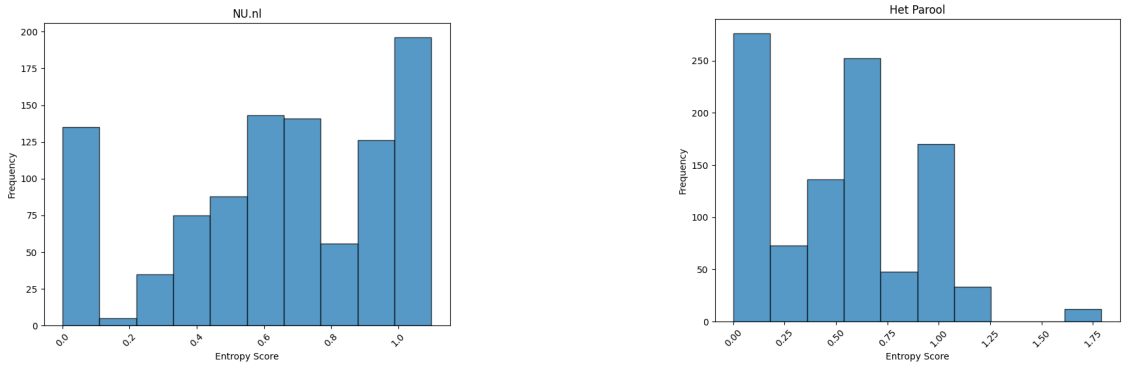

Figure 8.4: Histograms of entropy scores for NU (left) and Parool (right).

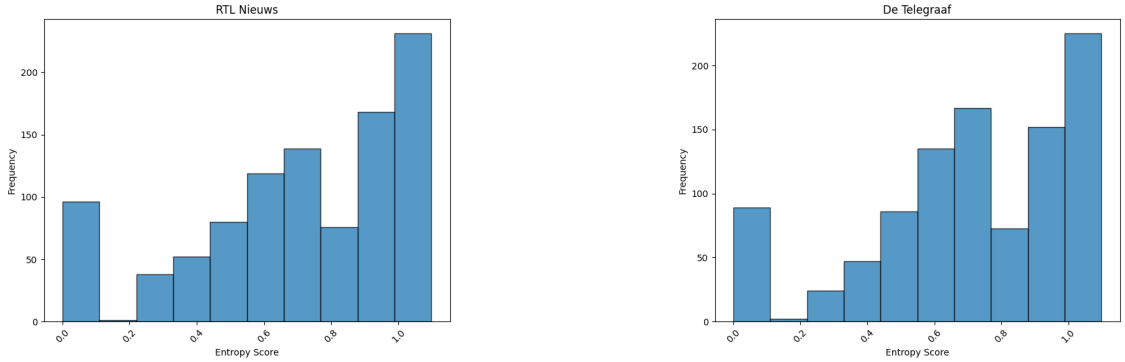

Figure 8.5: Histograms of entropy scores for RTL (left) and Telegraaf (right).

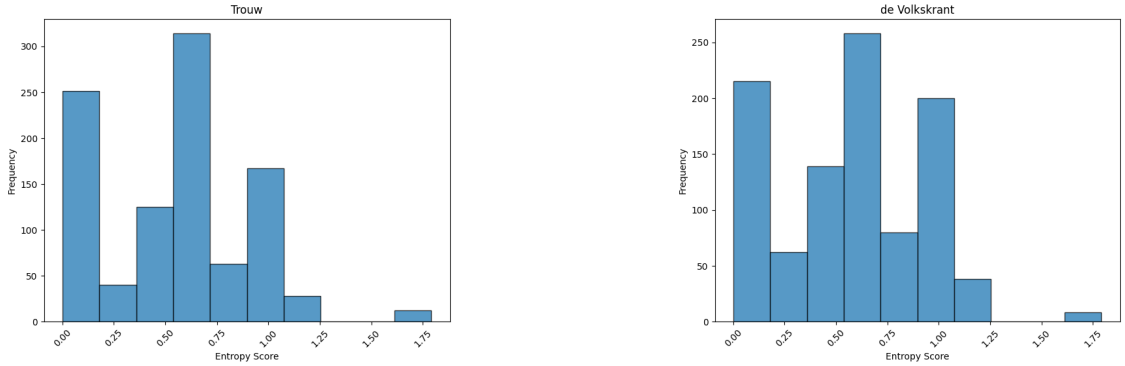

Figure 8.6: Histograms of entropy scores for Trouw (left) and Volkskrant (right).

Table 8.1: Part 1: Examples of some of the text instances from the dataset and their entropy scores.

Content	Entropy score
Morgen gaat Finland naar de stembus. Premier Marin gaat samen met de radicaal-rechtse partij De Finnen en de liberaal-conservatieve Coalitiepartij gelijk op in de peilingen. NOS	0.622
Nee! Heel snel bingen dan maar! AD.nl	1.081
Alsnog zitten geregeld mensen met stadionverbod toch in het stadion. RTL Nieuws	0.728
Is er bij Expeditie Robinson net als in het voetbal een VAR nodig? AD.nl	0.899
'We zijn gezegend dat we hier mogen wonen. Zo voelt het heel erg' AD.nl	0.437
'We zijn gezegend dat we hier mogen wonen. Zo voelt het heel erg' AD.nl	0.437
Hoe raakt een autohandelaar uit het oosten des lands betrokken bij een fraudezaak waarbij 830.000 euro werd ontvreemd? RTL Nieuws	1.016
Het positief geteste echtpaar dat gisteren werd aangehouden omdat het was vertrokken uit het quarantainehotel in Badhoevedorp, verbleef daar vrijwillig. Ze moesten wel in quarantaine, maar niet noodzakelijkerwijs in dat hotel. NOS	1.009
Oud-bokser Mike Tyson (56) heeft zijn naam verbonden aan een coffeeshop in de Amsterdamse Spuistraat. Daarbij moet het niet blijven. Er zijn plannen voor meer Tyson-coffeeshops in Amsterdam, aldus een woordvoerder. Het Parool	0.379
Ze moeten eerst een training op locatie volgen en de eerst beschikbare plek is niet eerder dan 20 december. AD.nl	0.955
In een video is ze vlak na de operatie te zien en houdt ze de implantaten vast. AD.nl	0.352
Een van de drie zaken waarvoor rapper Ali B wordt vervolgd, draait om zangeres Ellen ten Damme. Zij deed geen aangifte, maar maakte wel melding van seksueel ongewenst gedrag van Ali B. Het Parool	0.689
Eigenlijk zijn ze verpleegkundige, student en natuurkundige; nu zijn ze een verloren gewaande elfenprins, een kunstzinnige sater en een opvliegerige oude dwerg. Nieuwe, jongere doelgroepen ontdekken het „klassieke nerdspel" Dungeons & Dragons. Het is nu écht niet meer alleen voor een nerdy niche. NRC	0
Actievoerders van Farmers Defence Force en Samen voor Nederland willen vanmiddag in het Zuiderpark demonstreren tegen het stikstofbeleid, maar ook tegen de trage afhandeling van de toeslagaffaire en het schadeherstel in Groningen. NOS	0.633

Table 8.2: Part 2: Examples of some of the text instances from the dataset and their entropy scores.

Content	Entropy score
Het uiterlijk van Jutta Leerdam is haar visitekaartje, maar we mogen het er als nette mensen niet over hebben. We horen haar prestaties te bezingen. Terwijl het lijf van de schaatser juist zo prachtig is, schrijft columnist Marijn de Vries. „Ze mág er trots op zijn, en wat mij betreft ook méér dan het gepaste, ingehouden trotse dat wij van sporters verwachten. Pronk ermee Jutta, go girl." NRC	0
Laat jij een vrachtwagen ook wel eens je remlichten zien? AD.nl	0.980
Nederland heeft volgens mensenrechtenorganisaties bijgedragen aan het leed van vluchtelingen op de Griekse eilanden. De organisaties stellen de staat verantwoordelijk. Trouw	0.980
Laat jij een vrachtwagen ook wel eens je remlichten zien? AD.nl	0.693
Brrr waar blijft de lente? De Telegraaf	0.876
Eindelijk, het R-getal is weer onder de 1! AD.nl	0.621
We weten het allemaal: vliegen is niet goed voor het klimaat. Maar met welke redenen stappen we nog wel in het vliegtuig? NRC	0.920
Regeringspartijen D66, VVD en CDA staken miljoenen in hun verkiezingscampagnes. Grote winnaar BBB gaf slechts 28 duizend euro uit aan advertenties. de Volkskrant	0.780
Dit weekend gebeurde het weer: een agressieve passagier zorgde tijdens een vlucht voor onrust. Wat kan het personeel daartegen doen? RTL Nieuws	0.750
En of ze gelijk hebben! NU.nl	0
Na maanden van stakingen gaan de werkgevers en vakbonden in het streekvervoer weer met elkaar onderhandelen over een nieuwe cao. Daarmee zijn ook de aangekondigde stakingen van de baan, die volgende week op maandag, woensdag en vrijdag zouden zijn. NOS	0.674
Wie wil er een rondje wandelen met Morris? AD.nl	1.023
Afschuwelijk! AD.nl	0.032

8.2 Results

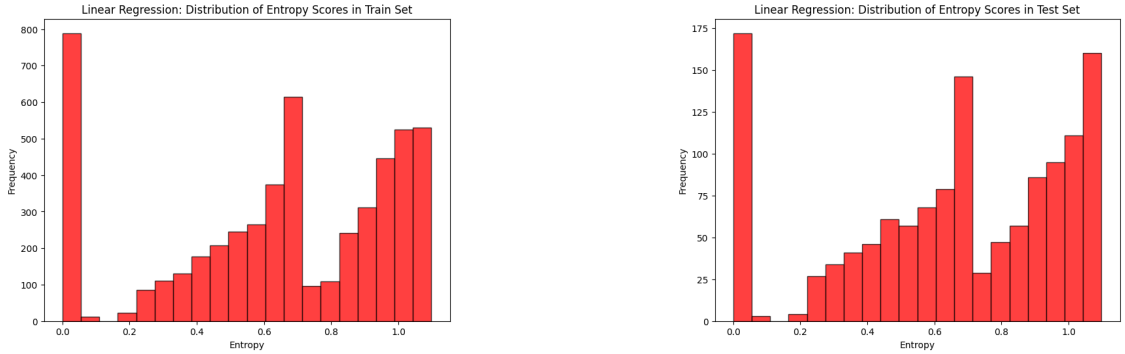

Figure 8.7: Histograms of entropy scores in linear regression - lasso train set (left) and test set (right).

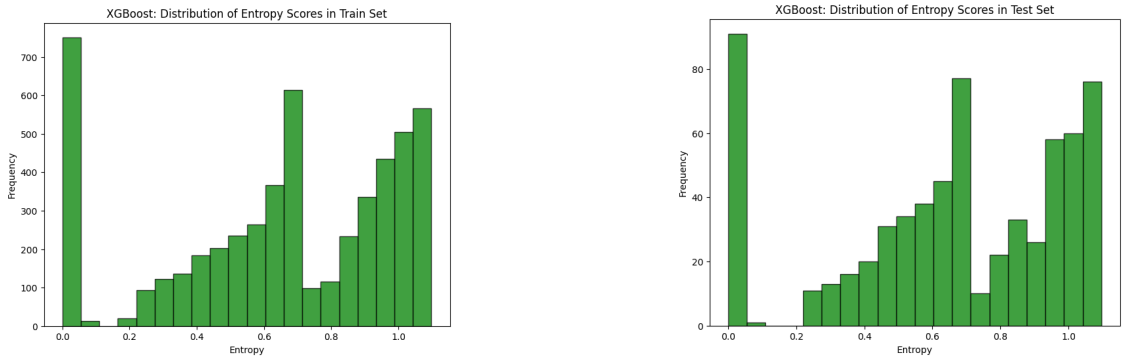

Figure 8.8: Histograms of entropy scores in XGBoost train set (left) and test set (right).

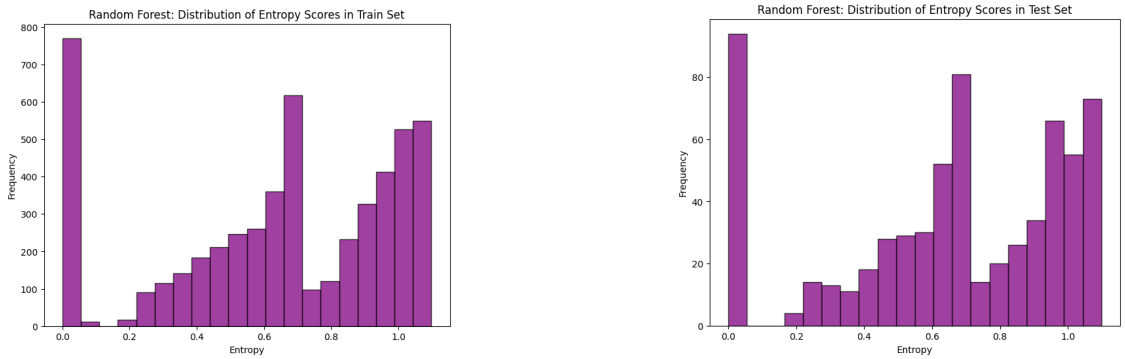

Figure 8.9: Histograms of entropy scores in random forest train set (left) and test set (right).

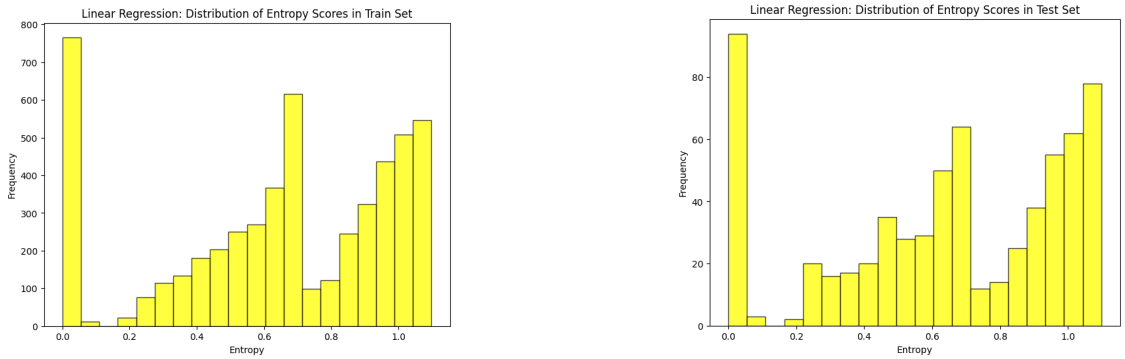

Figure 8.10: Histograms of entropy scores in SVR train set (left) and test set (right).

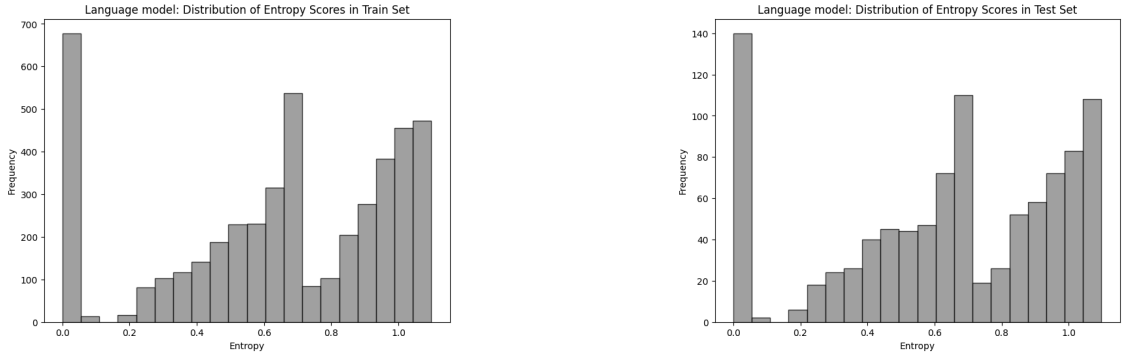

Figure 8.11: Histograms of entropy scores in language model train set (left) and test set (right).