universität wien

MASTERARBEIT | MASTER'S THESIS

Titel | Title

Political Leaning Prediction of Newspaper Articles

verfasst von | submitted by

Benedikt Köhler

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien | Vienna, 2025

Studienkennzahl lt. Studienblatt | Degree programme code as it appears on the student record sheet:	UA 066 926
Studienrichtung lt. Studienblatt | Degree programme as it appears on the student record sheet:	Masterstudium Wirtschaftsinformatik
Betreut von | Supervisor:	Univ.-Prof. Dr. -Ing. Benjamin Roth B.Sc. M.Sc.

Abstract

Political extremism, whether it is of far-left or far-right ideology, has become a critical issue faced by many countries worldwide. This, of course, is also the result of opinion-forming media coverage. Especially crises like the financial crisis in 2008, Corona or the recent Ukraine-Russia war seem to have an enormous influence on political motivated news coverage. [Peterson and Allamong, 2022] This thesis looks at using advanced natural language processing (NLP) techniques to predict the political biases in newspaper articles. How can we make use of NLP and pre-trained language models to detect the political leaning of news paper articles? And how can we optimize the prediction? To answer these research questions, we focus on a weak supervision approach and cross-validation methods to enhance prediction accuracy. With the help of pre-trained language models like DistilBERT, combined with innovative weight adjustment strategies, we aim at improving the classification of political leanings.

Our research focuses on issues such as the different ways political language is used by various news outlets and the necessity of efficient computing methods. With the help of several experiments, we show that our approach of adding weight to labels can improve the classification accuracy of a model. At the same time, we also show that our approach can quickly address difficulties with computing resources, because of very resource intensive algorithms. Therefore, this work sets the foundation for further development in this area which can lead to more reliable and efficient automated systems for political leaning detection.

This research shows that if you want to use weighted labels during training, more meaningful weights lead to better results than weights that are very similar. We found this by applying k-fold cross-validation methods, from which we used the validation scores for the articles in the left-out k-fold as weights for the labels and applied them to a final training. After making the weights more meaningful, for example by setting the minimum calculated weight to 0%, the maximum calculated weight to 100% and adjusting everything in between proportionally, we obtained the best results.

With another experiment (leave-one-out), which used a complete news publisher as a fold, we obtained less good results and were even significantly below our pre-trained language model baseline. This could be due to the fact that our dataset has a very large variance in total news articles per news publisher.

We found that there is still a lot of potential to improve political orientation detection and that the use of pre-trained language models offers a clear advantage over other methods such as logistic regression.

Kurzfassung

Extremismus, unabhängig davon, ob er aus dem linken oder rechten politischen Spektrum stammt, ist zu einer ernsthaften Herausforderung geworden, mit der viele Länder weltweit zu kämpfen haben. Dies ist unter anderem auch beeinflusst durch meinungsbildender Berichterstattung. Insbesondere Krisen wie die Finanzkrise 2008, Corona oder der jüngste Krieg zwischen der Ukraine und Russland scheinen einen enormen Einfluss auf die politisch motivierte Berichterstattung zu haben. [Peterson and Allamong, 2022] Diese Arbeit untersucht, wie man mithilfe von Natural Language Processing (NLP) die politische Ausrichtung eines Nachrichtenartikels vorhersagen kann. Wir fragen uns dabei, wie können wir NLP und Pre-trained Language Models nutzen, um die politische Ausrichtung von Zeitungsartikeln zu erkennen? Und wie können wir die Vorhersagen optimieren? Wir wenden dabei die Methode der Weak Supervision mit Cross-Validation an. Durch den Einsatz von Pre-trained Language Models wie DistilBERT und einer innovativen Anpassung von Label Gewichtungen während des Trainings, versuchen wir die Genauigkeit unseres Modelles zu verbessern.

Durch eine Reihe von verschiedenen Experimenten zeigen wir, dass sich mit unserem Ansatz - Label während des Trainings eine Gewichtung mitzugeben - die Genauigkeit eines Modells verbessern kann. Zeitgleich zeigen wir auch auf, dass man mit unserem Ansatz schnell auf Schwierigkeiten mit Rechenressourcen treffen kann. Dementsprechend kann die Arbeit ein Grundstein für weitere Forschungsarbeiten im Zusammenhang von politischer Ausrichtungserkennung der Nachrichtenartikel bilden, basierend auf Weak Supervision und Cross-Validierungsmethoden.

In dieser Arbeit wird dargestellt, dass wenn man gewichtete Labels während des Trainings nutzen möchte, aussagekräftigere Gewichtungen zu besseren Erfolgen führen, als wenn man Gewichtungen nutzt, die sich sehr ähnlich sind. Dies haben wir durch das Anwenden von k-fold Cross-Validierungsmethoden herausgefunden, bei welchen wir die Validierungsscores für die Artikel im ausgelassenen k-fold als Gewichtung für die Labels genutzt haben und diese bei einem finalen Training angewendet haben. Nachdem wir die Gewichtungen aussagekräftiger gemacht haben und zum Beispiel dadurch, dass wir die minimalste errechnete Gewichtung auf 0%, die maximalst errechnete Gewichtung auf 100% gesetzt haben und alles zwischendrin proportional angepasst haben, haben wir die besten Resultate erhalten.

Mit einem anderen Experiment (Leave-one-out), welches als Fold einen kompletten Nachrichtenverlag genutzt hat, haben wir weniger gute Resultate erhalten und lagen sogar deutlich unter unserer Pre-Trained Language Model Baseline. Dies könnte darauf zurückzuführen sein, dass unserer Datensatz eine sehr große Varianz an totalen Nachrichtenartikeln pro Nachrichtenverlag aufweist.

Zusammenfassend konnten wir feststellen, dass es noch viel Potenzial gibt, um die

politische Ausrichtungserkennung zu verbessern und dass der Einsatz von Pre-Trained Language Models in jedem Fall einen deutlichen Vorteil bringt im Vergleich zu anderen Methoden wie zum Beispiel Logistic Regression. Zudem können durch weitere Methoden, wie beispielsweise das Anpassen von Gewichtung pro Label, die Modelle noch weiter verbessert werden.

Contents

1. Introduction

As we encounter more and more political conflicts in our today's fast-paced digital world, understanding the political leanings and interests of news articles becomes more important. With the high availability and accessibility of online media, news article readers encounter a wide range of information, often influenced by different political perspectives. Identifying these political leaning accurately can help readers make better-informed decisions, have more neutral discussions and gain a more balanced view of current events.

This thesis explores, similarly as the work of [Liu et al., 2022], the use of advanced natural language processing techniques to predict the political leanings of newspaper articles. We focus on leveraging weak supervision and cross-validation methods to improve the accuracy of these predictions. Weak supervision allows us to use large amounts of noisy data for training, while cross-validation helps to improve our training efficiency and to ensure that our models are robust and generalizable.

Our study builds on the already existing pre-trained language models like DistilBERT, which have shown great performances in capturing the subtle nuances of language [SANH et al., 2019]. By combining these models with innovative weight adjustment strategies, similarly as proposed by the work of [Sedova and Roth, 2023], we are confident to enhance the performance of political leaning classification systems.

The research presented in this thesis addresses the important challenge of the need for efficient computational methods to improve the performance over today's baseline approaches. Through a series of experiments, we demonstrate the potential of our approach to improve classification accuracy in terms of political stance detection.

In the following chapters, we review the existing literature on political leaning prediction (2) describe our experimental setup (5), present our findings (6), and discuss the implications of our results (7). By introducing new methodologies in this field, we hope to contribute to better knowledge of political bias in the media and support the development of more accurate and reliable classification systems.

2. Foundation

The field of natural language processing (NLP) has witnessed exponential growth in recent years, driven by advancements in machine learning algorithms, deep learning architectures, and the availability of large-scale text corpora. In this section we are further trying to understand which work has already been done to support the topic of political leaning prediction of news paper articles utilizing NLP. Additionally, we are also investigating the current state-of-the-art methodologies and solutions in this domain.

2.1. Natural Language Processing

Natural Language Processing is a field that combines computer science, statistics, and linguistics to teach computers how to process and understand human language automatically. Early research, like the work of [Manning, 1999], introduced statistical methods for tasks such as language modeling, sentence structure analysis, and text classification. These statistical methods are applying probability-based techniques to handle the complexity of the human language.

Since its early foundations, NLP has gone through a noticeable evolution. A major breakthrough was the introduction of Word2Vec by [Mikolov et al., 2013], which transformed how words are represented by capturing their meanings through numerical vectors. This model uses two key methods—Skip-gram and Continuous Bag of Words (CBOW)—to learn word relationships. Skip-gram predicts surrounding words from a given word, while CBOW predicts a word based on its surrounding words. These compact word representations encode both meaning and grammatical patterns, making them useful for various NLP tasks.

While Word2Vec captures semantic relationships through neural networks, CountVectorizer from scikit-learn[1] [Pedregosa et al., 2011] is another more traditional approach to text vectorization. It converts text documents into numerical feature vectors based on term frequencies. This method creates a matrix of token counts, where each document is represented as a vector in which each dimension corresponds to a term from the vocabulary, providing a simpler but computationally efficient baseline for text classification tasks. The resulting document-term matrix can be used directly for machine learning tasks or further transformed using techniques like TF-IDF weighting.

Another important work to acknowledge is the breakthrough by [Bahdanau, 2014], which was a major step forward in improving attention mechanisms. His model for machine translation introduced a way to translate sentences of different lengths. Traditional

[1] https://scikit-learn.org/1.2/modules/generated/sklearn.feature$_e$*xtraction.text.CountVectorizer.html*

translation models, based on the encoder-decoder approach, could only convert a fixed-length input into a fixed-length output. In contrast, Bahdanau's approach processes words one by one while constantly checking for the most likely next word, allowing for more flexible sentence structures.

His work was motivated by [Cho, 2014] which discovered that a basic encoder-decoder translation model lacks efficiency when translating long source sentences. This is largely due to the fact that the encoder of this basic approach needs to compress a whole source sentence into a single vector, while the approach by [Bahdanau, 2014] as previously described, handles this issue more dynamically.

While much progress has been made in the field, two key studies that have significantly shaped its development are the works of [Vaswani, 2017] and [Devlin, 2018].

The self-attention mechanism in the Transformer model, introduced by [Vaswani, 2017], is an improvement compared to the other attention methods like the one we have seen earlier developed by [Bahdanau, 2014]. Instead of processing input step by step, like traditional models, self-attention allows the model to focus on the most important parts of the input. To do so, it compares each word (query) to all other words (keys). During the comparison it assigns scores based on how relevant they are. These scores are then adjusted using a softmax function, which converts them into weights. After that, the model then uses the weights to combine the corresponding word representations (value vectors) into a final result. This approach helps the model better understand context.

Additionally, multi-head attention takes this concept further by creating multiple sets of queries, keys, and values. Each 'head' focuses on different parts of the input, allowing the model to capture more complex relationships between words. This design speeds up the processing and helps the model understand long-range dependencies more effectively.

BERT (Bidirectional Encoder Representations from Transformers), introduced by [Devlin, 2018], improved the computational understanding of human language by making use of both directions in a sentence. Unlike earlier models like GPT, which processes text from left to right, BERT reads words in both directions at the same time. It does this using a method called masked language modeling (MLM), which randomly masks some tokens in the input and predicts only the masked tokens, allowing the model to learn contextual representations from both directions. The model therefore learns to predict words based on surrounding words. This method enables BERT to better understand the relation between words in different contexts.

Furthermore, BERT is built on top of the Transformer encoder from [Vaswani, 2017] and stacks multiple layers of self-attention and feed-forward neural networks. Firstly, it is trained on large datasets of texts to learn complex language patterns and relationships. After the pre-training, it can further be fine-tuned for specific tasks like text classification or as in our case the understanding of political leaning, while maintaining its deep understanding of language. BERT's ability to process text bidirectionally, combined with its large-scale pre-training, has set new benchmarks for understanding context in natural language processing.

With the advancements in this field, NLP is now used in many areas, such as sentiment analysis [Xu et al., 2019], speech recognition [Higuchi et al., 2022], text sum-

marization [Grail et al., 2021]. It is also used with notable success in social media analysis[Nguyen et al., 2020] and many other domains. Given its wide range of use cases, NLP is well-suited to address our research question.

For our experiments, we use Transformer models from [Vaswani, 2017] together with a BERT-based pre-trained language model. Additionally, we are also using the CountVectorizer from scikit-learn to create a logistic regression baseline, allowing us to compare deep learning methods with a more traditional approach.

2.2. Political Leaning Prediction

Given the significant role of ideology detection in political science, extensive research has been conducted in this area. For instance, [Mullins, 1972] discusses the challenges associated with accurately identifying political leanings. Mullins aimed to establish a concept that helps empirical research and enhances political ideology detection.

Additionally, the work of [Laver and Garry, 2000] made a significant contribution to this field by analyzing various systematic methods employed to estimate the political orientation of texts. Their approach goes beyond previous efforts, as they believe that with the amount of content of political viewpoints generated daily makes manual analysis and labeling impractical. Consequently, they attempted to create a computer-coding technique to deliver accurate estimates of policy positions.

Even today, research on this topic continuously goes on. Recent studies, such as those by [Guess et al., 2020] or [Peterson and Allamong, 2022] highlight the difficulties of detecting political ideology and misinformation as a result of rapidly spreading news coverage via the internet from various different sources, including unverified ones.

Given the technical improvements of machine learning, especially in the field of NLP, there has been research conducted to address the challenges associated with the fast spreading news coverage. Some recent approaches include utilizing NLP for predicting the political stance of Tweets, as seen in works such as [Preoţiuc-Pietro et al., 2017] or [Stefanov et al., 2020]. Others focus on predicting the political leaning of speeches or texts, as demonstrated in studies by [Iyyer et al., 2014], [Sawhney et al., 2020] or [Abercrombie and Batista-Navarro, 2020].

Nevertheless, considerable research has been conducted on predicting the political stance of newspaper articles. For instance, [Kulkarni et al., 2018] found that the title, content, and link structure of newspaper articles often already reveal their political ideology. To predict the political ideology they developed a new model, MVDAM (Multi-view document attention model), which outperformed baseline models significantly. In their approach, the model was trained using a dataset sourced from ALLSIDES.COM[2]. Another approach as in [Sinno et al., 2022] firstly carefully annotates newspaper articles utilizing an annotation agreement, which was developed by an political scientist. The dataset, comprising 721 labeled paragraphs, was divided into training, validation, and test sets with

[2]https://www.allsides.com/media-bias

a 80/10/10 split. Afterwards various model approaches were employed such as recurrent neural networks, pre-trained language models, focal loss and task-guided pre-training. Also [Liu et al., 2022] adopted a similar approach, utilizing the ALLSIDES.COM dataset. After a cleaning process and downsampling to approximately 2.3 million articles, they held out 30,000 articles for validation and manually labeled 888 articles as a test set. They fine-tuned a pre-trained language model (roBERTa) for training, resulting in a model that outperformed most baselines used in their experiment.

Although there is already existing research on the political leaning detection of newspaper articles, we are additionally pursuing other approaches towards political stance prediction using NLP and determine the current state-of-the-art methodology.

Comparing the previously mentioned studies, we observe significant similarities in their experimental setups. Whether predicting social media posts, speeches, or newspaper article leanings, as demonstrated in works such as [Stefanov et al., 2020], [Sawhney et al., 2020], [Abercrombie and Batista-Navarro, 2020], [Sinno et al., 2022] and [Liu et al., 2022], all experiments leverage a BERT-based pre-trained language model to outperform baseline performance. Particularly, the experiment by [Sinno et al., 2022] highlighted the superiority of a BERT-based approach over other methods. Considering these findings, a BERT-based approach emerges as the state-of-the-art solution.

3. Dataset Creation

In order to effectively train a pre-trained language model to accurately predict the political stance of newspaper articles, we need to utilize a comprehensive dataset which includes a diverse range of articles, each containing a corresponding label. In this chapter, we describe the selected dataset for the training and discuss the annotation process utilized in generating the validation and test sets.

3.1. Description of the Dataset

For this study, the POLUSA dataset [Gebhard and Hamborg, 2020] was selected to be utilized, which was created through a project of Lukas Gebhard of the University of Freiburg and Felix Hamborg of the University of Konstanz. Initially comprising 3.6 million articles, the dataset underwent extensive cleaning procedures, resulting in a refined version containing approximately 900,000 articles. The cleanup process involved several steps, including:

1. Base Selection
2. Near-duplicate removal
3. Selection of English news articles covering policy topics
4. Assignment of political leanings
5. Temporal and popularity-based balancing

The assignment of political leanings holds particular importance for this experiment. It is important to note that political leanings are not assigned to individual articles but rather to entire news outlets. This was conducted carefully by taking into account eight distinct measures.

> "To ensure reliable ratings, we systematically aggregate eight measures from prior work. The measures are self-declarations by outlets, results of content analyses by social scientists, and news consumer data from surveys and social networks. We label outlets with low agreement (<0.75) among the eight rating dimensions as 'undefined.' " [Gebhard and Hamborg, 2020]

Considering the information mentioned in the quote, all assigned political labels exhibit a confidence level exceeding 75%. In instances where the confidence level falls below

this threshold, a fourth label, 'undefined', was introduced. Consequently, the cleaned up dataset, which consists of approximately 900.000 articles, holds 4 different labels - left, right, center or undefined.

The POLUSA dataset contains articles from 18 different news outlets, which also include various metadata, such as authors, publication date, URL, political leaning and outlet's name. The distribution of articles within the dataset is as follows: 31% belonging to the left-wing, 27% to the center, 16% to the right-wing, and the remaining 26% labeled as undefined. Overall, this dataset provides a valuable resource for researchers investigating NLP techniques within the context of political media analysis.

3.2. Training Set Collection

In this section, we focus on the dataset characteristics, and the distribution of the data used for the training set. A robust and representative training set builds the base for a successful training of machine learning models. Understanding the structure and distribution of the training set is highly needed to evaluate the performance and generalizability of the trained models.

Firstly, we analyze the distribution of the dataset. For this research, two training datasets were created: a smaller set containing 365,389 articles and a larger set with 905,165 articles. Due to significant performance issues encountered during training of the model, we proceeded with the smaller training set. Consequently, this section focuses on the dataset used for our research.

Our training dataset contains 365,389 articles of 18 different news paper outlets. These articles are labeled as either right, left, center, or undefined in terms of political leaning. Since we are not considering articles with an undefined political leaning for our training, we filter them out, leaving us with 271,415 articles from 13 different outlets.

Figure 3.1 shows the distribution of articles by political leaning. The figure indicates that 111,261 articles are labeled as left, which is nearly double the number of right-leaning articles, which are in total 53,729. The colors in this graph represent different political leanings: Blue for left, red for right, and green for center.

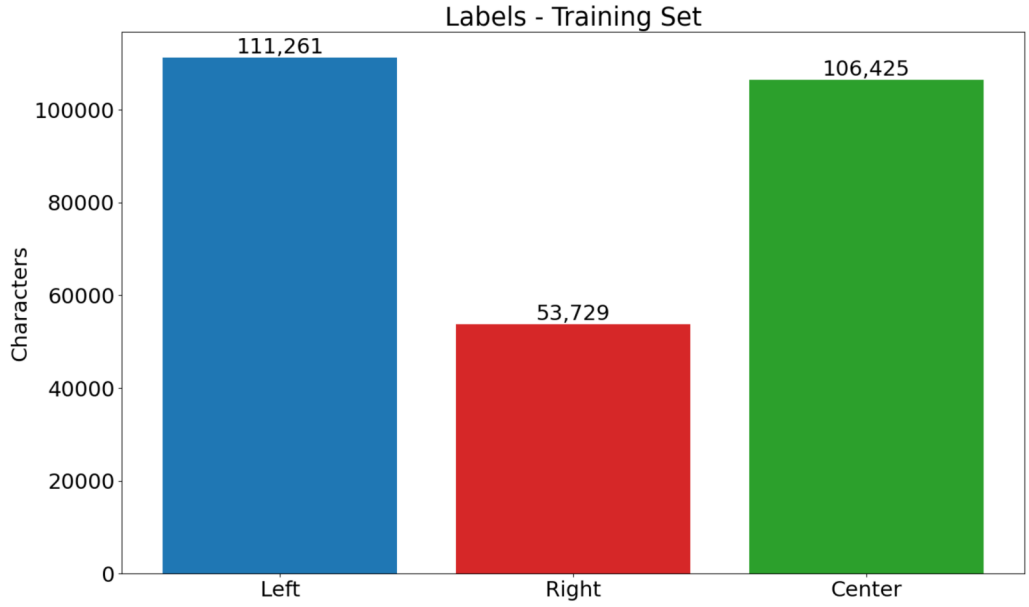

Figure 3.1.: Articles per Political Leaning of Training Set

The distribution of articles per outlet is illustrated in figure 3.2.

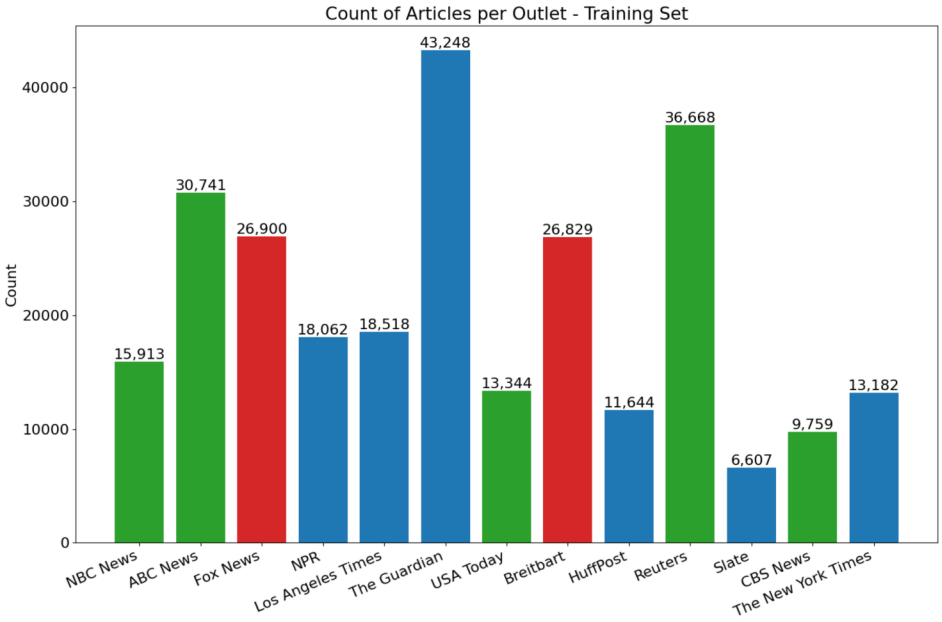

Figure 3.2.: Articles per news paper outlet of training set

Lastly, we also examine the length of the articles. As illustrated in figure 3.3 the articles length differs in a range from less than 500 characters up to more than 5500 characters per article. Most articles contain up to 2500 characters, with the number of articles decreasing as the length increases. This results in 68.77% of all articles in the training set containing fewer than 2500 characters. For the length analysis, only the headline and body of each article were considered, as these were the only two components used for training.

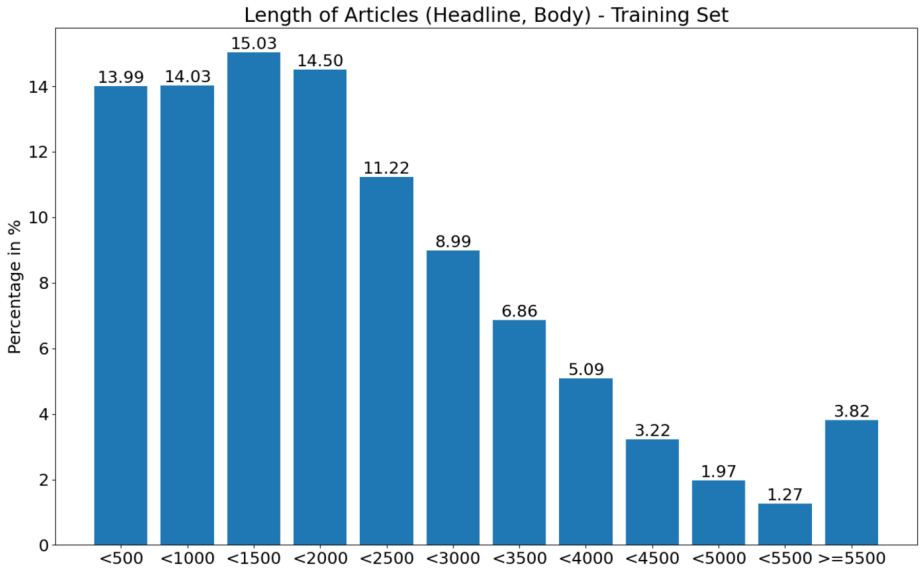

Figure 3.3.: Length of Articles of Training Set (in Characters)

A complete overview over the characteristics of the training data set can be retrieved from table 3.1.

Training Dataset	**Count**
Total Articles	365,389
Articles (excl. undefined)	271,415
Political Leaning Distribution:	
Left	111,261 (41.0%)
Right	53,729 (19.8%)
Center	106,425 (39.2%)
News Outlets:	
The Guardian (Left)	43,248
Reuters (Center)	36,668
ABC News (Center)	30,741
Fox News (Right)	26,900
Breitbart (Right)	26,829
Los Angeles Times (Left)	18,518
NPR (Left)	18,062
NBC News (Center)	15,913
USA Today (Center)	13,344
The New York Times (Left)	13,182
Huffpost (Left)	11,644
CBS News (Center)	9,759
Slate (Left)	6,607
Number of Outlets	13

Table 3.1.: Overview of Training Dataset Characteristics

3.3. Annotation of Validation Set and Test Set

Next, we discuss the process of manually annotating the validation and test datasets. Given the small scale and limited budget of this research, it is important to note that no annotator agreement was established beforehand. Consequently, the data is labeled based on our best judgment.

3.3.1. Setup of Labeling Tool

Labeling the data to our best knowledge and as efficient as possible needs some preperation. Since we do not want to be influenced by seeing more data than what the machine will see, we firstly started to develop a small tool to help us annotating the data. The tool was set up with the help of the python UI framework Tkinter[1]. When developing the

[1] https://docs.python.org/3/library/tkinter.html

tool, we made sure that we have a relatively simple layout which provides us only the following information:

1. The ID of the article in the dataset
2. The header of the article
3. The body of the article
4. Our own annotation (per default set to UNDEFINED)
5. The amount of articles rated in this dataset

Furthermore it provides the following actions for the user to navigate around the dataset and annotate the data:

1. Change the ID of the article and retrieve the article trough clicking on "Get".
2. Change the label of the leaning to either LEFT, RIGHT or CENTER.
3. Save the article current state by clicking on "Rate".
4. Navigate to the previous or next article in the dataset.

All the information provided can be seen in the example showed in figure 3.4.

As our datasets articles ranges from articles from 2017 until the end of 2019, we were eager to annotate articles from all years rather than concentrating only on one year. Therefore the goal was it to distribute the amount of annotated articles relatively equal throughout the time period that we have available. In total we labeled 157 articles from 2017, 130 from 2018 and 120 articles from 2019.

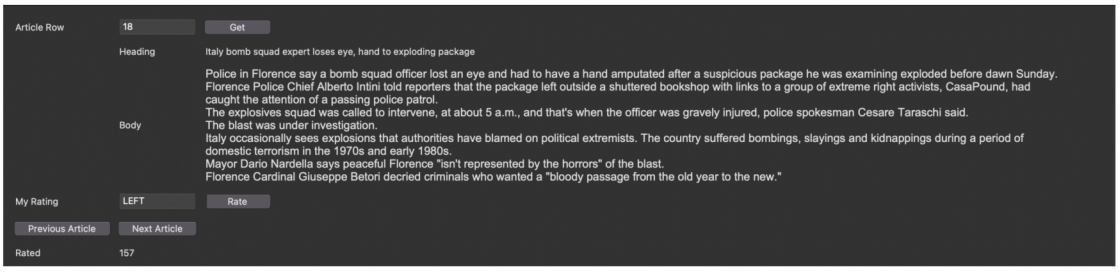

Figure 3.4.: Annotating Tool

During the annotation process, we are trying to stay as neutral as possible towards the articles. To achieve this, we avoid providing any information about the outlet's political

leaning or the article's corresponding label in our tool. This approach helps us preventing bias in our decisions. Maintaining neutrality is essential because the goal is to identify articles which labels may differ from the political leaning assigned to their respective news outlets.

3.3.2. Description of Validation and Test Set

In the previous section, we explained how we annotated a set of articles manually. After completing the annotation, we had a dataset of 407 articles ranging from the years 2017 to 2019. To ensure we have reliable gold-validation and gold-test sets for evaluating the accuracy of our computational approach, we needed to divide this dataset. While splitting the articles, we ensured that both sets included articles from all three years, rather than simply dividing the dataset evenly.

Therefore, to split the dataset, we ordered the articles by their publication date and assigned every other article to the test set and removing it from the validation set. As a result, the validation set consists of 204 newspaper articles, while the test set contains 203 articles. Furthermore, we wanted to ensure that the characteristics of both sets were similar and not significantly different. For this reason, we analyzed the key attributes of the validation and test sets.

Firstly, we examined the distribution of political leanings in the validation and test sets. The distribution for the validation set is shown in figure 3.5, while figure 3.6 illustrates the distribution for the test set. As indicated in these figures, the validation set includes 76 left-leaning articles, 46 right-leaning articles, and 80 center-leaning articles. Similarly, the test set contains 85 left-leaning, 42 right-leaning, and 78 center-leaning articles. When comparing these distributions to the training dataset described in section 3.2, we observe that they are relatively similar. Nevertheless, there are slight differences, which align with our expectations since the self-annotated data provides article-level labels rather than outlet-based labels.

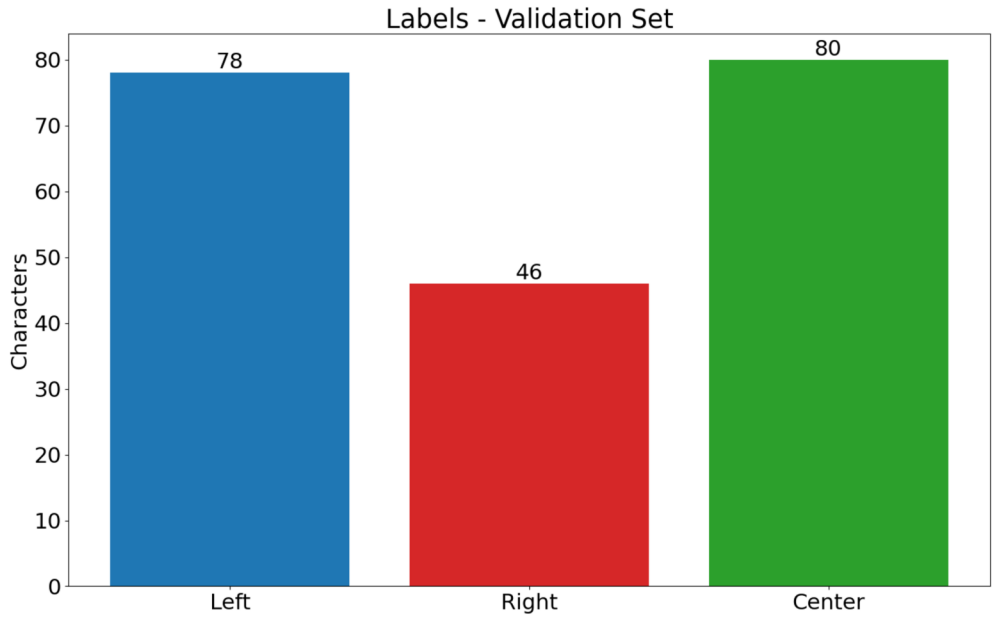

Figure 3.5.: Distribution of Political Leaning - Validation Set

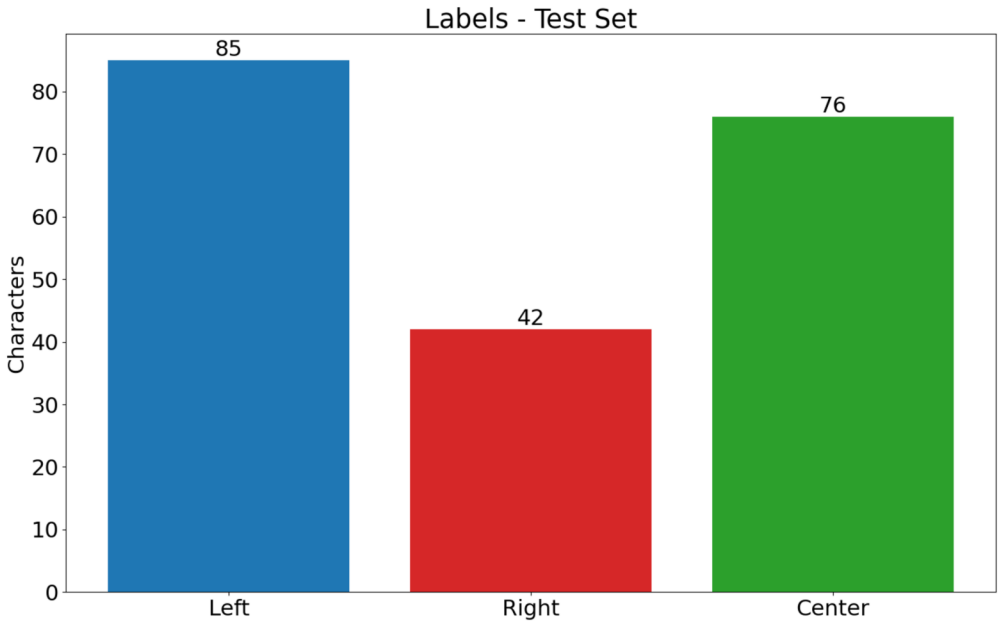

Figure 3.6.: Distribution of Political Leaning - Test Set

This leads us to the next interesting topic, in which we identify how many articles were assigned to a different label through our annotation process compared to the original label that was provided through the newspaper outlet. To explore this, we counted the number of label changes and identified how often a label shifted from one category to another. The results are presented in figure 3.7 and figure 3.8. These tables display the original outlet-assigned labels on the left-hand side, while the columns represent the self-annotated labels.

When analyzing the results, we can observe that for each political leaning, the most common self-annotated label matches the original label provided by the newspaper outlets. Nevertheless, a big amount of the articles underwent a label change: 42.43% in the validation set and 40.48% in the test set. Interestingly, most of these changes shifted to the closest adjacent label. For example, an article originally labeled as left-leaning was often reclassified as more centrist rather than right-leaning.

Validation Set

	To Left	To Center	To Right
Left	36	19	7
Center	12	35	7
Right	9	16	24
Undefined	21	10	8

Figure 3.7.: Change of Labels After Annotation - Validation Set

Test Set

	To Left	To Center	To Right
Left	42	13	6
Center	14	39	9
Right	16	14	24
Undefined	13	10	3

Figure 3.8.: Change of Labels After Annotation - Test Set

Taking into consideration the validation and test set annotation, we are left with the dataset characteristics as shown in 3.2.

	Training Set	**Validation Set**	**Test Set**
Total Articles	271,415	204	203
Political Leaning Distribution:			
Left	111,261 (41.0%)	78 (38.2%)	85 (41.9%)
Right	53,729 (19.8%)	46 (22.5%)	42 (20.7%)
Center	106,425 (39.2%)	80 (39.2%)	76 (37.4%)

Table 3.2.: Comparison of Training, Validation and Test Set

Finally, let's examine the length of the articles. Figure 3.9 shows the article lengths for the validation set, and figure 3.10 shows them for the test set. We can observe in these two figures, that article lengths vary from those in the training set. In the validation set, the longest articles are up to 4000 characters, while in the test set, they reach up to 3500 characters. Most articles in both sets - validation and test set - are between 0 and 2000 characters long.

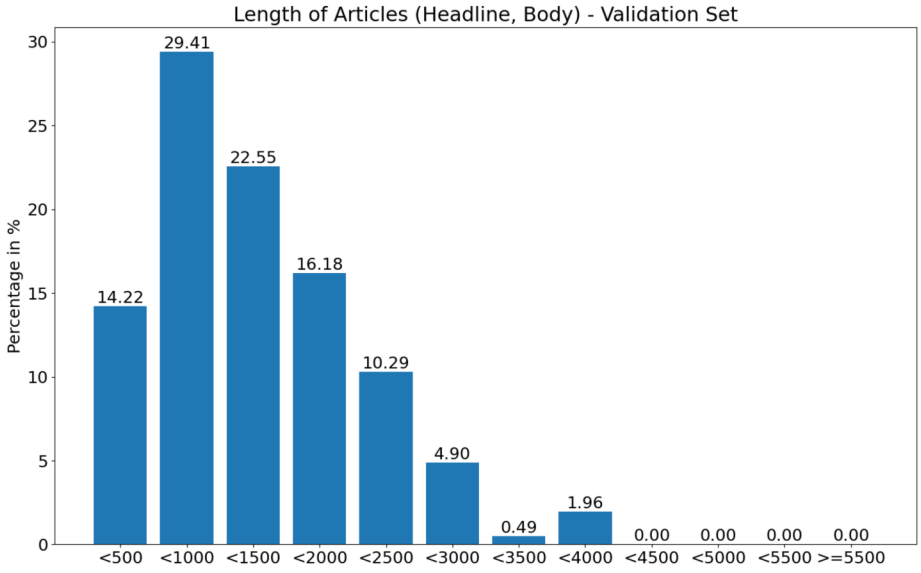

Figure 3.9.: Length of Articles - Validation Set (in Characters)

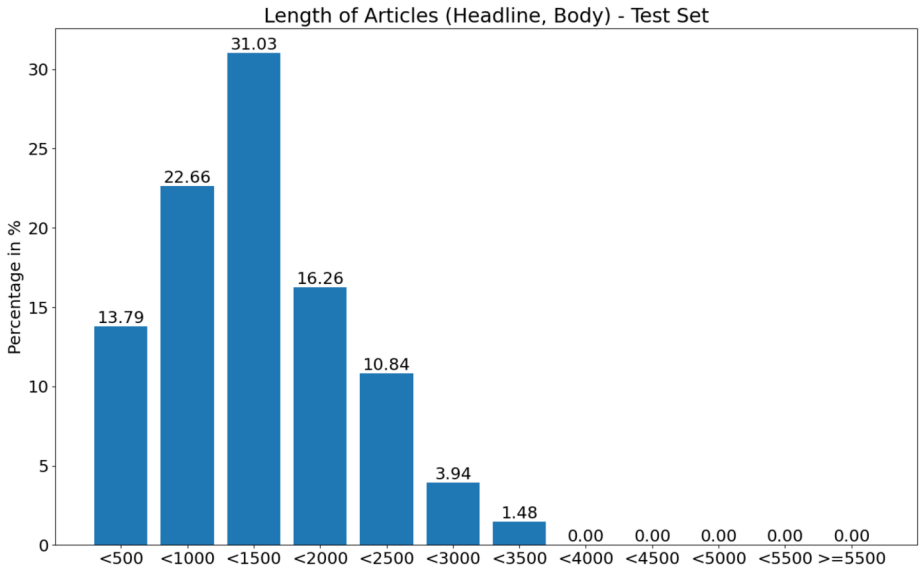

Figure 3.10.: Length of Articles - Test Set (in Characters)

3.3.3. Challenges Faced During Annotation

Although we had an efficient tool for annotating the data, the process still took a lot of time. Reading all the articles was time-consuming, and if we could not clearly identify the political leaning of an article, we chose to skip it and went on to the next article. This approach ensured that our validation and test sets were of high quality.

To guide our annotation, we have set up five identifiers to consider while annotating the data:

1. The author's opinion-based writing
2. Quotes that support either left or right views
3. References to race in the article's topic
4. Exaggerations to emphasize a point for either side
5. Making fun of political opinion

Furthermore, we did not have a formal annotator agreement set up, which means we lacked strict guidelines to make sure that articles were labeled correctly. However, we tried to follow the previously mentioned identifiers to help us decide when to label an

article as right, left, or center. Sometimes, it is fairly easy to determine the political direction of an article, while other times it is more challenging.

For instance, if a president discusses stopping illegal immigration and uses offensive language, and the article's author fully supports this, we can label it as right-wing. However, other articles might cover the same topic neutrally. This makes it very hard to choose the right label because the topic clearly leans to the right while the author might report about it without bias.

Ideally, news outlets should report topics without influencing readers' opinions. Therefore, while annotating, we focused not only on the topic's political leaning, but especially on how the writing style affected the reader's view. If an article remained neutral and did not push a particular leaning, we labeled it as center.

Lastly, it is important to note that annotating takes a lot of time. On average it took us approximately 1.5 minutes to read an article. This does not include the time we needed to elaborate the leaning of an article. Furthermore, through the skipping of articles that we were not sure about the leaning, it took as on average 5 articles for one to be labeled. This leaves us already at approximately 50 hours of pure reading time, not taking into consideration the set up of the tool, the data structure and the time to identify the label. Therefore, we focused on articles that provided enough information and text to determine their political leaning, while avoiding very long articles that might be too time-consuming to label.

4. Methodology

The following chapter explains the methods used for the study and how these methods should affect and improve the outcome and results of the experiment.

The purpose of this study is to utilize machine learning and improve the methods used to predict the political orientation of the newspaper articles. To achieve this goal, we made a decision regarding the methodology depending on the data that we can utilize for training of our algorithm. As we use the POLUSA dataset [Gebhard and Hamborg, 2020], we used the included data and its constraints to make a decision. Due to the labels in this data set, which are solely based on the political leaning of the news outlet, we have to assume that the assigned labels are correct for most of the articles but not for each and every one. Therefore, we came up with a solution that filters the noisy data as well as possible. To realize this, we are using a weakly supervised approach, which has the benefit of not needing manually labeled training data. In the next section, we have a deeper look into weak supervision and its characteristics, to get a better understanding on why this type of supervision is a good fit for this research.

4.1. Weak Supervision

Weak supervision is one of the pivotal techniques in machine learning, in particular in scenarios where obtaining large-scale labeled data is challenging or extremely expensive. In contrast to traditional supervised learning, which relies on carefully labeled datasets, weak supervision leverages noisy, limited, or imprecise sources of labeling to train models. This approach fits well with the objectives of this study, where the POLUSA dataset provides labels based on the political leanings of news outlets, inherently introducing noise and inconsistencies at the article level.

One of the main advantages of weak supervision is the ability to leverage diverse and extensive sources of weak labels without extensive manual annotation of the data. This is especially of benefit in political text analysis domains, where manual labeling can be time-consuming and conducted with a subjective perspective (3.3.3). By integrating multiple weak sources, such as heuristic rules, noisy labels, or external knowledge bases, weak supervision frameworks can aggregate these signals to produce a more reliable supervisory signal for model training [Ratner et al., 2016].

Recent advances in the topic of weak supervision have introduced sophisticated frameworks that improve the quality of generated labels. In particular, Snorkel [Ratner et al., 2017] has been a pioneering work, providing a systematized approach to programmatically generating and managing weak labels. Taking this work into consideration, following research has focused on improving the scalability and robustness of weak supervision

techniques.

For example, [Karamanolakis et al., 2021] present a scalable method with their work ASTRA that also leverages unlabeled data and provides pseudo-labels through self-training for instances that could not be otherwise labeled with existing rules. Through their research they are trying to ensure that the pseudo-labels maintain high fidelity regardless the inherent noise in individual sources. In addition to ASTRA, [Park et al., 2023] introduced PLAT, a framework that leverages zero-shot text classification models to create weak labels without the need for any labeled training samples or hand-crafted rules. By using models trained on subtasks, PLAT increases the quality of weak labels and significantly improves the performance of downstream classifiers, especially in low-resource environments.

In the context of predicting the political orientation of newspaper articles, weak supervision offers a pragmatic solution to the problem of label noise introduced by outlet-based labeling. This study leverages the labels from POLUSA dataset and reduces potential inaccuracies utilizing a weakly supervised approach. We achieve this by creating labeling functions that identify very noisy articles in the label category and reduce their importance to the model. This is expected to improve the quality of the training data for the predictive model.

Furthermore, the performance of the incorporated model of cross-validation techniques with weak supervision will always be tested against gold-data, which is a set of manually labeled data. This approach not only enhances the model's robustness but also validates the efficiency of the weak supervision framework in capturing the underlying political orientations accurately [Zhang et al., 2021] while preventing a overfitting of the model. Similarly does the work of [Liu et al., 2022] also use an approach in which they are testing their trained model against manually annotated data.

4.2. Cross-Validation

Cross-validation is a technique in machine learning, used to evaluate the performance of a model and prevent overfitting. It works by dividing the data into training and testing subsets in a systematic way, helping to measure model performance, reduce overfitting, and improve reliability.

Regarding to Berrar [Berrar et al., 2019], cross-validation serves as a method for evaluating models and as an important step in selecting the most suitable model. It offers a more precise estimate of a model's performance on new, unseen data compared to a basic train-test split. The paper highlights that cross-validation helps in identifying the model that best generalizes to independent datasets.

Furthermore, the paper discusses variations of cross-validation, such as stratified k-fold cross-validation, which ensures that each fold is representative of the entire dataset by maintaining the same distribution of classes. This is particularly important in imbalanced datasets where certain classes are underrepresented.

Two of the common forms of cross-validation are the K-Folds and Leave-one-out cross-validation, which are also used for this study.

4.2.1. K-Folds

K-Fold Cross-Validation is one of the most widely used methods. In this approach, the dataset is divided into "K" equally sized folds. The model is trained on "K-1" folds and is after the training validated on the left out fold that is remaining. This process is repeated as many times as we have folds, with each fold serving as the validation set exactly once. Usually the final performance metric is the average of the scores from each iteration. [Kohavi et al., 1995]

4.2.2. Leave-One-Out

Leave-One-Out Cross-Validation is a different variant of K-Fold Cross-Validation where "K" is equal to the number of data points or special cases in the dataset. This means that each iteration uses a single data point or a case as the validation set and the remaining data points as the training set. [Berrar et al., 2019] In our setup, we can make use of the different outlets and use them as a case. Therefore, we can leave out one outlet and use it as the validation set, while we use the remaining outlets for the training.

4.3. Purpose of Employing Weak Supervision With the Support of Cross-Validation

In this section, we are bringing the two concepts together of weak supervision and cross-validation to understand on how it can help for our work.

The integration of weak supervision with cross-validation techniques serves as a good approach to improve the reliability and performance of machine learning models, particularly in set ups where labeled data is scarce or inherently noisy [Sedova and Roth, 2023]. In this study, the main goal of employing weak supervision is to use the POLUSA dataset's outlet-based labels effectively, which introduce a degree of label noise at the article level.

Weak supervision helps the model make use of noisy labels by giving them different levels of importance using weighting techniques. This technique of applying weightings to labels allows the model to focus more on reliable labels while reducing the influence of less reliable ones, which should lead to improvement of overall accuracy. A method like this can be especially useful in cases where we have access to a comprehensive database with noisy data or manual labeling is not feasible.

In this setup, cross-validation plays an important role as it provides a way to evaluate the performance of the model and determine the best weights for the weak labels. The dataset is divided into several subsets during the cross-validation. The model is trained on some subsets of the data and tested on the remaining ones. Through this repeated training and testing we are able to identify the performance of the model on a small subset of the data and use the created metrics for our weighting technique.

The weights assigned to the weak labels are computed during the cross-validation training phases. After each iteration of a fold, we can analyze how each label or batch affects the accuracy of the model through the different training and validation splits.

Therefore, the system can identify which labels are more reliable in representing the underlying political orientations. Consequently, labels that consistently contribute to higher validation performance are assigned greater weights, while those that introduce noise are down-weighted. This adaptive weighting mechanism ensures that the model learns from the most reliable signals within the weakly supervised data, thereby enhancing its predictive capabilities.

Furthermore, this setup aligns with recent conducted research in weak supervision frameworks, which emphasize a procedure of leveraging multiple weak labeling sources and systematically managing their associated uncertainties [Sedova and Roth, 2023]. By combining weak supervision with cross-validation-based weighting, the study not only proposes a approach to tackle the problems created by noisy labels but also establishes a foundation for our particular research for the reliable model training in politically oriented textual analysis.

Therefore, the purpose of employing weak supervision supported by cross-validation in this study is to optimize the utilization of noisy, outlet-based labels from the POLUSA dataset. This should help the model in distinguishing between reliable and unreliable labels through adaptively applying weights to the labels confidence.

5. Experimental Setup

This chapter describes the experimental framework designed to evaluate the effectiveness of predicting political leanings in newspaper articles using weak supervision and cross-validation techniques. The experimental setup is structured to face previously describe challenges in working with noisy, outlet-based labels while ensuring robust model evaluation and improving performance to comparable baselines.

The experiments are designed to test the hypothesis that incorporating weak supervision with adaptive label weighting, can improve the accuracy of political orientation predictions compared to traditional supervised learning approaches. This setup particularly focuses on handling the noise present in the POLUSA dataset, where labels are derived from news outlet political leanings rather than article-specific annotations.

One substantial part of the experimental setup was of course the creation and the labeling of the validation and test set as described in 3.3.

All the other key components of the experimental setup are covered in this chapter: First, the selection and implementation of DistilBERT as our primary language model, followed by necessary preprocessing steps. We then establish a logistic regression baseline to test our results against, before fine-tuning DistilBERT with our weak supervision approach. Finally, we describe our cross-validation methodology, which employs multiple strategies (small k-folds, large k-folds, and leave-one-out) to compute weights for weak labels.

5.1. Rationale for Choosing DistilBERT

DistilBERT, which is a distilled version of BERT (described in 2.1), was selected as our primary language model for several reasons. Firstly, it demonstrates strong performance in various NLP tasks while being significantly more efficient than its larger counterparts (like BERT or roBERTa). Through knowledge distillation, DistilBERT retains 97% of BERT's language understanding capabilities while being 40% smaller and 60% faster [SANH et al., 2019]. This efficiency is a big aspect for our experimental setup, which involves multiple training iterations across different cross-validation folds with a big set of data, where computational performance is highly needed.

Compared to the original BERT base model (110M parameters) and RoBERTa base (125M parameters)`https://huggingface.co/transformers/v2.9.1/pretrained_models.html`, DistilBERT's lighter architecture (66M parameters) significantly reduces computational requirements without substantially compromising performance [SANH et al., 2019].

The fact, that the model is a pre-trained language model provides significant transfer

learning benefits. Through pre-training on vast amounts of general text data, pre-trained languange models like DistilBERT have developed a good understanding of language patterns, contextual relationships, and semantic nuances [Gururangan et al., 2020]. This pre-existing knowledge can be transferred to our specific task of detecting political leanings, allowing the model to require less task-specific training data to achieve good performance.

These transfer learning features are highly valued due to the presence of noisy labels and the extensive amount of language in our dataset. Additionally, DistilBERT's popularity in the research community and its availability in widely used frameworks make it easier to reproduce results and compare them with future studies. [Wolf et al., 2020].

The characteristics of efficient computation and proven reliability, make DistilBERT a good choice for our experimental framework, especially when we consider the computational constraints and the need for multiple training iterations in our cross-validation setup.

5.2. Preprocessing Steps

The POLUSA dataset is provided as a collection of CSV files, each containing details such as the outlet, headline, lead, body, and more information for each article. To make this data usable for our study, we started by transforming it into a structured format. Each CSV file was loaded using pandas[1] and then saved in a compressed binary format with joblib[2] for storage efficiency.

This binary data was then utilized for the annotation process, as explained in section 3.3.

After completing the annotation, we ensured that the training dataset did not include any self-annotated articles. Only after ensuring this, we proceeded with preprocessing the training, validation, and test sets.

To ensure the quality and consistency of our textual data, we implemented a comprehensive preprocessing pipeline that transforms the raw article text into a standardized format suitable for model training. The preprocessing steps are applied to the headline, lead, and body of each article in both training and evaluation datasets.

[1] https://pandas.pydata.org/
[2] https://joblib.readthedocs.io/en/stable/

Our preprocessing pipeline consists of the following sequential steps:

1. **Text Cleaning**
   - Remove HTML artifacts (e.g., 'amp;')
   - Remove URLs and hyperlinks
   - Remove hashtag symbols
   - Normalize multiple spaces to single spaces
2. **Text Normalization**
   - Convert all text to lowercase
   - Remove punctuation marks
   - Remove numerical values
3. **Linguistic Processing**
   - Tokenize text into individual words using NLTK's word tokenizer
   - Remove common English stop words (e.g., "the", "is", "at")
   - Apply Porter stemming to reduce words to their root form

The processed text is then reconstructed by joining the processed tokens with spaces. This standardized format helps reduce noise in the data and ensures consistent treatment of linguistic variations, while maintaining the essential semantic information needed for political leaning classification. The preprocessing is implemented using the NLTK[3] library, which provides tools for natural language processing tasks.

For efficiency, we process the larger training datasets using pandas' map function for parallel processing, while smaller validation and test sets are processed sequentially. The processed datasets are then compressed and stored for subsequent model training and evaluation phases.

After preprocessing, we made some additional adjustments to the datasets, such as adding a column to represent the political leaning label as an index instead of text. These steps were necessary for the algorithm to work properly. However, they did not affect the normalization of the data and are therefore not discussed in further detail here.

5.3. Training Logistic Regression Baseline

Before we start with the dataset preparation for our logistic regression baseline, we look at what logistic regression is. Logistic regression is a fundamental statistical model widely used for binary and multi-class classification tasks [Hosmer Jr et al., 2013]. It models the probability of a categorical dependent variable by applying the logistic function to a linear combination of input features. This makes it suitable for text classification tasks

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

which have the goal to assign a discrete label to documents. Although logistic regression has a relatively simple setup compared to modern deep learning approaches, it is still widely used as a baseline model due to its interpretability, computational efficiency, and its capability to process large sets of sparse features, which are typical in text analysis.

5.3.1. Dataset Preparation for Logistic Regression

Before training the logistic regression baseline model, we needed to prepare our datasets by converting the categorical political leanings into numerical labels. The same was done for the preprocessed data as well. This conversion is necessary because machine learning models require numerical input. We implemented a label encoding scheme where:

- RIGHT $\rightarrow$ 0
- LEFT $\rightarrow$ 1
- CENTER $\rightarrow$ 2
- UNDEFINED $\rightarrow$ 3

This encoding was applied to different dataset splits:

- Training set: Using the outlet-based political leanings from POLUSA
- Validation and test sets: Using our manually annotated ratings

5.3.2. Training the Logistic Regression Model

For our baseline model, we implemented a logistic regression classifier using scikit-learn[4]. The training process involved several key steps:

1. **Data Preparation**
   - Loaded the preprocessed training data, excluding articles labeled as "UNDEFINED"
   - Combined headlines and body text into a single corpus
   - Handled missing values by replacing them with empty strings
2. **Feature Extraction**
   - Utilized CountVectorizer (2.1) to transform text into numerical features
   - Removed English stop words to reduce noise
   - Experimented with different max_features values (ultimately settling on 2000) to optimize the model's performance
3. **Model Training and Evaluation**
   - Trained the logistic regression model on the vectorized training data
   - Evaluated performance on both validation and test sets
   - Computed accuracy scores to establish baseline performance metrics

This baseline implementation provides a standard point of comparison for our weak supervision approach, using traditional supervised learning techniques with bag-of-words features. We compare the results in the next chapter.

[4] https://scikit-learn.org/1.2/modules/generated/sklearn.linear$_m$*odel.LogisticRegression.html*

5.4. Fine-Tuning DistilBERT

To establish the optimal configuration for our weak supervision approach, we first conducted several fine-tuning experiments with DistilBERT to find the best possible configuration to move forward with. The fine-tuning process involved systematic exploration of various hyperparameters and training configurations:

1. **Hyperparameter Optimization**
   - Batch sizes: Tested values of 4, 8, 16 and 32
   - Learning rates: Experimented with 5e-05, 3e-05, 2e-05, 1e-05
   - Weight decay: Evaluated different values (1e-4, 1e-3, 1e-2, 1e-1, 2e-1, 5e-4)
   - Number of epochs: Tested 2, 3, 4, 5 epochs
2. **Training Strategy**
   - Initially trained on a smaller subset to quickly identify promising configurations
   - Implemented early stopping to prevent overfitting
   - Used AdamW optimizer with CrossEntropyLoss
   - Tracked both training and validation metrics throughout fine-tuning
3. **Performance Monitoring**
   - Measured accuracy, precision, recall, and F1-scores
   - Tracked performance separately for each political leaning category
   - Saved best-performing model based on validation accuracy

This systematic approach to fine-tuning helped establish the optimal configuration for subsequent experiments with our weak supervision methodology. The specific performance results and optimal parameters will be discussed in detail in the Experimental Results chapter 6, as well as the fined-tuned test results of this baseline.

5.5. Cross-Validation Setup

In this study, the cross-validation setup is of big importance, as it defines how we can use this method to fine-tune the weighting of labels to generate the best possible performance, aiming to surpass the baseline model.

Our aim is to reduce label noise by assigning a confidence weight to each label. The weight which will be applied to the labels in computed during the the cross validation. In the final step, we use a similar algorithm as the one of the baseline model, but we apply the computed weights to each label. These weights are derived from the validation score obtained for each batch of labels during the k-fold cross-validation. For example, if a batch of 4 labels is used for validation and achieves a validation score of 75%, we adjust the weight of these labels. Starting with an initial weight of 100%, we multiply it by the

75% validation score, reducing their influence during the final training and validation process to 75%.

Our initial hypothesis was that using a smaller k-fold setup would enable more precise weighting of the training data. To test this, we implemented a k-fold cross-validation approach with 16,963 folds, calculated by dividing the training dataset of 271,415 articles into batches of 16. This setup would assign a unique weight to each batch, improving the model's ability to differentiate between reliable and unreliable labels during training.

However, we could not move forward with this approach as the computational time required was too much. Each fold took approximately one hour to run, making the total training time excessively long and impractical for our experimental needs. Realizing that we need a more feasible solution, we pivoted to alternative cross-validation strategies that balanced precision and computational efficiency.

Consequently, we adopted 5 different cross-validation setups which will be further explained in the following subsections.

5.5.1. Many K-Folds

To align with our initial hypothesis, we used a many k-fold approach to ensure the training process remained manageable. Knowing that training a single fold would take approximately one hour, we calculated the smallest fold size necessary to complete training within a reasonable timeframe. With a target of around training 5 days, we settled on 132 folds, which corresponds to dividing the dataset size of 271,415 by 2048. While the batch size remained at 16, all 2048 labels in the validation fold were assigned the same weight, simplifying the process while still adhering to our methodology.

5.5.2. Few K-Folds

The few k-fold approach addresses the issue of generating weights to reduce the impact of noisy labels in a different way. In this method, we use significantly fewer folds —just 12 in total— which are randomly split. This cross-validation process is repeated 9 times, each time using a different random seed to create a unique data split. After each run, we record the validation scores. Finally, we calculate the average validation score for each label across these 9 runs and use this average as the weight assigned to the labels.

5.5.3. Expressive K-Folds

Since each k-fold contains a relatively large number of labels, we assume that the resulting weights will not differ significantly. To make sure that we have also weights that differ significantly, we designed an additional experiment to amplify the difference of the weights. In this experiment, we take the weights computed from both the many k-fold (see Section 5.5.1) and the few k-fold (see Section 5.5.2) setups and proportionally scale them to make the weights more distinct. The minimum weight thus be at least 50% lower than the maximum weight. To achieve this, we apply a weight adjustment calculation as described in formula 5.1.

$$w_{adjusted} = w_{original} - \frac{w_{max} - w_{original}}{w_{max} - w_{min}} \cdot 0.5, \quad \text{where } w_{max} > 0.5 \tag{5.1}$$

5.5.4. Extremely Expressive K-Folds

To take this a step further, we apply a similar approach as described in Section 5.5.3. However, in this case, we adjust the weights so that the smallest computed weight is reduced to 0%, while the largest weight is set to 100%. All other weights are scaled proportionally within this range. This adjustment is carried out using the calculation outlined in formula 5.2.

$$w_{normalized} = \frac{w_{original} - w_{min}}{w_{max} - w_{min}} \tag{5.2}$$

5.5.5. Leave-One-Out

In the final experiment, we used a slightly different method compared to the previously described approaches. In our Leave-One-Out cross-validation setup, we excluded all articles from a single news outlet to serve as the validation set. This ensured that the training process did not involve any articles from the same news outlet as those in the validation set. Similar to the earlier methods, we saved the validation scores as weights for the corresponding labels. As we are having 13 different news paper outlets in our dataset, we can expect 13 different weights to apply.

6. Experimental Results

This chapter presents and analyzes the results of our experiments in predicting political leanings of newspaper articles using weak supervision and cross-validation techniques. We evaluate the effectiveness of our various approaches by comparing them against both the logistic regression baseline and the fine-tuned DistilBERT model without weight adjustments.

Our experimental framework tested five distinct cross-validation strategies for weight computation:

- Many k-folds (132 folds)
- Few k-folds (12 folds, repeated 9 times)
- Expressive k-folds (with 50% adjustment factor on weight)
- Extremely expressive k-folds (0-100% weight range)
- Leave-one-out cross-validation (by news outlet)

For each experiment, we evaluate performance using multiple metrics including accuracy, F1-score, precision, and recall. These metrics provide detailed infromation about how well each model performs in identifying political leanings across different categories (left, right, and center). Particularly important is the model's ability to identify cases where the outlet-based labels differ from the actual article content, as identified during our manual annotation process.

The following sections present our findings in a systematic manner, beginning with baseline performance metrics and progressing through each cross-validation strategy. We pay special attention to how different weight computation methods affect the model's performance considering the noisy labels in the dataset. The results will help us understand which approach best addresses the topic of political leaning prediction or is the most promising for future work.

6.1. Overview of Results

Our experiments revealed significant variations in performance across different model configurations and approaches. Given the imbalanced nature of our test dataset (Right: 42 samples, Left: 85 samples, Center: 76 samples), we report weighted averages for all metrics (F1-score, precision, and recall) to account for the different class distributions. We provide an overview over all our reported metrics in the table 6.1.

Table 6.1.: Comparison of Model Performance Across Different Approaches

Model Setup	**Accuracy**	**F1-Score**	**Precision**	**Recall**
Logistic Regression	41.38%	35.37%	35.44%	41.38%
DistilBERT Baseline	54.68%	54.76%	55.49%	54.68%
Many k-folds	54.19%	54.28%	54.73%	54.19%
Few k-folds	54.19%	54.33%	54.90%	54.19%
Many k-folds Expressive	55.17%	55.14%	55.56%	55.17%
Few k-folds Expressive	55.67%	55.71%	56.06%	55.67%
Many k-folds Extremely Expressive	54.68%	54.69%	54.84%	54.68%
Few k-folds Extremely Expressive	55.67%	55.73%	55.99%	55.67%
Leave-One-Out	47.78%	47.26%	47.69%	47.78%

The logistic regression baseline achieved moderate results with a weighted accuracy of 41.38% and a weighted F1-score of 35.37%, establishing a minimum performance threshold for our more sophisticated approaches.

The fine-tuned DistilBERT baseline showed substantial improvement over the logistic regression, reaching a weighted accuracy of 54.68% and a weighted F1-score of 54.76%. This improvement demonstrates the advantages of using pre-trained language models for political leaning classification, as they can better capture the nuanced language patterns in news articles.

Our weak supervision approaches with different cross-validation strategies showed varying degrees of success:

- The many k-folds approach (132 folds) performed similarly to the DistilBERT baseline, with a weighted accuracy of 54.19% and a weighted F1-score of 54.28%.
- Few k-folds (12 folds, repeated 9 times) achieved comparable results with 54.19% weighted accuracy and a slightly higher weighted F1-score of 54.33%.
- The expressive k-folds variants showed the most promising results:
  - Many k-folds expressive achieved 55.17% weighted accuracy and 55.14% weighted F1-score.
  - Few k-folds expressive performed best overall, reaching 55.67% weighted accuracy and 55.71% weighted F1-score.
  - The extremely expressive variants showed similar improvements.
- The leave-one-out approach performed notably lower at 47.78% weighted accuracy. Nevertheless, it is still better than the logistic regression baseline.

Interestingly, the expressive variants of both many and few k-folds consistently outperformed their standard counterparts, suggesting that amplifying the differences between weights helps the model better distinguish between reliable and unreliable labels. The lower performance of the leave-one-out approach indicates that generalizing across different news outlets remains challenging, possibly due to outlet-specific writing styles and editorial policies.

These initial results suggest that while weak supervision can improve performance over standard approaches, the method of weight computation and adjustment can further improve the performance of the final model. The following sections will analyze these results in more detail, examining the specific strengths and weaknesses of each approach.

6.2. Comparative Analysis

6.2.1. Baseline Performance

The logistic regression baseline, despite its simplicity, gave us helpful insights into the challenges of political leaning classification. Its relatively low performance (41.38% accuracy) highlights the complexity of the task and demonstrates the limitations of simpler machine learning approaches. This baseline performance helped establish a minimum threshold against which to compare our more sophisticated methods.

The DistilBERT baseline demonstrated the benefit of a pre-trained language model with a significant improvement compared to the logistic regression baseline. With its 54.68% accuracy representing a 13.3% increase over the logistic regression. This substantial improvement can be attributed to DistilBERT's ability to understand context and capture subtle linguistic patterns. The model's strong performance validates our choice of DistilBERT as the foundation for our weak supervision approaches.

6.2.2. Impact of Cross-Validation Strategies

The various cross-validation strategies we tested showed interesting patterns in how weight computation affects model performance. The standard k-fold approaches (both many and few) performed similarly to the DistilBERT baseline, suggesting that simple weight assignment based on validation performance alone provides limited benefits. This is especially noticeable when investigating the narrow range of computed weights. The many k-folds approach produced weights between 87.69% and 91.48%, while the few k-folds approach resulted in an even tighter range of 88.54% to 89.66%. These small differences between weights likely explain why these approaches showed no improvement over the baseline.

Nevertheless, the expressive variants showed notable improvements. The few k-folds expressive experiment resulted in the best overall performance with 55.67% accuracy and 55.71% F1-score. This success indicates that making sure to have reasonable weight differences helps the model to better distinguish between reliable and unreliable labels. The improvement was consistent across both many and few k-folds setups. This shows that the expressiveness of weights matters to the models performance.

The extremely expressive variants showing similar improvements to their regular expressive counterparts. Therefore, this experiment did not add many additional benefits. There might be a sweet spot in weight adjustment, where too extreme differences between weights do not enhance the models accuracy anymore or even could potentially limit the model's learning capacity.

6.2.3. Weight Adjustment Effects

The impact of the weight adjustment strategies proved to be particularly interesting. The expressive variants, which ensured a minimum 50% difference between maximum and minimum weights, consistently outperformed their standard counterparts. This improvement suggests that creating more pronounced differences between reliable and unreliable labels could help the model to learn more effectively.

The leave-one-out approach's lower performance (47.78% accuracy) reveals the challenges of generalizing across different news outlets. The low performance of this experiment could potentially be traced by the imbalance of the dataset regarding the number of articles per outlet.

Precision and recall metrics across all approaches showed similar patterns to the accuracy scores, with the expressive variants maintaining better balanced performance across all political categories. This consistent improvement across metrics suggests that our weight adjustment strategies enhanced the model's ability to distinguish between political leanings, rather than just optimizing for a particular metric.

These comparative results demonstrate that while weak supervision can improve political leaning classification, the method of implementing and adjusting the weights can be of benefit. The success of the expressive variants indicate that a appropriate method and the right balance in weight adjustment is beneficial to improve the overall performance.

6.3. Key Findings

Our experimental results reveal several important findings about using weight adjustments in weak supervision for political leaning classification in news articles. Firstly, the substantial performance gap between the logistic regression baseline (41.38% accuracy) and the DistilBERT baseline (54.68% accuracy) demonstrates the value of using pre-trained language models for this complex classification task.

The most important key result is that weak supervision with appropriate weight adjustments can further improve classification performance. The few k-folds expressive experiment achieved the best results with 55.67% accuracy, surpassing both baseline models. Although, this improvement is not significantly higher in absolute terms, it represents a meaningful progress.

The narrow range of weights produced by standard cross-validation approaches (between 87% and 91%) proved to be insufficient to effectively distinguish between reliable and unreliable labels. However, our expressive variants, which amplified these differences, consistently showed better performance. This suggests that creating more pronounced weight differences helps the model better handle noisy labels in the training data.

For the leave-one-out approach, the computed weights ranged from 0.0 to 1.0, which is actually indicating a bigger difference. However, the overall test accuracy was likely affected by the large imbalance in the number of newspaper articles per outlet, which made it difficult to generalize effectively across different publications. Therefore, the model also had a lower performance (47.78% accuracy) when testing on completely unseen news outlets. This could additionally also indicates that political language may vary significantly across different publications.

These findings indicate that we can further enhance weak supervision methods by creating and adjusting label weights. The optimal approach appears to be one that creates meaningful differences between weights while avoiding extreme adjustments that might oversimplify the learning task.

Our results with the k-folds approaches, combined with the success of using expressive weight adjustments, support the idea that our initial hypothesis about smaller fold sizes may have some validity. The improvements observed when emphasizing weight differences suggest that using more precise, batch-level weights (as originally planned with 16,963 folds) could lead to outcomes that are even more promising. However, due to the computational limitations of our current setup, we were unable to fully test this hypothesis. Therefore, this could be an interesting direction to explore for future research with access to greater computational resources.

7. Discussion and Conclusion

7.1. Discussion

In this research we explored how we can utilize the results of cross-validation in a weakly supervised setup to enhance the prediction of the political leaning of newspaper articles. Our experiments showed that using pre-trained language models like DistilBERT improves model performance significantly over traditional methods like logistic regression. This matches with findings in the literature which show that BERT-based models have a good understanding of complex language patterns.

One of the main findings is the additional benefit of weight adjustment in weak supervision. The expressive k-folds setup, consistently outperformed their standard counterparts. More pronounced distinctions between reliable and unreliable labels can consequently help the model to learn more effectively.

The leave-one-out approach did not perform as well as expected, showing how hard it is to apply the model to different news outlets. This could be due to the imbalance of the dataset or even due to writing policies in each entity, so we decided not to explore this approach further.

7.2. Conclusion

In conclusion, the research conducted in this work has shown that weak supervision combined with effective weight adjustment strategies, can improve the performance of political leaning classification models. Using pre-trained language models like DistilBERT can be of big benefit, when conducting similar experiments. Our work demonstrate the potential of expressive weight adjustments to further improve prediction results.

Overall, our findings during this research on political leaning prediction contribute to the research on this topic and show the benefits of using of combining weak supervision with advanced NLP techniques. By further improving these methods, we can better understand political language and make automated classification systems more accurate.

8. Limitations and Future Work

8.1. Limitations

While our study has showed which potential weak supervision and cross-validation techniques in political leaning classification can have, several limitations must be acknowledged. Firstly, due to limited computing power, we couldn't fully explore the option of extremely fine-grained k-fold cross-validation. This limitation affected our ability to test the initial hypothesis of using 16,963 folds for more precise weights.

Another challenge was the high imbalance in our dataset to conduct a experiment across news outlets. Our leave-one-out approach could be more promising if the data is more balanced and the applied weights are therefore more meaningful.

8.2. Future Work

A particularly promising direction for future research would be to test our biggest limitation during this research, which is the initial hypothesis of using extremely fine-grained k-fold cross-validation (with 16,963 folds) to create more precise weights per label. Since our results with many k-folds and expressive weight adjustments suggest that using more detailed weights might improve performance, this could be a interesting direction to explore. Nevertheless, it would also require significant computational resources but could provide meaningful finding for the optimal granularity of weak supervision weights.

Looking into different ways to adjust weights and improving the expressive variants could lead to better performance. Political stance detection but also other related NLP topics could benefit by continueing to improve these methods and therefore buiding more accurate prediction models.

List of Tables

List of Figures

Bibliography

[Abercrombie and Batista-Navarro, 2020] Abercrombie, G. and Batista-Navarro, R. T. (2020). Parlvote: A corpus for sentiment analysis of political debates. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5073–5078.

[Bahdanau, 2014] Bahdanau, D. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

[Berrar et al., 2019] Berrar, D. et al. (2019). Cross-validation.

[Cho, 2014] Cho, K. (2014). On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

[Devlin, 2018] Devlin, J. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[Gebhard and Hamborg, 2020] Gebhard, L. and Hamborg, F. (2020). The polusa dataset: 0.9 m political news articles balanced by time and outlet popularity. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020*, pages 467–468.

[Grail et al., 2021] Grail, Q., Perez, J., and Gaussier, E. (2021). Globalizing BERT-based transformer architectures for long document summarization. In Merlo, P., Tiedemann, J., and Tsarfaty, R., editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1792–1810, Online. Association for Computational Linguistics.

[Guess et al., 2020] Guess, A. M., Nyhan, B., and Reifler, J. (2020). Exposure to untrustworthy websites in the 2016 us election. *Nature human behaviour*, 4(5):472–480.

[Gururangan et al., 2020] Gururangan, S., Marasović, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., and Smith, N. A. (2020). Don't stop pretraining: Adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*.

[Higuchi et al., 2022] Higuchi, Y., Yan, B., Arora, S., Ogawa, T., Kobayashi, T., and Watanabe, S. (2022). BERT meets CTC: New formulation of end-to-end speech recognition with pre-trained masked language model. In Goldberg, Y., Kozareva, Z., and Zhang, Y., editors, *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5486–5503, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

[Hosmer Jr et al., 2013] Hosmer Jr, D. W., Lemeshow, S., and Sturdivant, R. X. (2013). *Applied logistic regression*. John Wiley & Sons.

[Iyyer et al., 2014] Iyyer, M., Enns, P., Boyd-Graber, J., and Resnik, P. (2014). Political ideology detection using recursive neural networks. In Toutanova, K. and Wu, H., editors, *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1113–1122, Baltimore, Maryland. Association for Computational Linguistics.

[Karamanolakis et al., 2021] Karamanolakis, G., Mukherjee, S., Zheng, G., and Awadallah, A. H. (2021). Self-training with weak supervision. *arXiv preprint arXiv:2104.05514*.

[Kohavi et al., 1995] Kohavi, R. et al. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada.

[Kulkarni et al., 2018] Kulkarni, V., Ye, J., Skiena, S., and Wang, W. Y. (2018). Multi-view models for political ideology detection of news articles. In Riloff, E., Chiang, D., Hockenmaier, J., and Tsujii, J., editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3518–3527, Brussels, Belgium. Association for Computational Linguistics.

[Laver and Garry, 2000] Laver, M. and Garry, J. (2000). Estimating policy positions from political texts. *American Journal of Political Science*, pages 619–634.

[Liu et al., 2022] Liu, Y., Zhang, X. F., Wegsman, D., Beauchamp, N., and Wang, L. (2022). Politics: Pretraining with same-story article comparison for ideology prediction and stance detection. *arXiv preprint arXiv:2205.00619*.

[Manning, 1999] Manning, C. (1999). *Foundations of statistical natural language processing*. The MIT Press.

[Mikolov et al., 2013] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

[Mullins, 1972] Mullins, W. A. (1972). On the concept of ideology in political science. *American Political Science Review*, 66(2):498–510.

[Nguyen et al., 2020] Nguyen, D. Q., Vu, T., and Tuan Nguyen, A. (2020). BERTweet: A pre-trained language model for English tweets. In Liu, Q. and Schlangen, D., editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14, Online. Association for Computational Linguistics.

[Park et al., 2023] Park, S., Kim, K., and Lee, J. (2023). Cross-task knowledge transfer for extremely weakly supervised text classification. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5329–5341, Toronto, Canada. Association for Computational Linguistics.

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

[Peterson and Allamong, 2022] Peterson, E. and Allamong, M. B. (2022). The influence of unknown media on public opinion: Evidence from local and foreign news sources. *American Political Science Review*, 116(2):719–733.

[Preoţiuc-Pietro et al., 2017] Preoţiuc-Pietro, D., Liu, Y., Hopkins, D., and Ungar, L. (2017). Beyond binary labels: Political ideology prediction of Twitter users. In Barzilay, R. and Kan, M.-Y., editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 729–740, Vancouver, Canada. Association for Computational Linguistics.

[Ratner et al., 2017] Ratner, A., Bach, S. H., Ehrenberg, H., Fries, J., Wu, S., and Ré, C. (2017). Snorkel: Rapid training data creation with weak supervision. In *Proceedings of the VLDB endowment. International conference on very large data bases*, volume 11, page 269. NIH Public Access.

[Ratner et al., 2016] Ratner, A. J., De Sa, C. M., Wu, S., Selsam, D., and Ré, C. (2016). Data programming: Creating large training sets, quickly. *Advances in neural information processing systems*, 29.

[SANH et al., 2019] SANH, V., DEBUT, L., CHAUMOND, J., WOLF, T., and Face, H. (2019). Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

[Sawhney et al., 2020] Sawhney, R., Wadhwa, A., Agarwal, S., and Shah, R. (2020). Gpols: A contextual graph-based language model for analyzing parliamentary debates and political cohesion. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4847–4859.

[Sedova and Roth, 2023] Sedova, A. and Roth, B. (2023). ULF: Unsupervised labeling function correction using cross-validation for weak supervision. In Bouamor, H., Pino, J., and Bali, K., editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4162–4176, Singapore. Association for Computational Linguistics.

[Sinno et al., 2022] Sinno, B., Oviedo, B., Atwell, K., Alikhani, M., and Li, J. J. (2022). Political ideology and polarization: A multi-dimensional approach. In Carpuat, M., de Marneffe, M.-C., and Meza Ruiz, I. V., editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 231–243, Seattle, United States. Association for Computational Linguistics.

[Stefanov et al., 2020] Stefanov, P., Darwish, K., Atanasov, A., and Nakov, P. (2020). Predicting the topical stance and political leaning of media using tweets. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 527–537.

[Vaswani, 2017] Vaswani, A. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*.

[Wolf et al., 2020] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al. (2020). Transformers: State-of-the-art natural language processing. *EMNLP 2020*, page 38.

[Xu et al., 2019] Xu, H., Liu, B., Shu, L., and Yu, P. (2019). BERT post-training for review reading comprehension and aspect-based sentiment analysis. In Burstein, J., Doran, C., and Solorio, T., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2324–2335, Minneapolis, Minnesota. Association for Computational Linguistics.

[Zhang et al., 2021] Zhang, J., Yu, Y., Li, Y., Wang, Y., Yang, Y., Yang, M., and Ratner, A. (2021). Wrench: A comprehensive benchmark for weak supervision. *arXiv preprint arXiv:2109.11377*.

A. Appendix

GitHub Repository: https://github.com/benstah/political_leaning_prediction