Using Publisher Partisanship for Partisan News Detection

A Comparison of Performance between Annotation Levels

Chia-Lun Yeh

Using Publisher Partisanship for Partisan News Detection

A Comparison of Performance between Annotation Levels

by

Chia-Lun Yeh

in partial fulfilment of the requirements for the degree of

Master of Science
in Computer Science
track Data Science and Technology

at the Delft University of Technology
to be defended publicly on August 23, 2019 at 2:30 PM.

Supervisors:	Assistant Prof. dr. N. Tintarev
	Dr. A. Schuth
	Dr. B. Loni
Thesis committee:	Prof. dr. ir. G. J. Houben
	Assistant Prof. dr. N. Tintarev
	Assistant Prof. dr. H. Wang

Abstract

News is the main source of information about events in our neighborhood and around the globe. In an era of digital news, where sources vary in quality and news spreads fast, it is critical to understand what is being consumed by the public. Partisan news is news that favors certain political parties or ideologies. It is undesirable because news organization should aim for an objective and balanced reporting. An automatic system that can classify news to be partisan or non-partisan is thus desired. Such a system (partisan news detector) requires a sufficient amount of data to learn the pattern of partisanship in news articles. However, these labels are expensive to collect since manual inspection of each article is required.

Inferring partisanship labels from the partisanship of publishers is another approach that has been used in previous research. By treating all articles by partisan publishers to be partisan news and those by non-partisan publishers to be non-partisan, it is easy to collect a large number of labeled articles. This way of deciding labels is noisy, making it more difficult for a detector to learn.

This thesis compared the performance of using publisher-level labels and article-level labels for partisan news detection. The detector was designed as a binary classifier. We compared the difference in performance across several feature sets to ensure the observed difference was due to the annotation level, not the choice of specific classifiers. The experiments were performed on two datasets of different properties to ensure the generalizability of the results. We found that classifiers trained with publisher-level labels have higher recall but lower F1-score compared to classifiers trained with article-level labels. We also observed that the classifiers overfit on publishers but reducing the overfitting with feature selection did not improve the performance. Comparing the performance difference between the two datasets, we concluded that an important factor that determines the performance achievable by the publisher-level labels is the quality of publishers that are included in the dataset. This is valuable for future dataset collection.

Our work provides a benchmark performance of publisher-level labels, which can be used as baselines for future research that investigate other methodologies to utilize the noisy labels. We also compared the performance between the two levels of annotation and concluded that partisan news detectors trained with article-level labels are more practical to be used in a fully-automated system since they have on average 10% higher F1-scores than those trained with publisher-level labels. However, the high recall of the latter makes them applicable in use cases where high recall is desired.

Preface

This 10-month journey started last summer when I first approached Nava, my dear supervisor from TU Delft. I remember clearly that afternoon when we sat by the canal in front of the cat cafe and talked about potential topics. I was nervous, anxious, and unsure of what I wanted to do. A few weeks later, I approached Anne, my supervisor at DPG Media. We agreed upon a broad topic of finding viewpoints in news to crack the filter bubble. So began the bumpy ride! My supervisors, and my daily advisor, Babak, have been the most critical yet supportive people throughout my work. I want to thank Nava for her diverse perspectives and high-level questions that forced me to rethink my methodologies and narrow down my directions. I want to thank Anne for always being upbeat about new ideas, for making the dataset collection happen, and for his insightful interpretations. I also want to thank Babak for brain-storming with me, checking my weekly progress, and giving invaluable feedback throughout each phase of my thesis. The NPS squad in DPG Media has been a positive influence due to their effective way of working and open-mindedness, which I sincerely admire and appreciate.

Besides, I was lucky to have a group of smart and fun friends at TU Delft that contributed to my well-being during difficult times. My friends from the faculty, Ning, Barbara, Nirmal, Ania, Ombretta, FengLu, Po-Shin, Ziyu, Aishwarya, Dhruv, and Kanav. I loved those moments we spent during coffee breaks and lunch at VMB, as well as encounters when we updated each other and felt that we were all in this together. I especially want to thank Barbara for helping me choose the topic, and reaching out when I felt most confused. My most adorable Taiwanese girls, Flora, Kelly, and Rose. Thank you for being there, involving me in excursions, dinner gatherings, and saying yes to moments when I needed you. Ramon, you are one hot spice in my life. Your frank judgments on various things guided me at times most needed. Sebastian, thank you for the thesis tips and for being there during the stressful process of making professional choices for my life. Finally, CY, thank you for giving me constructive feedback on my conclusions, for getting me out of my work and for the endless talks covering all aspects of life.

My constant drive and happiness came from some of the most important people. My best friend back home, Michael, thank you for patiently listening to my random thoughts and helping me structure them, for calming me down and accompanying me during interview preparations and house-searching, and for simply understanding me. My dearest mom and dad, thank you for supporting me to study in the Netherlands, for not putting stress on me, and for trusting me in all the decisions I made along the way. Thank you all.

Chia-Lun Yeh
Delft, August 2019

Contents

List of Figures

List of Tables

1

INTRODUCTION

At DPG Media[1], each news article that is published goes through a pipeline that decides whether it should be pushed as a notification to a user of the online news application [3]. The decision of which article is relevant to which user is often based on the topics and locations that a user is interested in, or based on the similarity of word usage with previous articles that the user has read.

Besides feeding readers with news articles that are interesting to them, DPG Media is also interested in providing articles that better inform them about different perspectives of a news story. For example, if a user strongly supports a political party, he or she might read more articles that are positive about the party. If the pipeline makes decisions based on these previously-read articles, it pushes more articles of similar opinions to the user, possibly strengthening the user's somehow one-sided viewpoints. This way of optimizing personalization algorithms can create "filter bubbles" [4], and influence public opinions [5]. The Pew Research Center reported that the global public wants politically balanced news [6]. In the report, 35 of the 38 nations that were surveyed agreed that it is never acceptable for a news organization to favor one political party over the others when reporting the news. Therefore, it is important to understand the partisanship of news articles to fulfill the public's wish. This can be done manually by editors who select non-partisan articles to be published. However, manual selection is an time-consuming task. It is thus of interest to design a system that automatically finds the partisanship of news articles to help select less partisan news.

Such a system often needs news articles with known partisanship to learn the patterns and linguistics cues that express partisanship. However, similar to manual selection, articles with labeled partisanship are expensive to obtain. As we show in Chapter 2, there are few datasets that exist for the task. On the other hand, the partisanship of news publishers has been studied actively and several resources are available. For example,

[1] A Belgian publishing company that owns national and local news publishing brands in the Netherlands. Formerly de Persgroep: https://www.persgroep.nl/

AllSides[2] and Media Bias/Fact Check (MBFC)[3] are websites that rate the partisanship of news media in the United States (US). Figure 1.1 shows an example of the ratings by AllSides. They categorize publishers into left, lean left, center, lean right, and right. The ratings are derived from several sources, including the public, domain experts and independent research. Since these ratings are mainly based on the evaluation of the news articles that a publisher publishes, we assume that publishers that are rated to be partisan on these websites publish more partisan news articles. By treating all articles by partisan publishers as partisan and all others as non-partisan, we can collect a large number of labeled news articles without reading and rating them individually.

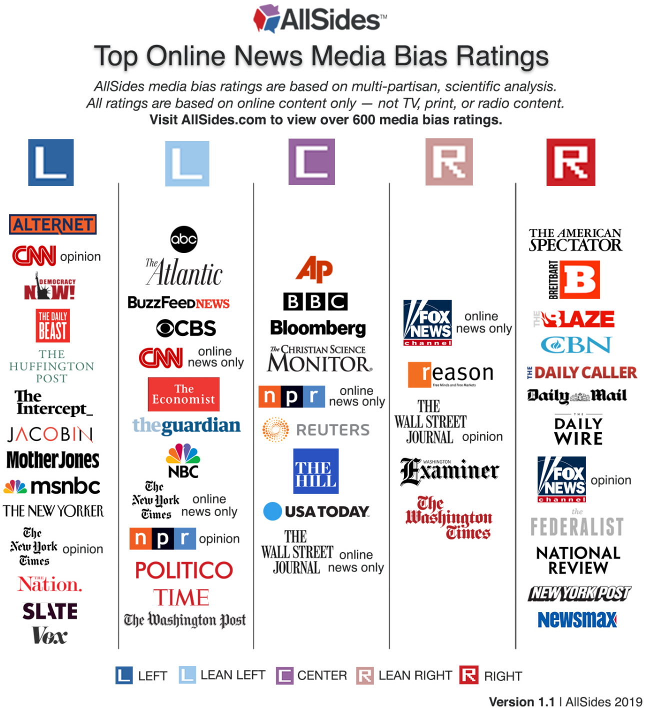


Figure 1.1: Publisher partisanship ratings by AllSides.

Although this way of labeling is faster and cheaper, it also creates a problem of noisy labels as not all articles from a publisher would be of the same partisanship. For example, the US media Breitbart News Network [4] is considered a far-right publisher. Although most of its articles support the right-wing party, some news pieces are non-partisan. In the Netherlands, media are less polarized than those in the US and partisan publishers

[2] https://www.allsides.com/unbiased-balanced-news
[3] https://mediabiasfactcheck.com/
[4] https://www.breitbart.com/

are less consistent in the stances they take on different topics. Moreover, a non-partisan publisher is in general defined to be one that does not predictably show opinions favoring a political party or ideology. If a publisher is labeled as non-partisan because it doesn't show partisanship at all, it is good as its articles can be safely labeled as non-partisan. However, it can also be the case that the publisher leans to different sides of the political spectrum on different topics and favors all of them somewhat equally. In this case, individual articles would be partisan, and we would label them wrongly. A partisan news detector that is based on publisher-level labels thus has to find patterns within some wrong labels. This is expected to have an impact on how well the system can predict partisanship. What we are interested in is understanding how applicable this paradigm is and how much it affects the performance compared to a detector that learns from article-level labels, which are labeled manually per article. We refer the two labeling strategies as different "levels of annotation" because we consider labeling articles with publisher partisanship as a high-level annotation scheme and labeling articles one-by-one as a more fine-grained, low-level one.

The objective of this thesis is to implement and evaluate partisan news detectors to investigate how the level of annotation affects performance. We address the following research questions.

1. What is the benchmark performance of predicting partisanship of articles with publisher-level labels?

2. What is the performance of predicting partisanship of articles with article-level labels compared to publisher-level labels?

1.1. Research gap and contribution

In Chapter 2, we surveyed methods and datasets for partisan news detection. From the survey, we concluded that a research gap in the field is an understanding of how publisher-level annotation would affect the performance of article-level prediction. To address this gap, we made the following contributions.

1. We benchmarked the performance of partisan news detectors trained with publisher-level labels across several features. Then we compared it with the performance of detectors trained with article-level labels to study the performance difference caused by annotation levels.

2. We collected a Dutch dataset for partisan news detection that includes publisher-level and article-level labels. This allowed us to see how our results on one dataset generalize to another one with different properties.

3. We empirically analyzed the cause of the performance gap between the two levels of annotation. This included the overfit on publishers and three dataset properties, training size, the number of publishers, and the ratio of left and right publishers.

1.2. THESIS OUTLINE

The thesis is structured as follows. Chapter 2 gives a background study on approaches to determine partisanship of publishers, detect partisanship in text, as well as datasets with labeled partisanship. Chapter 3 builds on the background and details the implementation of our methodology. In Chapter 4, we explain the collection and annotation process of the datasets that were used in our study. Chapter 5 addresses the first research question by benchmarking the performance of a partisan news detector that learns from publisher-level labels. We further analyze the performance and attempt to improve in Chapter 6. In Chapter 7, we address the second research question by using article-level labels to investigate the improvement that can be made. We conclude our work, explain the applicability of such a system, and point out future work in Chapter 8.

2

Background Literature

In Chapter 1, we formulated our research goal as **implementing and evaluating** a **partisan news detector** to investigate **how the level of annotation affects performance**. To reach the goal, we break down the statement and studied different pieces of it to further scope our work. This chapter presents the literature and positions our work in this broad research area.

To build a partisan news detector, we first study what partisan news is. In section 2.1, we define what partisanship means in our context and how it is presented in news media. Then, we studied the approaches that have been proposed to implement the detector. We found little work addressing the exact problem but several works addressing similar tasks. We briefly introduce these related tasks in section 2.2 and dive into the approaches in section 2.3. Since detecting partisanship of an article is commonly formulated as a classification problem, we focus on the features and classification models that have been proposed. Then, we investigated the last but arguably most important component of our goal: how annotation level affects the performance, in section 2.4. This was done by researching available datasets, their annotation level, and the achieved performance. In section 2.5, we present the research gaps that were identified.

2.1. Partisanship in news media

To understand how partisanship is expressed in news articles, we first explain what partisanship is. Then, we explain how news media incorporate partisanship in their publications in general.

2.1.1. What is partisanship?

According to Cambridge Dictionary, partisanship is "the quality or action of strongly supporting a person, principle, or political party, often without considering or judging the matter very carefully" [1]. This definition is in line with what we deal with in this thesis. We are interested in knowing whether a news article presents a favoring opinion towards certain political ideologies or political parties. This favoring opinion can be strong or mild. For example, politics in the United States is more bipolar than most countries in Europe. Many extremely polarized news media exist and publish news articles that resemble fake news and propaganda. In this case, the partisanship is strong and intended to influence the public (called hyperpartisan in our text). In the Netherlands, however, politics is less bipolar and news media are less extreme. The partisanship is of less magnitude. The intensity can influence how partisanship is expressed in language, and make it easier or harder to detect.

The "partisanship" that we refer to in the work is largely the same as what the general public refer to when they talk about political bias, political leaning, and political slant. We chose to use partisanship consistently to avoid the overwhelming terminologies that have been used in literature. Note that we are interested in whether the political partisanship exists in an article, not which political ideology is presented. The latter often assumes that the texts are partisan and aims to find the polarity of partisanship, while we don't assume anything on the text and aim to find out whether it is partisan or not. However, we included selected literature about the latter in the following sections because we believe that the linguistic cues and datasets are highly valuable for our task as well.

2.1.2. How is partisanship presented in news media?

In news media and journalism, the partisanship can be presented in several forms. For example, the stories selected to be covered, the stories that are put in the front page, the reporting language and sentiment used by a journalist, the picture that is selected to go with an article, the person that is being quoted, etc, can all carry partisanship that influence public opinions towards an event and a person.

D'Alessio and Allen [7] suggested three bodies of media bias, which are used commonly by researchers in news partisanship [8–10]. They are selection bias, coverage bias, and statement bias (bias is the same as partisanship but we use bias here to align with their term). We introduce them in the following.

Selection/gatekeeping bias Selection bias occurs when editors and journalists decide which news stories to present to the public and which not. Due to the limited space and time, it is inevitable to choose a portion of stories to cover. The omission of certain events can result in a partial understanding of the full story. Selection bias is difficult to quantify because we do not have information about all the news in the universe to study the amount of selection. Arguably, selection bias is easier to be quantified for news aggregators. A news aggregator collects articles from online newspapers, blogs, etc and presents them to users. In this case, the number of available articles is known, and the articles chosen by an aggregator can be analyzed to study the degree of partisanship.

[1] https://dictionary.cambridge.org/dictionary/english/partisanship

Coverage bias Coverage bias is the preference of giving larger coverage to a certain party, events, or person. Again, due to the limited space, the editors have to decide how much space is used to present a story after it is selected. Sáez-Trumper *et al.* [9] measured coverage bias by the length of articles about a story and by the numbers of mention of people from different political parties. However, it is challenging to argue what a fair coverage distribution is. For a two-party electoral issue, it is sensible to say that the two parties should get equal space and mentions. But for issues with multiple standpoints, this is nontrivial [7].

Statement bias In addition to deciding which stories to present, and how much space is given for the coverage, partisanship also comes from how the story is presented. Statement bias refers to expressing a favorable or unfavorable opinion towards a party, event, or person. This is arguably the most quantifiable bias as we can decide the bias with content analysis on an article or paragraph level. For example, Sáez-Trumper *et al.* [9] computed the sentiment towards each person mentioned in the article to decide the degree of statement bias. By computing the sentiment of each statement, we can classify articles that contain no overtly opinionated statement as non-partisan. Articles that give an equal number of statements favoring different political parties can also be seen as non-partisan as they present balanced perspectives. Since our partisan news detector operates on the article level, partisanship in this work refers to statement bias.

2.2. RELATED TASKS

As Recasens *et al.* [11] pointed out in their work of detecting bias from Wikipedia, *"bias is linked to the lexical and grammatical cues identified by the literature on subjectivity, sentiment, and especially stance or arguing subjectivity"*, several fields of computational linguistics are closely related to detecting non-neutral, non-objective language. We introduce the related tasks of subjectivity analysis, sentiment analysis, and stance detection. In addition, due to the nature of partisanship and media landscape, we introduce two more related tasks, political ideology detection, and fake news detection. In this section, we give a brief explanation of each related field and explain the similarities and differences from our task.

2.2.1. SUBJECTIVITY ANALYSIS

Subjectivity analysis aims to identify whether a given text expresses opinions, evaluations, and emotions (subjective) or factual information (objective). Although it seems at first glance that subjective implies partisan while objective implies non-partisan, there are differences. For example, a subjective sentence like "I think the coffee is great" is an opinion but non-partisan. Partisanship can also be expressed in an objective language by using terms that accord with a party [12].

Subjectivity analysis is mostly done at the sentence or phrase level because a document often contains a mixture of subjective and objective sentences. This is different from our document-level task. Nevertheless, detecting subjectivity is closely related to detecting partisanship in terms of the language that is used. Extracting subjective sentences or words from the text or adding subjectivity as prior information has shown to improve the detection of sentiment and partisanship [13–15].

2.2.2. OPINION MINING AND SENTIMENT ANALYSIS

Opinion mining and sentiment analysis broadly denote the same field of study that analyzes evaluative text to find out people's opinions towards entities such as products, organizations, individuals and issues [16]. Most work has been carried out on explicitly subjective text such as product reviews and blogs. For example, analyzing whether a movie review is positive or negative [17].

Several aspects make it difficult to decide sentiments for news articles [18]. On a high level, we need to decide what sentiment means in news articles. Is an article that reports a tragedy considered negative? What if it is reported in an objective tone? For our purpose, we can formulate the partisan news detection task as detecting the sentiment expressed towards an entity by the author. For example, if we detect a strong sentiment in the article towards a politician, we can safely identify it as a partisan article. However, even with this formulation, it is difficult to apply the approach common in the field. Sentiment analysis often relies on detecting positive and negative terms and expressions, taking into account negations, intensifiers, and sarcasm [17, 19]. Partisan news usually carries emotion words but not necessarily. A news article that supports abortion would rarely use a lot of positive phrases to describe abortion but instead use arguments that support the right of women. Therefore, we expect sentiment words to be helpful features but insufficient to detect partisanship.

2.2.3. STANCE DETECTION

Stance detection aims to identify the relationship between a piece of text and a claim of the text. The most common use case is on online debate platforms [20]. Each topic on the platform contains a claim to be debated about. The aim is to find whether a post or a user agrees, disagrees, or is neutral about the claim [21].

Our task can be formed as a stance detection problem by forming or extracting a claim on the topic of the news article. News articles that show agreeing or disagreeing stances towards the claim are partisan and those that are neutral are non-partisan. However, formulating our problem in this manner makes it more complex as the formulation or extraction of a suitable claim is nontrivial.

2.2.4. POLITICAL IDEOLOGY DETECTION

Political ideology or orientation detection is the task of finding the political ideology expressed in texts such as political blogs and congressional speeches [22–24]. Although how political ideology should be modeled is controversial, it is commonly modeled as liberal, and conservative. Neutral can also be added, which is similar to non-partisan in our case. Some work used more fine-grained classes. Preoţiuc-Pietro *et al.* [25] used seven scales while Sim *et al.* [22] further split left ideology into 4 categories (progressive-left, center-left, etc) and right into 5 categories (far-right, populist-right, religious-right, etc).

The nature of our task is the same as that of political ideology detection if a "neutral/center" class is included. In this case, we treat all other ideology as partisan and the neutral class as non-partisan. The only difference is that we are considering news articles while most work in this field considers political texts.

2.2.5. FAKE NEWS DETECTION

Fake news detection tries to determine the truthfulness of claims in a news article and detect fake news articles [26]. Real fact-checking involves investigating external resources to check whether the reported story is true. However, work has been done from a text analysis perspective by investigating deceptive language. Although partisan news is different from fake news, hyperpartisanship is often difficult to express without distorting part of the truth. Therefore, from a content analysis perspective, the two are commonly studied together. For example, Stein *et al.* [27] showed that writing style features are useful for both hyperpartisanship detection and fake news detection.

We see that detecting partisan news can potentially be formed into other tasks but there are differences. These related fields provide insights into the linguistic cues that are used to express subjectivity, opinions, and stances.

2.3. COMPUTATIONAL METHODS FOR PARTISANSHIP DETECTION

Figure 2.1 shows an overview of the computational methods for partisanship detection. Partisan news detection is mostly approached using text analysis. Some work used other information such as user comments, reader votes, and cites from political blogs to derive article partisanship based on graph analysis [28–30]. These methods require user interactions with the article and can only be done after the article is published, which does not meet our motivation of selecting non-partisan news to be published. We thus chose to use pure text analysis methods.

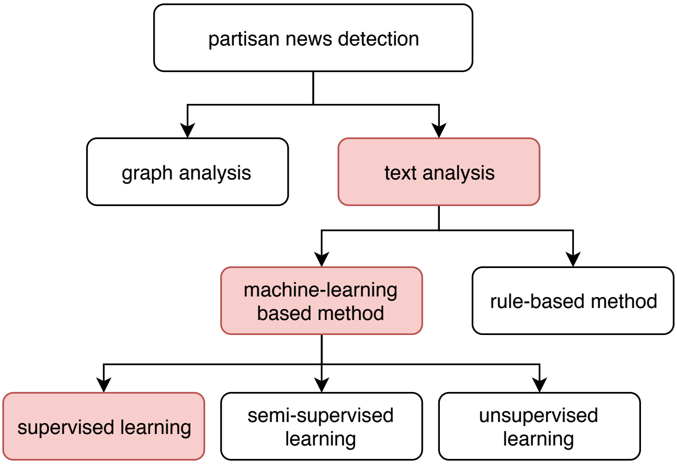


Figure 2.1: Overview of computational methods for partisan news detection.

Computational methods for text analysis can be grouped into rule-based algorithms and machine-learning-based algorithms. Rule-based algorithms define rules to classify texts. For example, to detect partisanship for American news, we can define rules such as "an article that has an equal number of pro-Democratic and pro-Republican statements is non-partisan". However, this is impractical because partisanship is expressed in various forms and contexts. It is thus difficult to construct rules that determine what comprises partisanship. For example, some viewpoints need more statements to support but the high number of favoring statements does not lead to stronger arguments and a more favorable sentiment overall. Also, partisanship might be expressed by the specific use of words in a context, which is not possible to define for each case. Therefore, we decided to use the machine-learning-based approach.

Machine learning is a data-driven approach that finds patterns in existing data and uses the generalized pattern to predict the behaviour of new data. It comes in three forms, supervised, semi-supervised, and unsupervised. The most common approach is to treat partisanship detection as a supervised classification problem [11, 23, 27, 31–35]. We discuss the components of supervised leaning in detail in section 2.3.1. We briefly discuss unsupervised learning in section 2.3.2 to introduce topic models.

2.3.1. SUPERVISED LEARNING

In a supervised classification task, the labels of the data are known. The algorithms can thus learn from the distribution of each class and find a decision boundary that separates the classes. A general text classification pipeline includes three steps, preprocessing, feature extraction, and model training. They are discussed in the following subsections.

PREPROCESSING

Preprocessing is an important step that cleans and prepares the raw data for feature extraction by parsing, tagging, and normalizing.

Tokenization A text is a sequence of characters. Tokenization is the task of parsing it into semantically meaningful pieces, called tokens. Tokens are usually referred to as terms or words. Tokenization is language-specific. For most alphabet-based languages, one can imagine a simple way of segmenting the text by any non-alphanumeric characters or separating by spaces. In practice, specific rules are often designed and applied to increase accuracy.

Removing stop words Some words are extremely common in the language and provide little information in helping us decide which class a document belongs to. These words are called stop words. For example, ’the’ and ’of’ in English. It is common to use a pre-constructed stop word list to remove these words before further processing.

Stemming and lemmatization Stemming and lemmatization are used to reduce morphological variations by turning terms into their base forms. For example, when we want to count how many partisan terms appear in a document based on a word list, if the term "murder" is in the list, we would want it to match "murdered", "murders", "murdering", etc in the document since they only differ due to grammatical reasons. Stemming is a computationally faster but crude approach that removes the end of the term. Lemmatization, on the other hand, analyzes the morphology of terms and returns the base form. For example, the term "meetings" has the stem "meet" and lemma "meeting".

FEATURE EXTRACTION

Converting text into effective feature vectors is a crucial part of the data-driven approach to text classification. Effective features separate the classes in the feature space. Some general NLP features have been shown effective in most text classification tasks, such as word embeddings and PoS-tags, while other features are designed with domain knowledge to extract information that makes sense to the task at hand. In this section, we explain some of the most widely-used features in literature.

Bag-of-Words (BoW) BoW is a term frequency feature of dimension $|V|$, where V is the vocabulary in the whole corpus. Each value in the vector is the frequency that the term occurs in the document. This weighting scheme is referred to as term frequency. We use $tf_{t,d}$ to denote the count of term t in document d.

BoW uses the raw count and treats each term equally important. However, terms that occur in many documents carry less information that discriminates the document from each other. Term frequency-inverse document frequency (tf-idf) is a scaling scheme that scales the raw counts with the inverse of the frequency that the term appears in all documents. Assuming that there are N documents in total, each tf-idf value is computed as

$$tf-idf_{t,d} = tf_{t,d} \times log \frac{N}{\sum_{i=1}^{N} tf_{t,i}} \tag{2.1}$$

This weighting effectively ignores common words.

One disadvantage of this representation is that the order of words is not preserved. A simple document with the sentence "my mom likes the dog" would have the same vector as another one with "the dog likes my mom". This can be remedied to a certain extent by treating a sequence of words as a "phrase" by using n-grams. An n-gram is a sequence of n words. Using the above example, instead of the 5 features ("my", "mom", "likes", "the", "dog"), we now have 6 additional features ("my mom", "mom likes", "likes the", "the dog", "dog likes", "likes my") if we choose n in [1,2]. The two sentences would then have different representations.

BoW and n-grams are effective in many document classification tasks, and often used as a baseline model [36]. Gentzkow and Shapiro [12] used extracted bi-grams (n=2) from US Congressional speech and found that phrases such as "death tax", "tax relief", and "war on terror" are used mostly by Republicans, as opposed to the usage of "estate tax", "tax break", and "war in Iraq" by Democrats. They also found that news publishers choose terms that aligned with the usage of their supported parties, hence the phrases can effectively be used to reveal partisanship of an article.

BoW and n-grams have two main problems.

1. Very high dimension: $|V|$ often has a dimension of tens or hundreds of thousands. The high dimension makes subsequent training of models difficult.

2. No encoding of word semantics: Since each word or phrase is encoded in a separate dimension, they are all orthogonal to each other. We lose the relationship of words. For example, we do not encode any information like "car" is semantically closer to "automobile" than "train" in the vector.

Word embeddings To tackle the aforementioned problems, algorithms were designed to embed each word into a continuous vector space of real numbers. The main idea behind these algorithms is that similar words appear in a similar context. By randomly initializing a vector for each word and optimizing the vectors, we obtain vectors that preserve semantics. Two algorithms are commonly used. The first is word2vec proposed by Mikolov *et al.* [37], using skip-gram or Continuous BOW (CBOW) model. The model predicts a word's context given the word or predicts the word given its context using a neural network. Words that have a large overlap of neighboring words end up with similar vectors. The second is GloVe proposed by Pennington *et al.* [38]. In GloVe, a global co-occurrence matrix is computed that records how often each word co-occurs with every other word in the corpus. The word vectors are then optimized to have large dot products if they co-occur a lot. The word vectors obtained from these models have been shown to preserve some linguistic structures. For example, the following holds:

$$cos(v_{frog}, v_{toad}) > cos(v_{frog}, v_{fish})$$
$$v_{clean} - v_{cleaner} \approx v_{dark} - v_{darker}$$
$$v_{man} - v_{woman} \approx v_{brother} - v_{sister}$$

, where v_w is the word vector for word w and $cos(v_i, v_j)$ is the cosine similarity between vector v_i and v_j The vector dimension is typically between 50 and 300.

To obtain word embeddings, we can either train the embeddings using in-domain corpora or use pre-trained word vectors. Training with in-domain corpora can capture the specific meanings of words in the domain and vocabularies that are not in general corpora. For example, in the medical domain, it's useful to train from scratch to capture specialized medical terms. There are also pre-trained vectors that are used commonly out-of-the-box. These embeddings are trained with large corpora with general texts such as news and web texts. Using these pre-trained vectors are convenient, and ensures that we don't overfit to the corpora we use for training.

Part-of-Speech (PoS) PoS is a categorization of words that have similar grammatical properties, for example, nouns, verbs, etc. PoS has been shown to help predict bias and hyperpartisanship [11, 27]. It is also used commonly in sentiment analysis because sentiment is mainly expressed by the usage of adjectives and adverbs [16].

Lexicon-based features Lexicon-based features use dictionaries with words labeled into specific categories to analyze texts. SentiWordNet [39] attaches positive and negative scores to each WordNet synsets. By accumulating the scores of the words within a text, one can have an idea of the polarity of the text. On the other hand, Harvard General Inquiry (GI) [40] and Linguistic Inquiry and Word Count (LIWC) [41] are dictionaries that assign words to categories related to emotions, social cognition, functions, etc.

Words such as "allied", "conservative", and "candidate" are marked as having clear political characters in GI. These word lists offer more fine-grained classes to understand the semantic and sentiment of the text.

Recasens *et al.* [11] incorporated several word lists (report verbs, factives, entailments) from literature to detect bias in sentences. Stein *et al.* [27] included features derived from related classes of GI to detect hyperpartisanship and fake news. LIWC is widely used in fake news detection [42, 43].

Classification algorithms

After features are extracted from texts, classification models are trained to find a function that maps the feature vectors to the correct class as well as possible. We do not go into mathematical details of how different classification algorithms are formulated because we do not aim to improve upon these algorithms. Instead, we introduce the concept and characteristics of some popular algorithms used in the literature. We summarize their properties in Table 2.1.

Table 2.1: Comparison of commonly used text classification models.

Model	Advantage	Disadvantage
NB	fast need less training data probabilistic output	assume linearly separable assume independent features
SVM	linear or nonlinear (kernel) good with high dimensional features	long training time
LR	fast easy to interpret probabilistic output	mainly used as a linear classifier
RF	good with high dimensional features	hard to interpret
NN	linear or nonlinear minimal preprocessing	need lots of data long training time

Many classification models that have been shown strong in text classification have been used in partisanship detection. The most popular ones include support vector machine (SVM) [31, 42], logistic regression (LR) [11, 43], naive Bayes (NB) [31], and random forest (RF) [27] classifiers. SVM is a maximum-margin classifier that tries to find a decision boundary that is maximally far from any point in the training data. It is naturally a linear classifier by can be transformed into a nonlinear one using kernel functions. SVM often performs well on high-dimensional data and is thus used often in text classification. It is, however, very inefficient to train with large training size. LR uses the logistic function and optimizes the parameters on the training data by maximizing the likelihood of the assumed posterior probability model. LR can be interpreted easily because the learned coefficients of each feature can be used to compute feature importance. NB is a generative classifier that is based on the Bayes theorem and assumes independence between features. Although it is fast and often works well on small datasets, the independence assumption rarely holds and the performance is thus limited. RF is an ensemble of decision trees. By constructing the classifier with multiple trees, each built with a random subset of data samples and features, RF reduces the variance and often performs better than using a single tree.

Neural networks have not been as commonly applied to partisan news detection as other text classification tasks. This is likely due to the lack of data to train the large number of parameters in these models. The only work is that of Kulkarni *et al.* [32]. They used a convolutional neural network (CNN) that was devised for sentence classification [44] to model the title of an article. For the content, a hierarchical attention network (HAN) proposed by Yang *et al.* [45] was used. HAN uses two bidirectional gated recurrent units (GRUs) with attention mechanisms. The first one operates on each word to encode them into sentences. The second one operates on the sentence representations and encodes them into a document representation that is finally used as input to a classifier. This network can model the sequential property (order of words) of text as well as the hierarchical property (words form sentences, sentences form documents). During the course of our research, the hyperpartisan news detection challenge (will be explained further in Chapter 4) was held. Many teams who participated also used neural networks to detect hyperpartisan articles [34, 46].

There is no absolute advantage of using one classification algorithm over another. The performance often differs per dataset and the features that are used.

2.3.2. Unsupervised learning

Unsupervised learning finds patterns in data without using pre-existing labels. In general, the method decides on a criterion of what it means for two samples to be similar, then it groups data samples that are similar together. Unsupervised learning is not as commonly used in partisanship detection. Here we only introduce topic modeling because it is important for text analysis and will be part of our method.

Topic Modeling

Topic models are generative probabilistic models that define how a document is created. For example, one of the most commonly used model, Latent Dirichlet Allocation (LDA), defines a topic as a multinomial distribution over words and an article as a mixture of topics sampled from a Dirichlet distribution. The original idea of LDA is to represent a text using some latent dimensions, which are being called "topics" and are groups of words that are used together in documents. To create a document, the model first chooses a topic distribution over the K topics. Each word is generated by picking a topic based on the distribution and then picking a word based on the topic's distribution over words. In general, we work backward to find the topics from a corpus. We estimate the topic distribution over words for the whole corpus and the topic distribution for each document. Then we can inference the topic distribution of new documents from the estimated distributions.

The main disadvantage of LDA is that the number of topics needs to be pre-defined. A proper number is crucial for a good clustering of topics that is meaningful. There are two main strategies to do that. One is to use clustering metrics to measure how good the clustering is. For example, Krestel *et al.* [8] treated topics as clusters and evaluate silhouette coefficients to find the proper number of topics. The second is to treat the topic number as a hyperparameter and see which value works best for the downstream task. Ahmed and Xing [47] evaluated the final classification accuracy on different numbers of topics to decide which number is the most beneficial to the downstream classification task.

Lin and He [13] proposed a joint sentiment/topic model by adding a sentiment layer between the document and the topic layer of LDA. Krestel *et al.* [8] used LDA to cluster articles into topics about climate change, then use SentiWordNet to find the sentiment towards the topics. LDA is most useful when we want to retrieve articles of different partisanship per topic. It can also be used to analyze the sentiment around each topic for different news media.

2.4. DATASETS

The aforementioned classification models require data to be trained and evaluated. In this section, we review datasets that have been used in literature, their annotation process, and the performance achieved. This helps us understand the existing resources and limitations.

2.4.1. EXISTING DATASETS

Table 2.2 summarizes the datasets that we surveyed with the size, labels, and annotation level. In the following, we describe each of them in more detail.

Table 2.2: Summary of surveyed datasets.

Dataset	Text source	Size	Label (#class)	Annot. level
bitterlemons [31]	blog posts	594	Palestinians, Israeli (2)	article
CMU Political Blog Corpus [48]	blog posts	13,246	conservative, liberal (2)	publisher
Yano *et al.* [49]	sentences from CMU blog	1,041	conservative, liberal (2)	sentence
IBC [22]	books and magazines	874	1. left, right, neutral (3) 2. fine-grained (10)	article
Iyyer *et al.* [23]	sentences from IBC	4,062	liberal, conservative, neutral (3)	sentence
BuzzFeed-Webis Fake News Corpus [27]	news	1,627	1. hyperpartisan, mainstream (2) 2. left, right, mainstream (3)	publisher
Kulkarni *et al.* [32]	news	120K	left, right, neutral (3)	publisher
SemEval2019 [1]	news	1M	hyperpartisan non-hyperpartisan (2)	publisher
		645	hyperpartisan non-hyperpartisan (2)	article

bitterlemons The bitterlemons corpus is not strictly a political ideology or partisanship dataset but we included it because it can be considered related. It consists of articles written by Palestinian and Israeli authors on specific topics. These articles were collected from the website bitterlemons.org, whose objective is to "contribute to mutual understanding between Palestinians and Israelis through the open exchange of ideas." Each week, 4 articles are published, two from each side. Of the two articles from one side, one is written by an editor and the other by an invited guest. In the original paper, Lin *et al.* [31] aimed to predict the perspective that an article is written from. For example, whether an article is written from a liberal or conservative perspective, and in their case, a Palestinian or Israeli perspective. They collected 594 articles between 2001 and 2005 with a balanced distribution between perspectives. Under the setup of the website, the labels are available when the articles were published so no further annotation was needed. The dataset is publicly available[2].

With the dataset, Lin *et al.* [31] trained NB and SVM classifiers based on the BoW features. They compared the performance of the articles written by editors and guests separately. The reported metric was averaged accuracy in a 10-fold cross-validation setting. Under this setting, NB consistently performed better than SVM, reaching an accuracy of 0.99 and 0.89 on editor and guest, respectively. To make sure that the classifier didn't learn the writing style of an author, they also experimented on training on editors' articles and testing on guests' articles and vice versa. The accuracy remained high at 0.90, which means that the method is effective. Ahmed and Xing [47] used a revised LDA model that models both topic, ideology, and the interaction between the two. They randomly split the data into 80% training and 20% testing and achieved an accuracy of 0.98 with their proposed model.

CMU 2008 Political Blog Corpus The CMU Political Blog Corpus contains articles from 6 political blogs in the year of 2008 near the U.S. Presidential election [48]. At that time, three of the blogs supported Barack Obama and the Democratic party while the other three supported John McCain and the Republican party. The articles within the blogs are labeled as liberal and conservative respectively. The corpus includes all blog posts from the sites with more than 200 words. They collected 13,246 posts in total.

From this corpus, Yano *et al.* [49] selected 1,041 sentences that are prone to bias using bi-grams and LIWC. They then assigned crowdsourcing tasks to label the sentences based on their intensity of partisanship and orientation of partisanship (liberal or conservative). The task was assigned to people that reside in the United States and had more than 90% approval history on the platform. From the collected annotation, they further filtered out answers from annotators that shows low agreement. The final inter-rater agreement was measured by an average of pair-wise kappa of the most frequent 50 annotators. The score was 0.55 for labeling intensity of partisanship and 0.50 for labeling orientation. From the annotation, they found that a blog site contains more sentences that match its known political leaning, i.e. a conservative blog uses more conservative sentences as labeled by independent workers. This is a confirmation that people can perceive the polarity of partisanship in sentences and that the site's polarity can be observed by the content it publishes. Ahmed and Xing [47] used the dataset to test their

[2] http://perspective.informedia.cs.cmu.edu/data/bt.1.0.zip

algorithm. They used articles from 4 blog sites for training their mv-LDA model and evaluated on the other 2 blog sites. The achieved accuracy is around 0.68.

Ideological Books Corpus (IBC) The original IBC consists of books and magazines whose authors are considered representative of a political ideology. 112 books and 10 magazine titles were manually labeled by a domain expert into left, right, or center. All the issues under a magazine title were labeled by the title's ideology. Documents labeled as left or right were further annotated into finer ideologies such as far-left, populist-right, etc. In total 874 documents were labeled [22].

Iyyer *et al.* [23] further processed the dataset and picked out sentences in the documents that are politically biased. They used crowdsourcing to annotate 4,062 sentences at a sub-sentential level because their objective was to find which part of a sentence reveal political ideology. They did not further filter the annotators and annotation. The author included all sentences where at least two annotators agreed on a label on the root of the sentence. In the final dataset, there are 2,025 liberal sentences, 1,701 conservative sentences, and 600 neutral sentences. This dataset is publicly available [3]. With this annotated dataset, they trained a Recursive NN which reached an accuracy of 0.69.

BuzzFeed-Webis Fake News Corpus 2016 The BuzzFeed-Webis Fake News Corpus was collected for fake news detection, however, in the original paper, Stein *et al.* [27] also experimented on hyperpartisan news detection. The dataset consists of 1,627 articles from 3 left-wing, 3 right-wing, and 3 mainstream publishers. The articles are labeled by the publishers' partisanship. They developed an RF classifier based on writing style features including n-gram of characters, PoS, readability scores, etc. The performance was reported as an average of a 3-fold cross-validation performance, where each fold comprised of one publisher from each orientation. This was done to ensure that the classifier does not rely on publisher style. An accuracy of 0.75 was achieved for binary classification task of detecting hyperpartisanship. For a three-way classification, the F1-score on the left, right, and mainstream classes were 0.20, 0.57, 0.75. The dataset contains article-level labels for fake news so they did not evaluate hyperpartisan news classifiers on article-level labels. The dataset is publicly available [4].

Kulkarni *et al.* [32] The dataset used by Kulkarni *et al.* [32] was not published and was collected only for their work. We describe it as it is closely related to our task. They retrieved a list of 59 left, center, and right publishers from AllSides and extracted articles from them online. Their dataset consists of 120K articles in total with roughly balanced classes. To test their model, they randomly split the dataset into 80% training and 20% testing set. The model achieved an F1-score of 0.79, which is high in a three-way classification task. They compared with several baselines to show that their model outperforms many existing approaches.

[3] https://people.cs.umass.edu/~miyyer/ibc/index.html

[4] https://zenodo.org/record/1239675#.XQy6h4gzZPY

SemEval2019 The SemEval2019 Hyperpartisan News Detection dataset is explained in detail in Chapter 4 because it is one of our main datasets. We list it in Table 2.2 for comparison.

Table 2.3 summarize the performance achieved on the datasets, including the main features and models that were used.f

Table 2.3: List of results for partisan news or political orientation detection. N is the number of class in the classification task.

Dataset	N	Author	Features	Model	Performance
bitterlemons	2	[31]	normalized tf	SVM NB	Accuracy: 0.85 Accuracy: 0.90
	2	[47]	normalized tf	SVM mv-LDA	Accuracy: 0.94 Accuracy: 0.98
CMU-blog	2	[47]	normalized tf	SVM mv-LDA	Accuracy: 0.67 Accuracy: 0.68
BuzzFeed	2	[27]	character n-grams, readability scores, GI dictionary, PoS, ratios of quoted words, number of paragraphs	RF	Accuracy: 0.75 F1-score: 0.75
	3	[27]	same as above	RF	Accuracy: 0.60 F1-score: 0.51
Kulkarni et al.	3	[32]	word embeddings	MVDAM	Precision: 0.80 Recall: 0.80 F1-score: 0.80
SemEval2019	2	[33]	ELMo [50]	CNN	Accuracy: 0.82
	2	[34]	handcrafted features, USE [51]	LR	Accuracy: 0.82
	2	[35]	BERT [52], article-length, useful phrases	softmax	Accuracy: 0.81
	2	[53]	GloVe	SVM	Accuracy: 0.81

Of the datasets that have been introduced, BuzzFeed Webis Fake News Corpus 2016 and the dataset of Kulkarni *et al.* [32] are most related to our task. Other datasets concern differentiating the orientation of the article, e.g. Palestine v.s. Israeli, conservative v.s. liberal, or left v.s. right, while we are interested in detecting partisan news from non-partisan ones. Some of them are annotated on the sentence level and we are interested in the document level. Moreover, they are mostly from blog posts or political books. These articles have different characteristics from news articles because the authors are more explicit about their political partisanship.

We identified two problems with the datasets used in the field:

1. There are no benchmark datasets that are consistently used by the field for comparing the performance of different methodologies. Some work used data collected for their purpose and did not release the dataset.

2. There is no article-level labeled partisan news dataset, except the one that was released during our work (SemEval2019, which we will be explained in Chapter 4). Therefore, previous evaluations were performed on the publisher-level labels.

2.5. RESEARCH GAP

From the survey, we see that many features and models have been proposed. The methods overlap with those from sentiment analysis, subjectivity analysis, and fake news detection. We also see that there are some datasets but mainly from political blog posts. For news domain data, they are labeled only by publishers. Therefore, algorithms are trained and evaluated on the publisher-level labels. However, no one has studied how much these noisy labels affect the performance of partisan news detection. We thus identify a need to benchmark the models learned from both publisher-level and article-level labels to compare the performance difference. Moreover, the performance needs to be evaluated on article-level labels to indicate the performance we can expect on the future prediction of partisan news.

3

Methodology

In Chapter 2, we reviewed approaches to partisanship detection and existing datasets. We argued that formulating the problem into a machine learning classification task is more scalable and general for partisan news detection. Since our focus is on investigating the effect of different annotation levels on performance, we did not design new features and models. Instead, we used well-established features and models of different characteristics and compared the performance on two datasets and two annotation levels. In this chapter, we explain the implementation of our classification system. This system would be used in all the experiments in Chapter 5, 6, and 7.

3.1. Classification pipeline

We treat partisan news detection as a document-level classification task as it fits our objective of predicting a label for each news article. For our classification system, the input is the textual content of a news article. The output is a binary label indicating whether the article is partisan or non-partisan. Figure 3.1 illustrates our classification pipeline. The figure shows that the training and testing articles go through the same preprocessing and feature extraction modules. Before training the classification model, we check whether the data is imbalanced. If it is, we apply SMOTE or reweigh the misclassification costs to remedy the imbalance. The trained model is then used to predict a label for each test article. Finally, we evaluate the results with three evaluation metrics. Each of the following sections explains a component in the pipeline.

3.2. Preprocessing

The majority of the articles in our datasets are from online sources. To extract clean textual contents, we removed HTML tags, #-tags, @-tags, and URLs. We also normalized quotation marks, hyphens, and removed any formatting white spaces. Except when counting capitalized words, all the texts were lowercased before further processing. Other

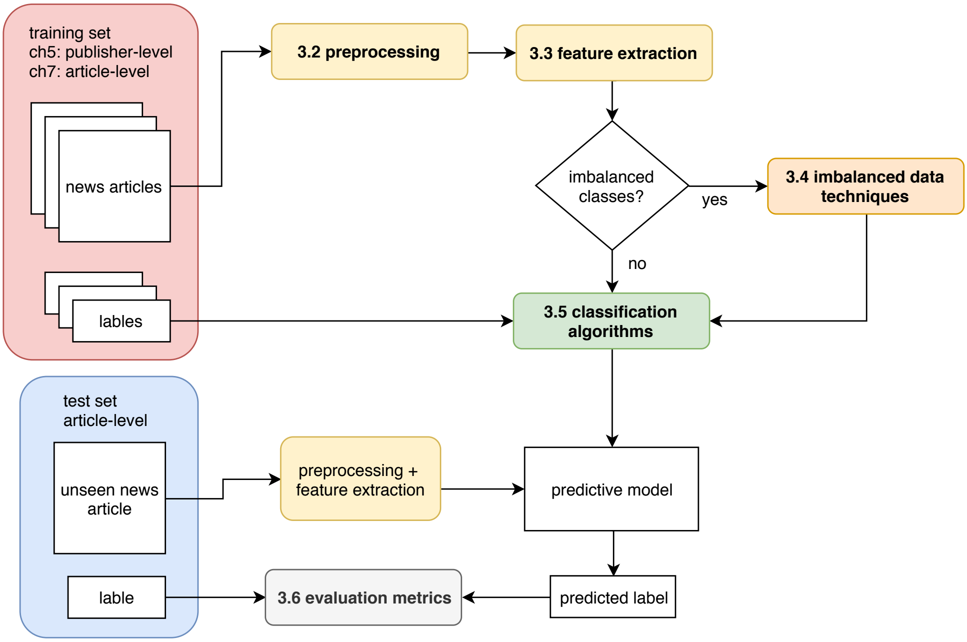


Figure 3.1: Classification pipeline of the partisan news detector.

preprocessing steps such as tokenization and lemmatization differed per feature so we explain them together with the features.

3.3. FEATURES

We constructed 5 feature sets. The first two encoded word-usage in the text, n-grams and averaged word embeddings (Word2Vec). The third captured the writing style of an author, taking the complexity of text and punctuation usage into account. The fourth was based on LDA topic models that capture the topic of an article, and the last was based on psycho-linguistic lexicons. Besides, we experimented with a combined feature set that consisted of word embeddings, writing style, topic, and lexicons to account for the interactions between them. The main reason that we used a diverse set of features is to investigate whether the performance difference between the two annotation levels varies across features. Additionally, we could investigate which feature is more effective under which level of annotation. Table 3.1 summarizes the preprocessing step and external pre-trained resources that were used to extract each feature. In the following, we describe how the feature vectors were constructed for each feature set.

3.3.1. N-GRAMS

To collect word n-grams, we tokenized the articles and extracted all unigrams and bigrams. For English articles, we used spaCy [54] to tokenize and lemmatize the texts to reduce morphologies. For Dutch, there was no existing lemmatizer so we did not lem-

Table 3.1: Summary of feature sets.

Feature set	Preprocess	Pre-training
N-grams	tokenize, lemmatize (English)	train w/ publisher-level part
Word2Vec	tokenize	pre-trained word vectors
Writing style	tokenize	pre-trained POS-tagger
Topic	tokenize, lemmatize (English)	train w/ publisher-level part
Lexicon	tokenize, stem/lemmatize (English)	external lexicons
Combined	Word2Vec + writing style + topic + lexicons	

matize the text. Terms that occurred in more than 60% of the corpus were discarded because they appeared too often and were assumed to have little discriminating power. Terms that occurred less than 0.01% of the corpus were also discarded due to the low frequency. The remaining terms were used as features. In a variation, we used the most frequent 50K terms to construct the n-gram feature with a smaller dimension.

To account for differences in article length, the tf-idf weighting was used. We used the publisher-level part of the datasets to decide the n-grams and their tf-idf weightings.

3.3.2. AVERAGED WORD EMBEDDINGS

We used the pre-trained word embeddings trained on a portion of Google News dataset and contains vectors for 3 million words and phrases. The vector dimension is 300 [1]. The pre-trained vectors were chosen because they were trained on news corpus and corresponded to our domain. For Dutch, we used the word vectors trained with the Dutch CoNLL17 corpus and contains 2.6 million words [55]. The vector dimension is 100.

Since we did not use sequential models such as RNN, as will be explained in section 3.5, we needed a fixed-length feature vector as input. The word vectors provided us with a fixed-length vector for each word. To aggregate them to represent an article, we averaged the vectors. For each article, we tokenized the text and removed stop words. Then we took the word vectors for each word in the pre-training vocabulary and averaged them. We limited the aggregation to at most 300 words since we found it to perform better empirically than using the whole article, especially for long articles.

3.3.3. WRITING STYLE

It has been shown that features that capture the general style of writing are useful for partisan and fake news detection [27, 42]. Writing styles are captured by the punctuation, word length, sentence length, syntactic functions of the words, etc. We explain the features in the following.

Readability There are different readability measurements, for example, the Flesch–Kincaid (F-K) and Automatic Readability Index (ARI). These measurements assume that sentences that contain more words are more difficult to follow than shorter sentences. Similarly, words that contain a lot of character and syllables are harder to read than those

[1] https://code.google.com/archive/p/word2vec/

that use fewer. We collected the components of these readability scores as our features. These included the average number of characters per word, the average number of syllabus per word, the average number of words per sentence, the average number of long words per sentence, and the average number of long words per article. Long words were words that have more than 6 characters.

Part-of-Speech We used spaCy for PoS tagging. For English, the model is a CNN trained on OntoNotes with GloVe vectors trained on Common Crawl. The reported PoS accuracy is 96.92. For Dutch, the model is a multi-task CNN trained on the Universal Dependencies and WikiNER corpus. The reported accuracy is 91.57. After tagging, we counted the occurrences of each tag and normalized them by the total number of words.

Word usage and punctuation We also computed lexical diversity, number of stop words, number of all capitalized words, number of exclamation marks, quotes, and question marks to capture writing style. Lexical diversity is the percentage of unique words out of all words used. The English stop words we used were the default ones in spaCy. The Dutch stop words were collected from `https://github.com/stopwords-iso/stopwords-nl`.

There were in total 30 features for English and 28 for Dutch. We did not find an established method to derive the number of syllables in a Dutch word so we excluded two features based on syllables for Dutch.

3.3.4. TOPIC

Some topics are inherently more prone to partisanship than others. For example, a topic about healthcare or environment is often more controversial than one about entertainment. Therefore, we included features that were derived from topic models.

We trained LDA models with different numbers of topics using the publisher-level parts of the datasets. To train the model, each article was tokenized. Unigrams and bigrams were extracted so that frequent bigrams such as "new york" were modeled as terms. For English, we lemmatized the words for unigrams and bigrams. For Dutch, we did not lemmatize due to a lack of available lemmatizer, as mentioned earlier. We used 20, 30, and 40 topics. The estimated parameters were then used to infer the distribution of topics for each article. For example, if there are three topics, education, military, and politics, an article about educational reform might have a distribution of 60% education, 5% military, and 35% politics. The feature vector is thus [0.6, 0.05, 0.35]. The dimension of the feature is the number of topics. The training of LDA models and inference were both implemented with gensim [56].

3.3.5. PSYCHO-LINGUISTIC LEXICONS

This set of features were derived from diverse lexicons that have been built to capture subjectivity, sentiment, bias, emotions, and morality. For each lexicon, we counted the frequency of the words in each category and used the normalized frequency as the feature value. The lexicons we used are as follows.

MPQA Subjectivity lexicon [57] The MPQA (Multi-Perspective Question Answering) lexicon includes 8,222 words. Each word is tagged to be either strongly subjective or weakly subjective. Each word is also tagged with a PoS tag and its prior polarity (positive, neutral or negative). We collected six categories of words, namely the combination of {strongly, weakly} and {positive, neutral, negative}. For each article, we tokenized the text and tagged each token with a PoS tag. A word was counted only when the PoS also matched the ones in the lexicon.

Bing Liu's Opinion Lexicon [58] This lexicon is part of the techniques used by Hu and Liu [58] for mining product reviews. They collected 4,783 negative words and 2,006 positive words in total that describe opinions. The lexicon includes morphological variants. For example, "warm", "warmer", "warmly", and "warmth" are all included in the lexicon. We thus used it directly to count frequencies of words.

NPOV Bias Lexicon [11] The bias word list was constructed by checking the edit in Wikipedia. Due to the Neutral Point of View (NPOV) policy of Wikipedia, the text is constantly reviewed and edit to reduce bias. Recasens *et al.* [11] analyzed the edits and constructed the list by putting together words that were changed between edits. For example, murder, pro-life, far-right, and conspiracy are on the list. A total of 654 words are included. In addition to the biased word list, we used several word lists that were used by Recasens *et al.* [11] to detect these biased words in Wikipedia articles. These word lists were shown to help predict epistemological bias, which is a bias that presupposes things that are commonly considered as true or false to be the truth. The lists included factives [59], assertive verbs [59], hedges [60], implicatives [61], and report verbs. Table 3.2 shows the number of words in each of the list and some example words.

Table 3.2: Number of words and example words in each bias word list.

Word list	#words	Example of words included
factive	27	learn, note, reveal, amuse, strange, observe, suffice
assertive verbs	66	think, believe, suppose, imagine, argue, appear
hedges	100	almost, apparent, appear, claim, doubt, largely, likely
implicatives	32	manage, remember, sense, happen, attempt, allow
report verbs	181	admit, declare, imply, refer, vow, voice
NPOV bias words	654	assert, claim, democratic, homosexual, muslim

LIWC2007 [41] Linguistic Inquiry and Word Count (LIWC) is a dictionary that was developed to identify groups of words that express basic emotional and cognitive dimensions in psychology. It has been found that the words people use in their writings reveal their physical and mental health, therefore, LIWC was composed to help with text analysis that exploits this.

We used the LIWC2007 dictionary, which has almost 4,500 words and word stems categorized into 64 categories. A word can belong to more than one categories. For example, the word "cried" falls into categories of sadness, negative emotion, overall affect, verb, and past tense verb. We included all 64 categories as features.

Moral foundations dictionary [62] Moral foundations theory is a social psychological theory that tries to explain the variations in human moral reasoning using five sets of basic moral values, harm, fairness, ingroup loyalty, authority, and purity. It is frequently used to explain political ideology. Graham *et al.* [62] found that liberals value harm and fairness more than conservatives, while conservatives value all of the five categories more equally. They built a moral foundations dictionary. For each of the five moral foundations, the dictionary contains two sub-dictionaries with words that endorse the value (virtue) and recognition of the value being violated (vice). Using the dictionary, Fulgoni *et al.* [63] showed that liberal, conservative, and libertarian news publishers frame political issues differently by word usage. In addition, they were able to predict article partisanship with the dictionary above chance.

We added features derived from the moral foundation dictionary to our lexicon features to capture the moral foundations that each article presents. We expect that the distribution of word usage in each moral category would help decide whether the article is partisan or not. Table 3.3 shows the number of words in each category and a random set of words to give an idea of which words are included. A word that ends with a * means that it can match to any ending, i.e. it is the stem of a word and any derived form of the word counts. For this reason, we stemmed each word in the article before counting the frequency.

Table 3.3: Number of words and example words in each sub-dictionary of the moral foundations dictionary.

Foundation	#words	Examples of words included
HarmVirtue	15	safe*, peace*, compassion*, shelter, benefit*, defen*
HarmVice	34	hurt*, cruel*, ravage, attack*, abandon*, exploit
FairnessVirtue	25	fair, justice, reasonable, balance*, unprejudice*
FairnessVice	17	unfair, unjust, bias*, segregat*, exclusion, discriminat*
IngroupVirtue	28	nation*, family, group, loyal*, communit*, collectiv*
IngroupVice	22	foreign*, enem*, betray*, individual*, terroris*, immigra*
AuthorityVirtue	45	obey*, duty, legal*, leader*, rank*, status*
AuthorityVice	36	defian*, rebel*, illegal*, defy*, unfaithful, oppose
PureVirtue	34	pure*, clean*, holy, virgin, modesty, upright, innocent
PureVice	53	disgust*, disease, sin, dirt*, deprav*, exploitat*, debase*
MoralityGeneral	40	righteous*, ethic*, good, lesson, evil, offend, character

Dutch Lexicons All the aforementioned lexicons are in English. There are few resources in Dutch. Although we can translate our news article into English to use the lexicons or translate the lexicons into Dutch, both methods suffer from the quality of the translation. With current translation systems, the general semantics are often preserved but the subtle usage of language is difficult to preserve. Since we expected that the Dutch articles present partisanship subtly, we did not perform translation. We used two available lexicons. The first one was the positive and negative word lists constructed by Chen and Skiena [64]. They used several language resources to build a knowledge graph and propagated the sentiment from some seed words obtained from English to obtain the sentiment of a large vocabulary. The second one is annotated by human annotators [65].

The English lexicon feature contained in total 89 features and the Dutch one contained 7 features.

3.4. LEARNING FROM IMBALANCED DATA

Most standard machine learning algorithms expect balanced class distributions or equal misclassification costs. Therefore, they fail to capture important aspects of imbalanced datasets [66]. The article-level parts of our datasets are imbalanced, containing less partisan articles. We used 2 techniques to mitigate the imbalance, a re-sampling technique called SMOTE and weighting of misclassification costs.

Synthetic minority oversampling technique (SMOTE) To make the classes more balanced, we can simply replicate a portion of the minority class to pretend that we have more samples, this is called oversampling. SMOTE is a more advanced way to oversample by synthesizing new samples in the feature space. The idea is similar to data augmentation. The main difference is that data augmentation operates in the data space while SMOTE is more general and operates in the feature space.

The algorithm of SMOTE is as follows. Consider a minority sample x_i and its k nearest neighbors (k other minority samples that are closest to x_i in terms of Euclidean distance). The way to generate a new minority sample x_{new} is to randomly select one of the neighbors x_{zi}, connect it with x_i, and randomly choose a point on the line. In other words,

$$x_{new} = x_i + \lambda \times (x_{zi} - x_i) \tag{3.1}$$

where λ is a random number in [0,1] [67]. We used the implementation from imblearn [68] and sampled until the two classes are balanced.

Cost-sensitive weighting Most classifiers have a cost function that they minimize. When the cost for misclassifying the majority class is the same as the minority class, the classifier would favor the majority class as it optimizes the cost. To fix that, we weigh the cost using the class ratio so that the accumulated cost of misclassifying all the samples in the minority class is the same as misclassifying those in the majority class. We use the implementation from scikit-learn [69], where the weighting given to a class j is

$$w_j = \frac{n}{k \times n_j} \tag{3.2}$$

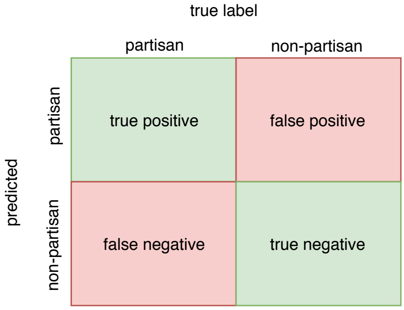
where n is the total number of samples, k is the number of classes, in our case 2, and n_j is the number of samples in class j.

3.5. CLASSIFICATION MODEL

We used two classification models, LR and SVM. The models were chosen due to their robustness to perform well on both large and small data sizes. For the implementation of LR and SVM, we used scikit-learn [69]. A linear SVM using the liblinear [2] was used for the publisher-level experiments because of the large number of samples. For the smaller article-level dataset, we used an SVM with a radial basis function (rbf) kernel because a kernel SVM can learn non-linear functions and often performs better on small datasets.

3.6. EVALUATION METRICS

Classification methods need to be evaluated to decide the effectiveness. The most intuitive metric to evaluate how well we predict new samples is to see the percentage of correct predictions. This is called accuracy. However, accuracy doesn't tell us how well we do for each class, i.e., how many partisan articles are not detected and how many detected articles are non-partisan. This is especially misleading when the class distribution is very skewed. For example, if we have 90 non-partisan articles and 10 partisan articles in the testing set, predicting all of them to be non-partisan would achieve an accuracy of 90%. In this case, precision, recall, and F1-score are more informative. These metrics are based on the confusion matrix as shown in figure 3.2. In a task with two classes, there are 4 cases, true positive (TP) is the case when we detect a partisan article; false positive (FP) is when we detect a non-partisan article; true negative (TN) is when a non-partisan article is predicted to be non-partisan; false negative (FN) is when a partisan article is not detected. Precision, recall, and F1-score are then defined as follows:



		true label	
		partisan	non-partisan
predicted	partisan	true positive	false positive
	non-partisan	false negative	true negative

Figure 3.2: Confusion matrix of a binary classification problem.

$$precision = \frac{TP}{TP + FP}$$

[2] https://www.csie.ntu.edu.tw/~cjlin/liblinear/

$$recall = \frac{TP}{TP + FN}$$

$$F1 - score = 2 \times \frac{precision \times recall}{precision + recall}$$

In our experiments, we report precision, recall, and F1-scores on the partisan class. By having multiple metrics, we can study the characteristics of different classifiers. We use the F1-score as the main metric to evaluate the performance of a model. This is because it is a general evaluation that incorporates both precision and recall. Whether precision or recall is more important depends on the use case and objectives, which is for the end-user to decide.

4

Datasets

To investigate the effect of the level of annotation, we needed a dataset that is labeled both on the publisher and the article level. The dataset that we used was the dataset for PAN @ SemEval 2019 Task 4: Hyperpartisan News Detection[1]. In addition to that, we collected a Dutch dataset from the publications of DPG Media. The main reason to include a second dataset is to investigate to what extent our conclusions from one generalize to the other. If we have similar observations on both datasets, we have more confidence in the conclusions we make. Table 4.1 summarizes the two datasets in terms of the language, size, class distribution, number of publishers, and annotation method. By studying a Dutch dataset, we also hope to understand the media landscape in the Netherlands and whether that affects the detector's ability to detect partisanship.

In this chapter, we introduce the datasets used in our work, explain how they were collected and annotated to understand their properties and limitations. We start with a brief survey of methods to annotate publisher and article partisanship in section 4.1. This helps us understand the pros and cons of the annotation methods used in the datasets. We then describe the collection process of the news articles and labels, starting with the existing SemEval2019 dataset in section 4.2, then dpgMedia2019 in 4.3. We elaborate on the limitations we had when collecting dpgMedia2019, and show the label statistics. Finally, we made two analyses on both datasets in section 4.4 to understand the general statistics and whether our assumption that partisan publishers publish more partisan articles holds for the datasets.

4.1. Annotation of partisanship

4.1.1. Publisher partisanship

There are mainly three approaches that have been used to derive publisher partisanship, audience-based, content-based, and crowdsource. In the following, we describe the approaches and limitations of them.

[1] https://pan.webis.de/semeval19/semeval19-web/index.html

Table 4.1: Summary of the SemEval2019 dataset by Kiesel *et al.* [1] and dpgMedia2019 collected by us [2].

Dataset	SemEval2019		dpgMedia2019	
Label level	publisher	article	publisher	article
Language	English	English	Dutch	Dutch
Size	750,000	645	103,812	766
%partisan articles	50.0%	36.9%	50.9%	26.2%
#publishers	383	256	11	11
Annotation method	content-based (MBFC)	crowdsource (Crowdflower)	audience-based (survey)	crowdsource (survey)
Inter-rater agreement	X	Krippendorf's alpha: 0.40	X	Krippendorf's alpha: 0.18

Audience-based method The audience-based method is premised on the argument that readers read news from sources that have similar ideology as them [70, 71]. Therefore, the partisanship of the readers reflects that of the publishers. For example, Gentzkow and Shapiro [71] asked the readers of different publishers to categorize their political outlooks into one of the five categories: very conservative, somewhat conservative, middle of the road, somewhat liberal, and very liberal. To decide the partisanship of a publisher, they averaged the number of its readers in each category. Compared to content analysis, asking the reader to rate their political partisanship is light-weight and easier to scale. However, it is prone to self-bias. Moreover, it provides only relative measures of partisanship because it is reported that small differences between publishers could lead to substantial audience fragmentation [72].

Content-based method The content-based method decides the partisanship of a publisher by analyzing the partisanship of the articles published by the publisher [8, 12, 72, 73]. Gentzkow and Shapiro [12] used US Congressional speeches from politicians with known partisanship to select phrases that are specific to each party. Then they measured the frequency that the phrases are used in a publisher's publications to decide its partisanship. Similarly, Krestel *et al.* [8] used tf-idf weighting to measure the similarity between German parliament speech and news articles to find which publisher support which party on certain topics. Budak *et al.* [72] used crowdsourcing to decide the partisanship of articles. Then they aggregated the articles to derive the partisanship of each publisher. Due to the difficulty of reliable large-scale content analysis, these studies typically focus on a subset of articles, which limits the scope of the findings. For example, highly partisan language from Congressional speeches appears in only a small minority of news stories.

Crowdsourcing In crowdsourcing, the annotation is decided by surveying the public. One example is from the survey of Pew Research Center, where they asked people to put news outlets on the left-right spectrum [74]. This is prone to bias, especially that frequent readers of a publisher often have a different perception of it from infrequent readers. In the survey, it was found that people who are not frequent readers of a publisher often consider the publisher more partisan than they are rated by their frequent readers. This suggests that publishers with fewer readers would have more biased ratings given by non-frequent readers.

4.1.2. ARTICLE PARTISANSHIP

There are two methods to label article partisanship, expert annotation [22] or crowdsourcing [72]. Expert annotation requires a few domain experts (generally 3 to 5 to check the agreement between them) to read each article and label its partisanship. It is expensive but rather reliable. This is often considered ground-truth "gold data". Crowdsourcing is a cheaper method that distributes the task to the public. The obtained labels are noisier, but with a proper design of annotation task and post-processing, non-expert labels can be as effective as expert annotations in terms of the performance of the machine learning model that is trained [75].

4.2. SEMEVAL2019

SemEval2019 was constructed by the organizers of the Hyperpartisan News Detection task [1]. The objective of the task was described as: "given a news article text, decide whether it follows a hyperpartisan argumentation, i.e., whether it exhibits blind, prejudiced, or unreasoning allegiance to one party, faction, cause, or person." Their definition of hyperpartisanship aligns with our definition of partisanship but differs in terms of intensity. Hyperpartisan news often omits some truth to support certain political parties, Some of the articles can even be considered fake news.

4.2.1. COLLECTION AND ANNOTATION OF PUBLISHER-LEVEL DATA

To collect the large publisher-level parts of the dataset, Kiesel *et al.* [1] first compiled a list of publishers whose partisanship is known. They compared the list compiled by journalists from BuzzFeed News [2] and the one from Media Bias/Fact Check (MBFC) [3]. The two lists differ in nine publishers, which were removed. Publishers that were categorized as "Least Biased" or "Left/Right Center Bias" were considered non-hyperpartisan and otherwise hyperpartisan. Articles were then crawled from the publishers' websites and facebook feeds. They removed publishers that did not mainly publish political news and publishers that did not have a political section to collect articles from. After collection, articles with less than 40 words were discarded. The final dataset contains 754K articles, of which 750K are publicly available. Kiesel *et al.* [1] further split the dataset into training and validation sets with different publishers for the hyperpartisan news detection task. This is meant to prevent participants from overfitting on publishers. In both sets, the two classes are balanced. The hyperpartisan class has also a balanced distribution between

[2] https://www.buzzfeednews.com/

[3] https://mediabiasfactcheck.com/

left- and right-wing hyperpartisanship. The number of articles per class in the two sets is listed in Table 4.2.

Table 4.2: Summary of the data size and class distribution of the publisher-level part of SemEval2019.

Partisanship	Hyperpartisan		Non-hyperpartisan		
	left	right	left-center	neutral	right-center
Training set	150,000 (25%)	150,000 (25%)	70,053 (11.7%)	187,114 (31.2%)	42,833 (7.1%)
Validation set	37,500 (25%)	37,500 (25%)	23,473 (15.7%)	38,296 (25.5%)	13,231 (8.8%)

4.2.2. Collection and annotation of article-level data

The article-level part of the dataset is based on the dataset constructed by Vincent and Mestre [76]. Vincent and Mestre [76] used Crowdflower (now Figure Eight)[4] to distribute the annotation task of around 1,000 political articles. Each annotator was asked the question, "Overall, how biased is the article?". They provided the rating on a 5-point Likert scale. In total 50 annotators participated in the annotation task and each article was annotated by 5 to 15 annotators.

Quality control and agreement analysis

To control the quality of annotations, Vincent and Mestre [76] constructed a "gold dataset", where one-fourth of the distributed articles were randomly selected and labeled by two experts. This dataset was then used to evaluate the reliability of annotators. The initial inter-rater agreement was evaluated using Krippendorf's alpha and was 0.078. After removing unreliable annotators, they reached an agreement of 0.4, which was not high but reasonable for a difficult task like this.

In the end, a dataset of 1,274 articles was created, where 645 were released publicly and used in this thesis. Of the 645 articles, 37% is hyperpartisan.

4.3. dpgMedia2019

The second dataset, which we call dpgMedia2019, is a Dutch dataset that we collected. All the articles in the dataset were published by DPG Media. We picked 11 publishers in the Netherlands for the dataset. These publishers include 4 national publishers, Algemeen Dagblad (AD), de Volkskrant (VK), Trouw, and Het Parool, and 7 regional publishers, de Gelderlander, Tubantia, Brabants Dagblad, Eindhovens Dagblad, BN/De Stem, PZC, and de Stentor. The regional publishers are collectively called Algemeen Dagblad Regionaal (ADR). The dataset is publicly available[5].

[4] https://www.figure-eight.com/

[5] https://github.com/dpgmedia/partisan-news2019

4.3.1. Collection and annotation of publisher-level data

We used an internal database that stores all articles written by journalists and ready to be published to collect the articles. From the database, we queried all articles that were published between 2017 and 2019. We filtered articles to be non-advertisement. We also filtered on the main sections so that the articles were not published under the sports or entertainment sections, which we assumed to be less political. After collecting, we found that some articles were published by several publishers, especially a large overlap existed between AD and ADR. Since we wanted to label articles by its publisher's partisanship, each article should have a unique publisher. To deal with this problem without losing many articles, we decided that articles that appeared in both AD and ADR belonged to AD. Therefore, articles were processed in the following steps:

1. Remove any article that was published by more than one national publisher (VK, AD, Trouw, and Het Parool). This gave us a list of unique articles from the largest 4 publishers.
2. Remove articles that were published by both AD and ADR from ADR.
3. Remove any article that was published by more than one regional publisher (ADR).

The process assured that most articles are unique to one publisher. The only exceptions were the AD articles, of which some were also published by ADR. This was not ideal but acceptable as we show in the next section that AD and ADR publishers had the same partisanship labels. We ended up with 103,812 articles.

Annotation of publisher partisanship

At the time of the dataset creation, there was no comprehensive research about the partisanship of Dutch publishers, like that from MBFC, to our knowledge. Therefore, we needed to decide the publisher partisanship. We adopted the audience-based method to decide the partisanship because the data were available to us. Within the survey that will be explained in 4.3.2, we asked the annotators to rate their political leanings. The question asked an annotator to report his or her political standpoints to be extreme-left, left, neutral, right, or extreme-right. We mapped extreme-left to -2, left to -1, center to 0, right to 1, extremely-right to 2, and assigned the value to each annotator. Since each annotator was subscribed to one of the publishers in our survey, we calculated the partisanship score of a publisher by averaging the scores of all annotators that subscribed to the publisher. The final score of the 11 publishers are listed in Table 4.3, sorted from the most left-leaning to the most right-leaning.

There were two limitations to this result. First, we had a sampling bias. The number of people from whom we computed the score was small, and the people who participated in the survey were older readers, with an average age of 66. Second, we see that the scores differed little between publishers. Ideally, we would consider publishers with an absolute partisanship score higher than 1 as partisan. None of our publishers had an absolute score higher than 1. This can mean that they are not very partisan by nature, or that averaging the readers' partisanship was not the best method (for example, extreme-left and extreme-right readers would average out each other). To decide partisan publishers, we cut the line at the largest score difference and treated VK, Trouw,

Table 4.3: Publisher, number of annotators, and the computed partisanship score.

Publisher	#annotators	Partisanship score
de Volkskrant	780	-0.6372
Trouw	491	-0.5438
Het Parool	410	-0.4341
de Gelderlander	208	-0.2452
Tubantia	236	-0.1864
Brabants Dagblad	188	-0.1862
Eindhovens Dagblad	194	-0.0722
BN/De Stem	151	-0.0464
PZC	202	-0.0248
Algemeen Dagblad	695	0.0302
de Stentor	55	0.0727
	total: 3,610	mean: -0.2067

and Het Parool as partisan publishers and the rest as non-partisan. This result largely accorded with that from the news media report from Pew Research in 2018 [6], which found that VK was left-leaning and partisan while AD was less partisan. A consequence of this cutoff was that all the partisan publishers are left-leaning. In SemEval2019, the hyperpartisan publishers composed of both left- and right-leaning ones.

Table 4.4 shows the publisher-level part of dpgMedia2019, with the number of articles and class distribution.

Table 4.4: Number of articles per publisher and class distribution of dpgMedia2019.

Partisanship	Partisan			Non-partisan	
Publisher	de Volkskrant	Trouw	Het Parool	AD	ADR
#articles	11,761	21,614	19,498	40,029	10,910
Total	52,873 (50.9%)			50,939 (49.1%)	

4.3.2. Collection and annotation of article-level data

Instead of using a crowdsourcing platform for annotating articles, we utilized a platform in the company that had been used by the market research team to collect surveys from the subscribers of different news publishers. The survey worked as follows: Annotators were first presented with a set of selected pages (usually 4 pages and around 20 articles) from the print paper the day before. They could select an article each time that they had read, and answer some questions about it. We added 3 questions to the existing survey. The first question (Q1) asked the level of partisanship (from unbiased to extremely biased). The second question (Q2) asked the polarity of partisanship (left or right), and the third question (Q3) asked which entities the article is partisan towards. The annotators could provide free texts such as pro-capitalism, anti-Muslims, etc. We also asked the political standpoint of the annotators. The complete survey can be found in Appendix A.

The reason for using this platform was two-fold. First, the platform provided us with annotators with a higher probability to be competent with the task. Since the survey was distributed to subscribers that pay for reading news, it's more likely that they regularly read newspapers and were familiar with the politics in the Netherlands. On the other hand, if we use crowdsourcing platforms, we need to design process to select suitable annotators, for example by nationality or anchor questions to test the annotator's ability. Second, the platform gave us more confidence that an annotator had read the article before answering questions. Since the annotators could choose which articles to annotate, it is more likely that they would rate an article that they had read and had some opinions about.

Using this platform also incurred some difficulties. First, the task was not as well-defined as on a crowdsourcing platform. We included the questions as part of an existing survey and we didn't want to create much burden to the annotators. Therefore, we did not provide descriptive text that explained which rating should be given in which case, as was done in the annotation task of SemEval2019. The annotations were more prone to annotator bias depending on how each person interpreted the task. Second, the annotators were a selective and biased group of people. For example, people who annotated article from de Volkskrant (VK) were regular VK readers. VK readers were found to be a more left-leaning and their ratings might reflect that. It is, however, nontrivial to know if we should adjust the ratings based on that and if so, how to scale.

The annotation task ran for around two months in February to April 2019. We collected more than 50K annotations for 1,536 articles from 3,926 annotators.

Annotation distributions

For Q1, where we asked about the level of partisanship, more than half of the annotations were non-partisan. About 1% of the annotation indicated an extreme partisanship, as shown in Table 4.5. For the polarity of partisanship, most of the annotators found it not applicable or difficult to decide, as shown in Table 4.6. For annotations that indicated a polarity, the highest percentage was given to progressive. Progressive and conservative seemed to be more relevant terms in the Netherlands as they are used more than their counterparts, left and right, respectively.

Table 4.5: Distribution of annotations of the level of partisanship.

non-partisan	reasonably non-partisan	somewhat partisan	partisan	extremely partisan	impossible to decide
52.85%	16.34%	10.54%	5.49%	0.91%	13.88%

As for the self-rated political standpoint of the annotators, nearly half of the annotators identified themselves as left-leaning, while around 20% were right-leaning. This

Table 4.6: Distribution of annotations of the polarity of partisanship.

left	right	progressive	conservative	others	not applicable	unknown
5.66%	2.74%	7.74%	2.78%	7.29%	54.81%	18.96%

is interesting because when deciding the polarity of articles, left and progressive ratings were given more frequently than right and conservative ones. This shows that these left-leaning annotators were able to identify their partisanship and rate the articles accordingly.

Table 4.7: Distribution of annotations of self-identified political standpoints.

extreme-left	left	middle	right	extreme-right
1.14%	46.87%	32.71%	19.14%	0.14%

As mentioned previously, we suspected that the annotators would induce bias in ratings based on their political leaning and we might want to normalize it. To check whether this was the case, we grouped annotators based on their political leaning and calculated the percentage of each option being annotated. In figure 4.1, we grouped options and color-coded political leanings to compare whether there were differences in the annotation between the groups. We observe that the "extreme-right" group used less "somewhat partisan", "partisan", and "extremely-partisan" annotations. This might mean that articles that were considered partisan by other groups were considered "non-partisan" or "impossible to decide" by this group. We didn't observe a significant difference between the groups. Figure 4.2 shows the same for the second question. Interestingly, the "extreme-right" group gave a lot more "right" and slightly more "progressive" ratings than other groups. In the end, we decided to use the raw ratings because how to scale the ratings based on self-identified political leaning needs more investigation.

QUALITY CONTROL AND AGREEMENT ANALYSIS

The question that we are interested in this thesis is Q1. In addition to the 5-point Likert scale that an annotator could choose from (non-partisan to extremely partisan), we also provided the option to choose "impossible to decide" because the articles could be about non-political topics. When computing inter-rater agreement, this option was ignored. The remaining 5 ratings were treated as ordinal ratings. The initial Krippendorff's alpha was 0.142, using the interval metric. This alpha is not high, but better than the initial value achieved in SemEval2019 dataset. However, we didn't have the "gold data" like SemEval2019 to filter out unreliable annotators. Instead, we used a more engineering-driven approach to devise filtering steps based on the information we had. These steps are as follows:

1. Remove uninterested annotators: we assumed that annotators that provided no information were not interested in participating in the task. These annotators always rated "not possible to decide" for Q1, 'not applicable' or "unknown" for Q2,

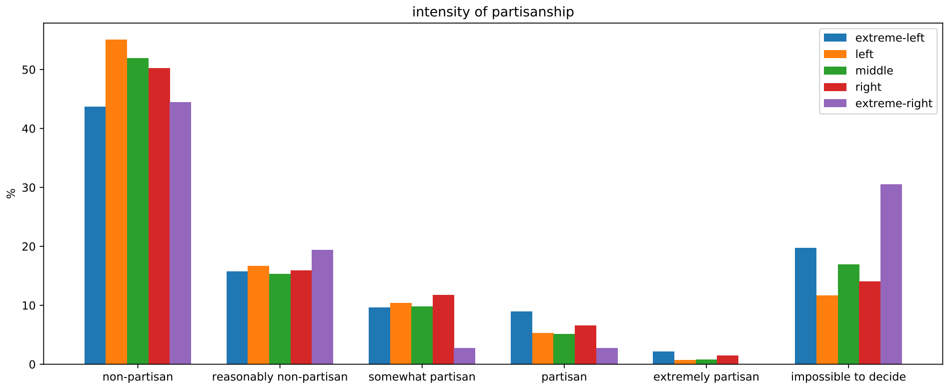


Figure 4.1: Percentage of annotation grouped by political leaning and annotation for the level of partisanship.

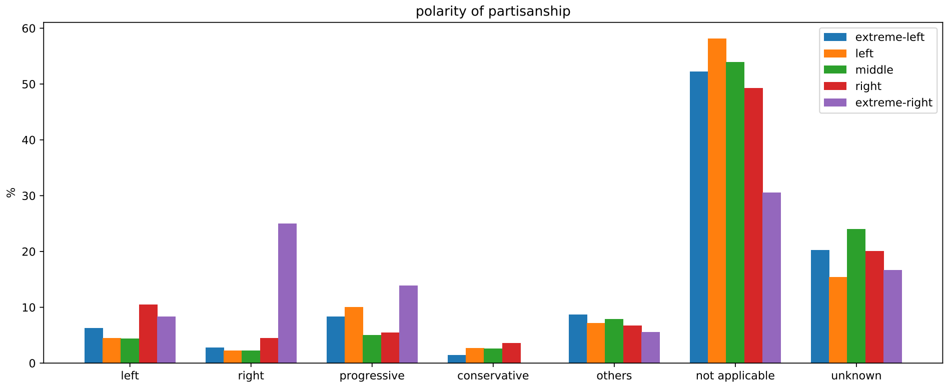


Figure 4.2: Percentage of annotation grouped by political leaning and annotation for the polarity of partisanship.

and provide no textual comment for Q3. There were in total 117 uninterested annotators and their answers were discarded.

2. Remove unreliable annotators: as we didn't have a "gold data" to evaluate reliability, we used the free text that an annotator entered in Q3 to compute a reliability score. The assumption was that if an annotator was able to provide texts with meaningful partisanship description, he or she was more reliable in performing the task. To do this, we collected the text given by each annotator. We filtered out text that didn't answer the question, such as symbols, 'no idea', 'see above', etc. Then we calculated the reliability score of annotator i with equation 4.1, where t_i is the number of clean texts that annotator i provided in total and N_i is the number

of articles that annotator i rated.

$$score_i = \frac{t_i + 1}{N_i} \times (t_i + 1) \tag{4.1}$$

The computation of the reliability score was based on two requirements. First, an annotator who provided meaningful text for more articles receives a higher score. Second, an annotator who provided meaningful text for a larger percentage of articles from all the articles he or she rated receives a higher score. We added one to t_i so that annotators that gave no clean texts would not all end up with a zero score but would have different scores based on how many articles they rated. In other words, if an annotator only rated one article and didn't give textual information, we considered he or she reliable since we had little information. However, an annotator that rated ten articles but never gave useful textual information was more likely to be unreliable. The reliability score was used to filter out annotators that rarely gave meaningful text. The threshold of the filtering was decided by the Krippendorff's alpha that would be achieved after discarding the annotators with a score below the threshold.

3. Remove articles with too few annotations: articles with less than 3 annotations were discarded because we were not confident with a label that was derived from less than 3 annotations.

4. Remove unreliable articles: if at least half of the annotations of an article were "impossible to decide", we assumed that the article was not about issues of which partisanship could be decided.

Finally, we mapped ratings of 1 and 2 to non-partisan, and 3 to 5 to partisan. A majority vote was used to derive the final label. Articles with no majority were discarded.

In the end, 766 articles remained, of which 201 were partisan. Table 4.8 shows the number of articles and the percentage of partisan articles per publisher. The final alpha value was 0.180.

4.4. ANALYSIS OF THE DATASETS

In this section, we analyze the properties and relationship of the two parts (publisher-level and article-level) of the datasets. In Table 4.9, we listed the length of articles of the two parts. We see that SemEval2019 has a big difference of average article length between the publisher-level and the article-level parts, while dpgMedia2019 has a similar distribution. This was likely because the publishers were mostly different between the two parts of SemEval2019 but were identical in dpgMedia2019.

The second analysis validated whether our assumption that partisan publishers publish more partisan articles holds. To do this, we used the article-level labels and calculated the percentage of partisan articles for each publisher. This value is then compared with the publisher partisanship. For SemEval2019, the publisher partisanship is retrieved from MBFC website. We map extremely-partisan (extremely-left and right) to 3, partisan (left and right) to 2, (left- and right-center) to 1, and least partisan to 0. Since

Table 4.8: Number of articles and percentage of partisan articles by publisher.

Publisher	#article	%partisan
de Volkskrant	166	27.11
Trouw	140	25.00
Het Parool	121	28.93
de Gelderlande	46	19.57
Tubantia	34	41.18
Brabants Dagblad	32	31.25
Eindhovens Dagblad	20	35.00
BN/De Stem	34	17.65
PZC	30	26.67
Algemeen Dagblad	133	24.06
de Stentor	10	0.00
	total: 766	mean: 26.24

Table 4.9: Statistics of the length of articles.

Number of words	SemEval2019		dpgMedia2019	
	publisher-level	article-level	publisher-level	article-level
Mean	804.7	546.9	470.1	471.2
SD	1221.9	530.8	387.5	275.1
50% percentile	492.0	392.0	381.0	451.0

this is an ordinal rank, not a continuous variable, we computed the Spearsman's correlation between it with the percentage of articles that were labeled by annotators as partisan. There are in total 16 publishers where we had values and the correlation was 0.55. For dpgMedia2019, we calculate Spearsman's correlation between the partisanship score derived from the readers and the percentage of partisan articles annotated by annotators for each publisher. The correlation was 0.21.

We see that SemEval2019 had a medium correlation while dpgMedia2019 had a low correlation. There were two reasons behind this. First, the articles in dpgMedia2019 expressed more subtle partisanship. The articles were reviewed by professional editors, reducing the usage of extreme language. On the other hand, some articles from SemEval2019 were very extreme. This affected the difficulty of the annotation task and agreement that could be reached. The article-level labels of dpgMedia2019 were noisier due to this, and due to the lack of gold data. Second, the nature of the publishers. Some hyperpartisan publishers in SemEval2019 were extreme publishers that resembled propaganda and fake news producers. These publishers consistently published hyperpartisan articles. The publishers from DPG Media were less partisan. The partisanship score we computed were less accurate and they were expected to be partisan only in a portion of the articles. The noisy partisanship scores of the publishers and the inconsistency in the partisanship of articles made the correlation low. The low correlation in dpgMedia2019 would make it more difficult for the detector to learn.

5

BENCHMARK FOR PUBLISHER-LEVEL LABELS

This chapter aims to answer the first research question (RQ):

> **RQ1: What is the benchmark performance of predicting partisanship of articles based on publisher-level labels?**

We hypothesized that although the performance would be bounded by the high-level and thus noisy labels, the classifier can predict partisan articles to a certain extent. This is based on the assumption that partisan publishers publish more partisan articles and non-partisan publishers publish more non-partisan articles. Therefore, the information contained in the data is more than noise.

We describe the experimental setup in section 5.1, including how we used the datasets and trained the models so that our experiments are reproducible. Then, we list the results that were obtained on the two datasets in section 5.2. Finally, we discuss the results and problems we found in section 5.3.

5.1. EXPERIMENTAL SETUP

In the experiments, we used the publisher-level parts of the datasets. For the SemEval2019 dataset, we sampled 300K articles from the training set to use for training and validation in a cross-validation setting. The main reason that we did not use the whole training set was that it took very long to obtain results with the original setup of training on 600K training samples and validating on 150K validation samples. We argue that 300K samples from around 200 publishers should be a large enough dataset to study the features and the effect of annotation level, especially that our classification models (LR and SVM) don't require as many data samples to perform well as models such as deep neural networks. In the following, the training set we refer to is this 300K subset of SemEval2019 and the whole publisher-level part of dpgMedia2019 (100K).

For our experiment, we used 5-fold cross-validation on the training set to choose the feature dimension, model hyperparameters, and classification algorithm to be used is the final test. The experimented values are listed in Table 5.1. We chose the parameters that achieved the highest average F1-score of the five folds. The cross-validation folds were split randomly but the two classes were kept balanced. We then trained the selected algorithm with the selected regularization factor with the whole training data. We took 40% of the article-level parts of the datasets as a global held-out test set that is used to report the performance of all experiments. We report the test metrics of precision, recall, and F1-score.

Table 5.1: Hyperparameters experimented using cross-validation.

Hyperparameters	Values
feature dimension	n-gram: original vector size, using the most frequent 50K terms topic: 20, 30, 40
model	LR, SVM
regularization	0.01, 0.025, 0.05, 0.075, 0.1, 0.25, 0.5, 0.75, 1, 2.5, 5, 7.5, 10

5.2. RESULTS

In the following, we show the results that we obtained with different feature sets. We also list the random baseline and a baseline where we always predict the partisan class. Since the random baseline achieved different scores each time, we list an average of 10 runs.

5.2.1. SEMEVAL2019

Table 5.2 shows the results for the SemEval2019 dataset. All our models using different features outperformed the baselines. We observed that recall was much higher than precision across features. This means that the classifier over-predicted articles to be partisan. The writing style feature achieved the best precision of 51.70% and F1-score of 62.81%. The best recall was achieved by Word2Vec at 95.79%, finding the majority of the partisan articles. When we combined several features, the performance decreased from using some of the individual features.

5.2.2. DPGMEDIA2019

We ran the same experiment on dpgMedia2019 and Table 5.3 shows the result. We see that all features outperformed the baselines, and the recall was higher than precision, similar to SemEval2019. The best F1-score achieved was 48.80% using the topic features and SVM. N-gram with SVM was the second strongest model, reaching the highest precision of 35.39%.

Table 5.2: Result of predicting partisan news with models trained by publisher-level labels (SemEval2019).

Features + Model	Precision	Recall	F1-score
Random	37.21	50.71	42.92
Always partisan	36.90	100.00	53.91
N-gram + SVM	43.69	94.74	59.80
Word2Vec + SVM	43.13	**95.79**	59.48
Writing style + SVM	**51.70**	80.00	**62.81**
Topic + SVM	44.12	94.74	60.20
Lexicon + SVM	47.40	86.32	61.19
Combined + SVM	43.81	96.84	60.32

Table 5.3: Result of predicting partisan news with models trained by publisher-level labels (dpgMedia2019).

Features + Model	Precision	Recall	F1-score
Random	26.50	50.35	34.72
Always partisan	26.24	100.00	41.57
N-gram + SVM	**35.39**	77.78	48.65
Word2Vec + SVM	32.64	77.78	45.99
Writing style + SVM	34.25	61.73	44.05
Topic + SVM	33.81	87.65	**48.80**
Lexicon + LR	27.03	**98.77**	42.44
Combined + SVM	34.57	69.14	46.09

When comparing the performance on the two datasets, F1-scores of dpgMedia2019 were at least 10% lower than that of SemEval2019 across all features. The following subsection is an effort to investigate possible reasons for the difference. It does not contribute to the research question but can be useful information for future dataset construction.

WHY WAS THE PERFORMANCE WORSE ON DPGMEDIA2019 THAN ON SEMEVAL2019?

When constructing the publisher-level part of the dpgMedia2019 dataset, we identified several differences from SemEval2019. First of all, the data size was smaller. Although we did not expect the difference between 300K and 100K training samples would affect the performance in our experiments, we validated how the training sample size influenced the results. We also studied two other differences that we suspected to lead to lower performance. The first one was the number of publishers, which is 11 for dpgMedia2019 and 193 for SemEval2019. This was potentially a problem because fewer publishers could make the classifier more prone to overfit on the publisher-level labels. The second was the ratio of left-leaning and right-leaning publishers within the parti-

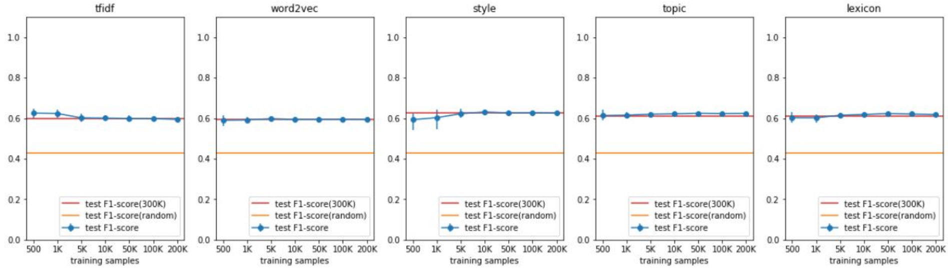


Figure 5.1: Performance of SVM with varying numbers of training samples (SemEval2019).

san publishers. In dpgMedia2019, all the partisan publishers are left-leaning while it is balanced in SemEval2019. The articles from the left-leaning publishers do not span the full partisanship space. We thus suspected the classifier to overfit on the left-leaning ideology.

The following investigations were performed on SemEval2019 with the SVM classifier to see whether specific properties of the dataset contributed to the performance. We experimented on SVM because it outperformed LR in all previous experiments on SemEval2019.

Training sample size In this experiment, we wanted to validate that the size of the training samples did not influence our result. We ran experiments on 5 features with different training sizes (500, 1K, 5K, 10K, 50K, 100K, 200K). In each experiment, we randomly sampled articles from the training set, keeping the class balanced. Each experiment was run 10 times to obtain the mean and variance of the performance. We show in Figure 5.1 the average F1-scores with error bars of word n-gram (tf-idf), Word2Vec, writing style, topic, and lexicon features. We plot the F1-score we achieved with 300K samples as a horizontal red line and the random baseline as a horizontal orange line.

For all features, the F1-score remained relatively stable. The performance we achieved with 5K samples was similar to that achieved with 300K. The variance decreased when the sample size increased, which is expected. From the results, we verified that the features and models we used were not affected by the size of the training data. Thus, the low performance on dpgMedia2019 is unlikely to be improved by collecting more data from the 11 publishers. The nearly constant performance also gave us confidence that sampling 300K for training and validation from the 750K available data for SemEval2019 did not lead to lower performance. In general, a training size of 10K is sufficient to achieve the best average performance achievable by the classifiers that we used with low variance.

Number of publishers In this experiment, we took 7 hyperpartisan-left, 7 hyperpartisan-right, and 14 non-hyperpartisan publishers from our training set with the most articles. We didn't use all of the publishers because some publishers had few articles and were difficult to sample from. For each setup, we sampled 2,500 articles from N left-hyperpartisan publishers, 2,500 from N right-hyperpartisan publishers, and 5,000 from $2N$ non-hyperpartisan publishers. This ensured that we had 10K training samples and balanced classes in each experiment. We varied N from 2 to 7. Each setup was performed 10 times to obtain the mean and variance of the performance. Figure 5.2 shows the average F1-scores and error bars of 5 features, word n-gram (tf-idf), Word2Vec, writing style, topic, and lexicon. We plot the performance we achieved with all publishers as a red horizontal line.

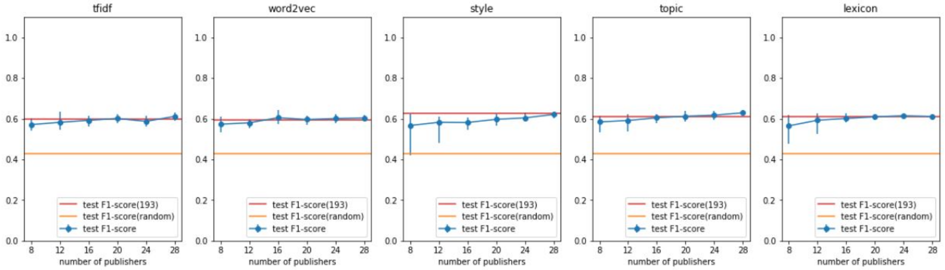


Figure 5.2: Performance of SVM with varying numbers of publishers (SemEval2019).

A general trend we observed was that average F1-scores increased slightly and the variance decreased as we included more publishers. For the writing style and lexicon features, this was more notable. However, the number of publishers did not seem to be the main factor here. For example, although the writing style feature had a low average F1-score when using 8 publishers, the variance were large. In at least one run, it achieved the same performance as using 193 publishers. Since in each run, we randomly chose publishers to be included, it seemed to be the case that some publishers provided more information than others. If those publishers were included, the performance was better. In the case when more publishers were used, there was a higher probability of including these "informative" publishers. Therefore, the average performance was better when more publishers were used. To sum up, the number of publishers was not the variable here. The quality of the publishers that were included was more likely to be influential.

Balancing of partisan publisher To test the effect of including both left and right publishers, we ran experiments with different ratios of hyperpartisan publishers. In the first setup, we used 7 hyperpartisan-right and 7 non-hyperpartisan publishers. The second setup included 6 hyperpartisan-right, 1 hyperpartisan-left, and 7 non-hyperpartisan publishers, etc. After sampling the publishers, we randomly sample 5K hyperpartisan articles and 5K non-hyperpartisan articles from all the available samples. Each setup was performed 10 times. Figure 5.3 shows the average F1-scores and error bars. None of the figures shows a better or worse performance depending on the ratio between left and right publishers.

We thus concluded that the reason for the lower performance of dpgMedia2019 was not due to the smaller training size, the smaller number of publishers, or the imbalance of partisan-left and partisan-right publishers. It is more likely due to the publishers that

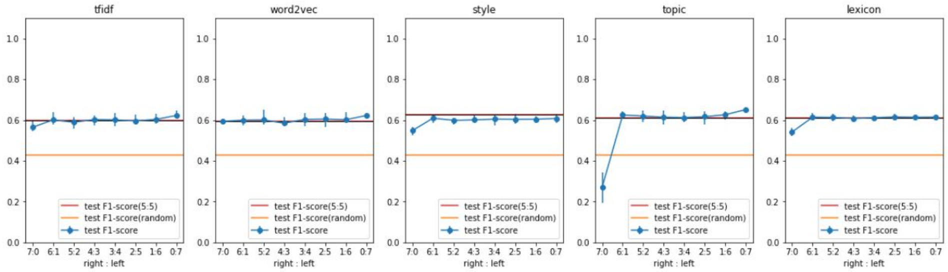


Figure 5.3: Performance of SVM with varying ratios between left and right publishers (SemEval2019).

we included. In Appendix C, we list other experiments that supported the claim that which publishers are used for training affects the performance significantly.

5.3. DISCUSSIONS

From Table 5.2 and 5.3, we answered RQ1. We benchmarked the performance using several feature and model combinations.

Ans: The F1-scores achieved were 63% and 49% for the two datasets. For both datasets, all classifiers outperformed the random and always-partisan baselines. This supports our hypothesis that the publisher-level labels are noisy but contain enough information to train a partisan news classifier.

When training the classifiers, we observed high cross-validation F1-scores across all features. The complete tables with validation F1-scores can be found in Appendix B. In summary, validation F1-scores were always higher than test F1-scores. For some models, the scores reached 85% to 90%. However, the performance did not generalize to the article-level test set. Since the validation set was a subset of the training set, and thus labeled by publisher partisanship, the problem was most likely due to the difference in labels between the validation and test set.

There are two sub-problems related to the labels that limited the learning process of the classifier. First, the classifiers we trained using the publisher-level labels were trained with the following objective:

"Given a news article, predict the partisanship of its publisher."

Under the current experimental setting, we do not know if the classifier learned to predict the publisher or partisanship of an article. The high validation F1-score seemed to signal that the classifier captured publisher information. Second, the validation F1-score was not a reliable metric to choose models. This is because that the metric was calculated from publisher-level labels that were partially incorrect. Even if our classifier could predict partisan articles perfectly, the validation F1-score would be imperfect because the labels do not distinguish between articles. Therefore, deciding model hyperparameters based on this metric is not ideal.

In the next chapter, we further investigate the problems we identify and attempt to mitigate them.

6

Error analysis: is it overfitting?

In this chapter, we further analyze the result we obtained in Chapter 5. The analysis was based on the observation that validation F1-scores were higher than test F1-scores. In a machine learning task, a high validation performance and low test performance can indicate two things. First, the validation set and test set come from different distributions and the validation set has a more similar distribution to the training set. For SemEval2019, that can indeed be one of the reasons since the test set was created to consist of different publishers from the training data. However, all the articles were collected from the same 11 publishers in dpgMedia2019. The performance gap observed in dpgMedia2019 thus indicated that this was not the main reason. The second possibility is that we overfit the hyperparameters on the validation set. This means that by choosing the model parameters based on the validation performance, we were biased towards the validation data. We can mitigate this overfitting by using more regularization (using a lower regularization factor in LR and SVM), forcing the model to be more generalizable. However, our datasets are different from general machine learning tasks, which lead to a third possibility, the model overfitted on the publisher label.

Before diving into the analyses, we examine again why we expect the publisher-level labels to work or not work. Imagine that there exists a feature space where the partisan articles and non-partisan articles can be separated perfectly. We illustrate this in Figure 6.1. Here we show a two-dimensional space for simplicity. The stars symbolize partisan articles and diamonds symbolize non-partisan articles. Publishers are coded in colors and numbered from 1 to 8. Publishers 1, 5, 7, and 8 are partisan and we assume that they have more partisan articles (more stars than diamonds). Publishers 2, 3, 4, and 6, on the other hand, have more non-partisan articles.

In this illustration, a partisan news detector should separate stars from diamonds, as shown in Figure 6.2a. However, our classifier could only learn from the colors, not the shapes. It thus tries to separate pink from blue, as illustrated in Figure 6.2b, and learns a different decision boundary.

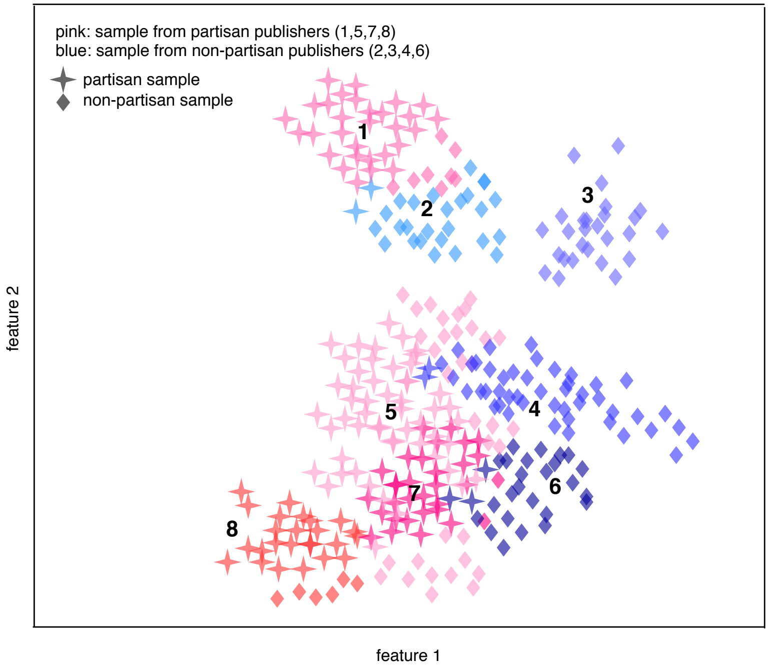
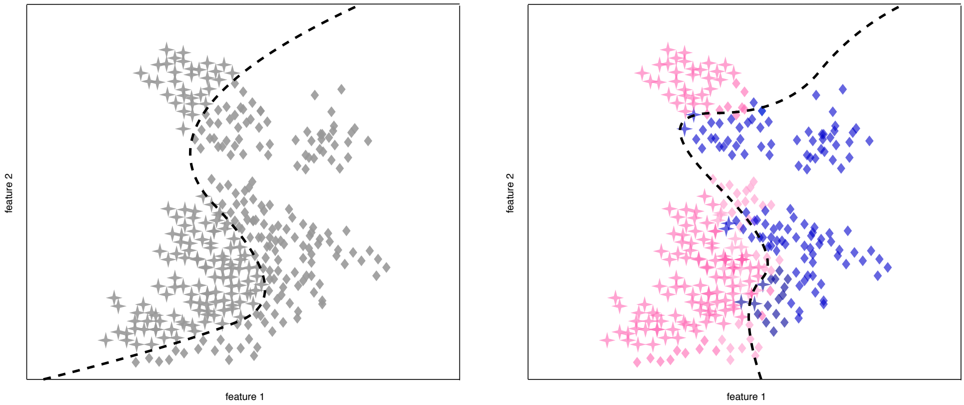


Figure 6.1: Illustration of news articles from 8 publishers in a hypothetical 2-dimensional space.



(a) True classification boundary of article partisanship.

(b) Classification boundary being learned from the publisher partisanship.

Figure 6.2: Illustration of how publisher-level annotations affect the classifier.

We see that whether we can learn the correct decision boundary depends on which samples we get from each publisher and how consistent the publishers are about the partisanship. As mentioned in the last chapter, the publisher-level annotation would train a classifier that has the objective of

"Given an article, predict the partisanship of its publisher.".

Although we have hoped that by collecting many publishers and forcing the classifiers to learn the common patterns between these distinct publishers, the pattern being learned would be partisanship, it is likely that the classifiers pick up publisher-specific patterns during training.

In this chapter, we first validate whether this is the case by hypothesizing that the classifiers cannot generalize to unseen publishers in section 6.1. Then, we attempt to improve the classifiers by selecting features that are less prone to overfit on publishers in section 6.2.

6.1. VALIDATION WITH UNSEEN PUBLISHERS

The validation set in the experiments in Chapter 5 consisted of a random subset of the training set. It thus contained the same publishers from the training set. The high validation F1-scores can thus be an indication that the classifiers pick up publisher patterns more than partisanship patterns. We thus hypothesize that if the classifiers need to predict for new publishers in the validation set, the performance would decrease. If that is the case, validating on unseen publishers should help us choose better models because the validation score would be a better indication of the performance of partisan news detection. We tested the following two hypotheses:

H1.1: The validation F1-score decreases if the validation set consists of different publishers from the training set.

H1.2: Using unseen publishers in the validation set helps select models that result in higher performance of detecting partisan news.

6.1.1. EXPERIMENTAL SETUP

The setup was similar to that in Chapter 5. We chose parameters with cross-validation and tested the final result on the test set. The main difference was that the training and validation folds in cross-validation had different publishers. To have a balanced class distribution under the limitation of non-overlapping publishers between training and validation folds, we used two folds instead of five during cross-validation for both datasets. For dpgMedia2019, a non-overlapping split resulted in imbalanced class due to the limited number of publishers. We therefore further down-sampled the majority class when training. For example, when we trained with articles from ADR, Trouw, and Het Parool and validated on articles from AD and VK, there were 10,494 articles from ADR. We thus downsample the articles from Trouw and Het Parool to the same number to train a balanced classifier. Therefore, in each cross-validation experiment, there were about 20K training data and 20K validation samples.

6.1.2. Results

Table 6.1 shows the validation F1-scores from previous experiments (random split), the new experiments (split with unseen publishers), and the relative difference between them for both datasets. Relative difference is computed as

$$\frac{x_{new} - x_{original}}{x_{original}} \times 100\% \tag{6.1}$$

Table 6.1: Comparison of cross-validation F1-scores using a random split of training and validation data and an unseen split that assigned different publishers in training and validation folds.

Features	SemEval2019			dpgMedia2019		
	random	unseen	diff (%)	random	unseen	diff (%)
N-gram	**90.61**	**65.40**	-27.82	**86.96**	58.67	**-32.55**
Word2Vec	78.36	55.48	-29.20	71.72	57.32	-20.08
Writing style	68.24	50.63	-25.81	70.58	56.04	-20.60
Topic	77.33	54.94	-28.95	67.12	54.02	-19.52
Lexicon	71.92	54.75	-23.87	67.18	51.12	-23.91
Combined	81.44	56.16	**-31.04**	77.58	**60.51**	-22.00

We observed decreased F1-scores of around 20% to 30% across all features for both datasets. This supported hypothesis H1.1. An example of this kind of overfitting is that some publishers tend to use their names in the articles a lot. Figure 6.3 shows two heat maps where we count the number of times each publisher name appeared in each article and then grouped them by publishers. We show the average number of times the word appears in an article in the left figure, which shows that some publishers use their names more on average in absolute frequency. We also show the frequency normalized by the total number of times the name appeared in all articles to study which publisher contributed most to the usage (right figure). Almost all publishers use their names more often compared to other publishers, as can be seen from the diagonal line. This means that a classifier might learn that a disproportionate use of certain names indicates partisanship, which would result in a high validation score but can hardly generalize to new articles.

Table 6.2 shows test F1-scores from random split, split with unseen publishers, and the relative difference between them for both datasets.

For the writing style and topic features, the test performance did not change. For other features, the change was inconsistent. Choosing models based on validation scores of unseen publishers did not help us select a better model for article-level prediction, which rejected Hypothesis H1.2. Moreover, although the new validation scores were

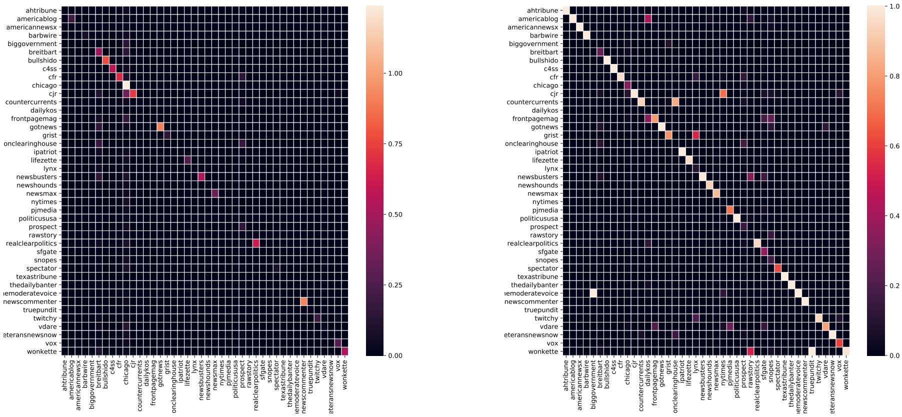


Figure 6.3: Left: Average number of using a publisher name (x-axis) per article grouped by publisher (y-axis). Right: same as left but normalized by the total number of times the publisher name appeared in all articles.

Table 6.2: Comparison of test F1-scores using a random split of training and validation data and an unseen split that assigned different publishers in training and validation folds.

Features	SemEval2019			dpgMedia2019		
	random	unseen	diff (%)	random	unseen	diff (%)
N-gram	59.80	55.49	-7.21	48.65	**49.04**	+0.82
Word2Vec	59.48	59.48	0.00	45.99	45.99	0.00
Writing style	**62.81**	**62.81**	0.00	44.05	44.05	0.00
Topic	60.20	60.96	**+1.26**	**48.80**	48.80	0.00
Lexicon	61.19	61.19	0.00	42.44	46.74	**+10.13**
Combined	60.32	59.41	-1.96	46.09	46.10	+0.02

closer to the test F1-scores, the correlation was low. A higher score on publisher-level labels did not imply a higher score on article-level labels. Therefore, evaluating classifiers using publisher-level labels is not indicative of the performance of partisan news detection.

6.2. FEATURE SELECTION BY PUBLISHER CLASSIFICATION

In the last section, we found that the classifiers were prone to fit to publishers instead of partisanship. Therefore, we hypothesized that by removing the features that were predictive of publishers, we could reduce overfitting and improve the performance of the classifiers. The hypothesis we tested was

H2: Classifiers learned from selected features that are less useful for predicting publishers have higher performance of partisan news detection compared to those learned from all features.

6.2.1. EXPERIMENT SETUP

To find the features that were important to publisher classification, we trained LR classifiers to predict the publisher of an article. We picked *N* publishers with the most articles and sample 1K articles from each of them to have a balanced class distribution. *N* was 10 for SemEval2019 and 6 for dpgMedia2019. We trained 2 classifiers on each dataset, one with n-grams and the other with the combined feature. We experimented with these two feature sets due to the large feature dimensions, which allowed more room for feature selection. After training, we used the learned parameters of LR to decide feature importance. Features with coefficients larger than the mean of all coefficients were considered important features. These features were removed when we trained our partisan news classifier.

6.2.2. RESULTS

We first show in Table 6.3 the accuracy achieved with the publisher classifiers. It shows that the classifiers were quite effective in finding the publisher of an article.

Table 6.3: Feature dimensions and accuracies achieved for publisher classification.

Feature + Model	SemEval2019		dpgMedia2019	
	Dim	Accuracy	Dim	Accuracy
Random baseline	–	10.00	–	16.67
N-gram + LR	517484	73.96	50000	57.76
Combined + LR	459	53.85	175	49.89

We then trained our partisan news classifiers using the unimportant features. Table **??** listed the test F1-scores of using the original features and feature that were unimportant to publisher classification. Under the dash line, we also show the performance achieved using features that were selected for publisher classification as comparison.

We observed conflicting results. For SemEval2019, using non-selected features improved the F1-scores on both feature sets. However, for dpgMedia2019, feature selection based on this method did not improve the performance. The reason could be that the selected features were also important in predicting partisanship. Removing them reduce overfitting, but the gain was not enough to compensate for the loss of information carried in those features. However, without further investigation, it is difficult to conclude. Hypothesis H2 was rejected as we did not find consistent evidence that this way of feature selection helped predict partisanship.

Table 6.4: Result of partisan news detection without feature selection, using selected features, and using non-selected features.

Features	Selection	SemEval2019		dpgMedia2019	
		Dim	F1-score	Dim	F1-score
N-gram	none	517,484	59.80	50,000	**49.23**
	non-selected	341,323	**63.38**	31,635	48.23
	selected	176,161	59.80	18,365	47.33
Combined	none	459	60.32	175	**46.09**
	non-selected	271	**61.33**	91	45.42
	selected	188	61.07	84	47,30

6.3. DISCUSSIONS

We have analyzed the overfitting of classifiers on publishers by ensuring that the validation set consists of unseen publishers. From our experiments, we concluded that validation scores on random publishers would drop significantly if we validate with new publishers. This, however, does not help select better classifiers. This cast doubt on previous research that reported performance using publisher-level labels. Kulkarni *et al.* [32] labeled their news articles using the partisanship of publishers. They randomly split the data into training and test set, and report test performance on several models. From our experiments, this evaluation method is likely to result in too optimistic performance because it is likely to overfit on publishers. On the other hand, Stein *et al.* [27] trained hyperpartisan news classifiers with articles by 6 publishers and reported performance on articles by 3 different publishers. From our experiment, this evaluation method would provide more practical performance scores, but cannot be used to compare different models as the scores are not indicative of which model perform better on the article-level labels.

We also attempted to mitigate the overfitting by selecting features that were less prone to overfit to publishers. However, we didn't observe a consistent improvement of using this selection method. We now turn to article-level labels and investigate how much performance we can gain from these cleaner labels.

7

Article-level labels

In Chapter 5, we set the benchmark of using publisher-level labels and investigate some properties of the datasets that can affect performance. In Chapter 6, we further analyze the reason for the poor performance, and investigate whether we can improve performance by reducing overfitting. We concluded that with the current features and models, we cannot further improve the performance using the publisher-level labels.

In this chapter, we study how much performance we can gain from using article-level labels to train our classifier. We aim to answer the second research question:

RQ2: What is the performance of predicting partisanship of articles with article-level labels compared to publisher-level labels?

We hypothesize that the fine-grained and clean article-level labels would bring performance increase across all features. In the following sections, we describe the experimental setup in section 7.1 and show the results in 7.2. Finally, we compare the performance of the two datasets and the two annotation levels in section 7.3.

7.1. Experimental setup

In the experiments, we use the article-level labeled parts of the datasets for training and testing the classifiers. We used 60% of the data for training and 40% for testing. The test set was the same as used in previous experiments. This ensured that the results were comparable between the two levels of annotations. Due to the class imbalance problem (37% and 26% partisan class for SemEval2019 and dpgMedia2019 respectively), we applied SMOTE or cost-sensitive weighting. We used 5-fold cross-validation to choose hyperparameters and the imbalance techniques based on F1-scores. Finally, the classifiers were trained using the whole training set and test on the test set.

7.2. RESULTS

7.2.1. SEMEVAL2019

Table 7.1 shows the result on SemEval2019. All features achieved an F1-score higher than 60%. The combined feature achieved the highest recall of 77.89% and F1-score of 72.55%. The precision and recall were comparable at around 60% to 70%.

Table 7.1: Results of predicting partisan news with models trained by article-level labels (SemEval2019).

Features + Model	Precision	Recall	F1-score
Random	37.21	50.71	42.92
Always partisan	36.90	100.00	53.91
N-gram + LR	**71.43**	57.89	63.95
Word2Vec + SVM	77.63	62.10	69.01
Writing style + LR	60.48	66.32	63.32
Topic + SVM	66.67	63.16	64.86
Lexicon + LR	57.02	72.63	63.89
Combined + LR	67.88	**77.89**	**72.55**

7.2.2. DPGMEDIA2019

Table 7.2 shows the result for dpgMedia2019. Most of the models outperformed the baselines. The n-gram and lexicon features performed worse than a baseline that always predicted partisan. The Word2Vec feature was achieved the best precision, recall and F1-score, reaching an F1-score of 58.96%.

Table 7.2: Results of predicting partisan news with models trained by article-level labels (dpgMedia2019).

Features + Model	Precision	Recall	F1-score
Random	26.50	50.35	34.72
Always partisan	26.24	100.00	41.57
N-gram + LR	50.94	33.33	40.30
Word2Vec + LR	**55.43**	**62.96**	**58.96**
Writing style + SVM	46.43	48.15	47.27
Topic + SVM	53.75	53.09	53.42
Lexicon + LR	31.71	48.15	38.24
Combined + SVM	53.09	53.09	53.09

7.3. DISCUSSIONS

With the results of article-level labels, we can now compare the two annotation levels. In Table 7.3, we list the F1-scores on both annotation levels, as well as the relative difference to quantitatively see the increase or decrease in performance. We observe that performance improved across all features for SemEval2019. The writing style feature, which overfit least on the publisher-level labels, had the smallest increase. For dpgMedia2019, there were two exceptions, n-grams and lexicon features. These two features performed worse than the baseline and had a decrease in F1-score. We think that the lexicon feature performed badly due to the small feature size of 7, which did not condition the problem well with small samples. For n-gram, the features might have overfitted on the publisher-level part of the data since we decided which terms to include based on that. We answered RQ2:

Ans: For SemEval2019, we achieved an F1-score of 72.55% with article-level labels. This was a 20% increase compared to that achieved by publisher-level labels. For dpgMedia2019, we achieved an F1-score of 58.96% with article-level labels, which was 28% better than publisher-level labels. Most feature sets had an increase of F1- score in both datasets.

Table 7.3: Comparison of test F1-scores on the two annotation levels.

Features	SemEval2019			dpgMedia2019		
	publisher	article	diff (%)	publisher	article	diff (%)
N-gram	59.80	63.95	+6.94	48.65	40.30	-17.16
Word2Vec	59.48	69.01	+15.32	45.99	**58.96**	**+28.20**
Writing style	**62.81**	63.32	+0.81	44.05	47.27	+7.31
Topic	60.20	64.86	+7.74	**48.80**	53.42	+9.47
Lexicon	61.19	63.89	+4.41	42.44	38.24	-9.90
Combined	60.32	**72.55**	**+20.28**	46.09	53.09	+15.19

Comparing the two datasets, we observed that the performance of dpgMedia2019 was lower than SemEval2019 on both annotation levels. We investigated possible reasons for a lower performance for the publisher-level labels in Chapter 5 and concluded that it was unlikely due to the data size, the number of publishers, and the left-right ratio of partisan publishers. The results on article-level labels also confirmed that some inherent problems made it harder to perform partisan news detection on dpgMedia2019, no matter which level of annotation. We list some possible reasons:

1. Lack of resources for the Dutch language: This is a general problem in NLP, where most research is done for the English language. The Dutch tokenizer and PoS-tagger have lower accuracies than English ones. We did not use lemmatizer for Dutch when extracting n-grams and topic features, possibly resulting in less effective features. The Dutch word embeddings that we used were trained with smaller

corpus, and we had very few Dutch lexicons to use. Therefore, all the features were less informative for the Dutch language.

2. Noisy article-level labels: For dpgMedia2019, we did not have "gold data" to perform standard quality control for the article-level annotations that were collected. This resulted in a low inter-rater agreement. This means that the article-level labels can be noisy as well. This would affect the training of the article-level classifiers because it has to learn from noisy labels. It would also affect the evaluation of both levels because the articles are more difficult to predict correctly.
3. Less severe media bias: The nature of the news landscape in the Netherlands is less partisan. This means that the general task of partisan news detection is more difficult because the language can be more subtle, which makes it harder to capture using simple features. Also, the publisher-level labels are more difficult to be applied as publishers are less consistent in their partisanship than those in the United States.

8
Conclusions

The main objective of the thesis was to evaluate how the annotation level of data would affect the performance of a partisan news detector. To achieve this, we compared the performance of several partisan news classifiers using diverse feature sets and classification models. This ensured that our conclusions were not limited by any specific choice of classifiers. In addition, we performed experiments on two datasets of different properties and languages to investigate how general the results are.

In this chapter, we conclude our work and explain what has been learned from it. Then, we point out the limitations of the thesis and directions for future work.

8.1. Conclusions and recommendations

At the beginning of our work, we discussed the difference between the annotation levels. We mentioned that publisher-level labels are easy to collect but noisy. We thus could expect a lower performance for classifiers that learn from publisher-level labels without performing any experiment. What we add to the research is a benchmark of the performance of publisher-level labels for future research to compare as a baseline, and the extent of increase in performance that one can expect if article-level labels are collected. We found that learning with publisher-level labels can perform better than random. On average, we can expect around 10% increase in F1-score if article-level labels are collected. The amount of increase depends on features and the quality of the labels. On the other hand, the recall is much higher for the publisher-level labels, which means that it can still be useful depending on the use case.

We also analyzed the overfitting of publisher-level labels. We found that if the validation set consists of the same publishers as the training set, the validation F1-scores are high, meaning that the classifiers overfit on the publishers. Using a validation set with unseen publishers resulted in a validation score closer to the performance achieved on article-level predictions. However, it did not help train or select a better classifier.

Finally, we analyzed some properties of the datasets. We showed on SemEval2019 that the performance of publisher-level labels is not affected by the size of training samples, the number of publishers included, and the balancing of left and right partisan publishers.

In summary, our research concludes the following about partisan news detection.

- Publisher-level labels are difficult to use to predict article-level partisanship. If only publisher-level labels are available, we expect F1-score better than random but hardly good enough for general practical use.
- Evaluating partisan news detectors on publisher-level labels is not indicative of performance on article-level labels. Therefore, without collecting any article-level labels, it's difficult to predict how well a model works in practice.
- A few hundreds of article-level labels can be quite useful, increasing F1-score by an average of 10%. The amount of increase depends on the feature and model being used, as well as on the quality of the labels that are collected.

From our conclusion, we would make the following recommendation to news organizations that intend to incorporate a partisan news detector in their systems, either as a filtering tool to select non-partisan news to be published or to show the partisanship of articles on user interfaces to better inform readers.

For an organization that does not want any partisan news to be published, detectors learned from the publisher-level labels are recommended because of the high recall. In this case, we are confident that the majority of partisan news articles would be detected, despite leaving us with few publishable articles. It is suitable as a pre-screening process followed by manual inspection to compensate for the low precision. On the other hand, detectors learned with article-level labels are more suitable in a general fully-automated system where it can improve the percentage of non-partisan news to be published. To be used as an informing tool on online news interfaces, high recall is undesirable because marking many non-partisan news as partisan would likely confuse readers more than informing them. However, this would require future empirical work that incorporates the classification results into interactive systems to study the acceptance of users. Finally, it is advisable to always collect some articles that are labeled on the article-level for evaluation. From our results, we observed that evaluation on the publisher-level labels does not indicate the performance on article-level labels.

8.2. LIMITATIONS AND FUTURE WORK

In this section, we discuss the limitations of our work and the degree of impact they had on our conclusions. Furthermore, we point out research directions that can provide more insights for the problem.

Correlation between publisher partisanship and article partisanship. One fundamental limitation of our work is the validity of using publisher partisanship to label article partisanship. We did not carefully examine whether the assumption of partisan publishers producing more partisan news and non-partisan ones producing more non-partisan news holds. Previous research on the US media has classified them into different political ideologies based on the articles they publish. They found a correlation between their estimations of media partisanship and public perception and surveys [12, 72]. This shows that partisanship exists in the articles which in aggregation reveals the partisanship of the publishers. Due to the complex dynamics between media and politics, which can vary between countries, we do not know whether we can assume the same for Dutch media. Although the assumption seems sensible, our perception of a publisher's partisanship can also come from other sources, such as the selection and coverage of news events, not necessarily the individual articles.

This casts doubt on whether the publisher-level labeling method is sensible in countries other than the United States. In extreme cases, the publisher-level labels might provide no information in predicting partisan articles. It is also doubtful whether it's valuable to study the publisher-level labels. Perhaps investigating how to more effectively use the limited article-level labels is more valuable for the task of partisan news detection.

Limited test data. We evaluated our classifiers on a test set with article-level labels. It is, however, small and might not be representative of the samples in the whole feature space. This is not a limitation specific to our research, but a problem that all machine learning tasks with small data size face. Our publish-level classifiers can be tested on all article-level labels instead of the 40% we used in our work. We used only 40% to have a fair comparison between the two annotation levels. We checked the performance of publisher-level classifiers tested on the entire article-level part of the datasets, and the results did not differ much. Therefore, this limitation did not influence our conclusions about which features perform better and the performance increase of article-level labels.

Limitations of datasets. The news in SemEval2019 was scraped from websites and facebook feeds of various sources. Although the authors tried to clean them, we found several articles that consist of advertisements, links, and abnormal texts. Some sources are quite scam-like, adding many catchwords such as "THIS JUST IN! THIS JUST IN!" in all of their articles. The articles are thus quite noisy and the features extracted were less informative. It is also doubtful what the classifiers learned in this case.

The articles in dpgMedia2019 are cleaner, as they are processed texts retrieved from the database. However, the labels in this dataset are less reliable. For the publisher-level part of the dataset, the partisanship scores of publishers were computed from a small number of readers that did not represent the full audience. Our method to average their political standpoints can also average out left and right-leaning readers. For example, a publisher with 55% extreme-right readers and 45% extreme left readers would be labeled non-partisan. However, it is possible that labeling this publisher as extreme-right, a result from the majority vote, is more sensible. It thus requires more research into the relationship between readership and media partisanship. For the article-level part of the dataset, we again suffered from sampling bias of the annotators. Also, due to resource constraints, we did not collect expert annotations as anchors to filter unreliable annotators. Therefore, we have a low inter-rater agreement and less confidence in the final label. When there are conflicting observations of results, we would thus trust the result from SemEval2019 more than dpgMedia2019.

8.2.1. Extensions and future directions

Dataset properties. From a data collection point of view, there are two future directions, one for each level of annotation. For the publisher-level part, it is useful to understand what properties of the dataset are important to training a good classifier. In our work, we investigated the size, the number of publishers, and whether the partisan publishers need to contain both left and right ones. We suggested that the "quality" of the publishers is most influential. However, we still don't know what contribute to the quality in our context. For example, is it the intensity of the partisanship of the publishers? In SemEval2019, center-left and center-right publishers were considered non-partisan. Perhaps excluding these publishers help reduce label noise. Other variables such as the coverage of the publisher (national or regional), size of publisher based on reader base can be explored as well. Studies about a systematic way to find which publishers are more useful to be included in the dataset would contribute to future dataset collection. Also, the number of articles we collect per publisher can affect the learning of the classifiers. In both datasets, we have some publishers with a large number of articles and many with few articles. It is possible that collecting an equal number of articles from each publisher helps alleviate overfitting on publishers.

For the article-level part, it will be useful to study the number of annotations needed per article to reach certain confidence in the labels, as well as how to learn from crowdsourced annotations. This is related to the work by Sheng *et al.* [77], who assessed the usability of repeated labeling when the label is imperfect, and how to use these labels for classifier training.

Weakly-supervised learning. We have approached the two levels of annotation from a dataset perspective. However, it can also be approached from the algorithmic perspective. How to use noisy labels to train good machine learning classifiers is related to the field of weakly-supervised learning [78]. For example, some theoretical models considered random noise in labels and tried to overcome the noise [79, 80]. Methods to combine the two levels of annotations are also potential. For example, using semi-supervised learning, we can propagate article-level labels to all articles. And if the propagated labels do not agree with the publisher-level labels, we discard the articles or correct the labels. The performance achieved by these methods can be compared with the performance we benchmarked in this work, which can further characterize what kind of noise is inherent in the publisher-level labels and how to better overcome it.

Explainable partisan news detector. Explainable classifiers are desirable because they help us understand what has been learned. This is useful in two ways. First, it gives us insights into how partisanship is expressed in news and what linguistic cues are more important. It assists news providers to avoid partisan language and informs readers to be more conscious of what is being consumed. Second, it improves the people's trust in the classifiers. Moreover, it helps error analysis. With explanations, we can investigate why publisher-level classifiers make more errors and further improve it. Explainability is especially important in partisan news detection because the bias in data can result in a biased classifier (it would be ironic if we want to detect bias but the classifier is itself biased).

Ways to achieve explainable classifiers include using white-box classifiers such as decision trees or logistic regression classifiers. RNNs with attention mechanism can also be visualized by looking at the attention weights to understand which words are considered important by the classifier to make decisions [81, 82].

BIBLIOGRAPHY

[1] J. Kiesel, M. Mestre, R. Shukla, E. Vincent, P. Adineh, D. Corney, B. Stein, and M. Potthast, *SemEval-2019 Task 4: Hyperpartisan News Detection,* in *Proceedings of The 13th International Workshop on Semantic Evaluation (SemEval 2019)* (Association for Computational Linguistics, 2019).

[2] C.-L. Yeh, B. Loni, M. Hendriks, H. Reinhardt, and A. Schuth, *Dpgmedia2019: A dutch news dataset for partisanship detection,* (2019), arXiv:1908.02322 .

[3] B. Loni, A. Schuth, V. Visse, M. van der Wees, L. de Haas, and J. Jansze, *Personalized push notifications for news recommendation,* in *Proceedings of the RecSys Workshop on Online Recommender Systems and User Modeling*, RecSys '19 (2019).

[4] C. R. Sunstein, *Republic.Com 2.0* (Princeton University Press, Princeton, NJ, USA, 2007).

[5] *Media Bias and Influence: Evidence from Newspaper Endorsements* (National Bureau of Economic Research., 2008).

[6] P. R. Center, *Public globally want unbiased news coverage,* (2018).

[7] D. D'Alessio and M. Allen, *Media bias in presidential elections: a meta-analysis,* Journal of communication **50**, 133 (2000).

[8] R. Krestel, A. Wall, and W. Nejdl, *Treehugger or petrolhead?: Identifying bias by comparing online news articles with political speeches,* in *Proceedings of the 21st International Conference on World Wide Web*, WWW '12 Companion (ACM, New York, NY, USA, 2012) pp. 547–548.

[9] D. Sáez-Trumper, C. Castillo, and M. Lalmas, *Social media news communities: gatekeeping, coverage, and statement bias,* in *CIKM* (2013).

[10] J. N. Druckman and M. Parkin, *The impact of media bias: How editorial slant affects voters,* The Journal of Politics **67**, 1030 (2005).

[11] M. Recasens, C. Danescu-Niculescu-Mizil, and D. Jurafsky, *Linguistic models for analyzing and detecting biased language,* in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (Association for Computational Linguistics, 2013) pp. 1650–1659.

[12] M. Gentzkow and J. Shapiro, *What drives media slant? evidence from u.s. daily newspapers,* Econometrica **78**, 35 (2010).

[13] C. Lin and Y. He, *Joint sentiment/topic model for sentiment analysis,* in *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09 (ACM, New York, NY, USA, 2009) pp. 375–384.

[14] A. Balahur, R. Steinberger, E. v. d. Goot, B. Pouliquen, and M. Kabadjov, *Opinion mining on newspaper quotations,* in *2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, Vol. 3 (2009) pp. 523–526.

[15] M. Jiang and S. Argamon, *Political leaning categorization by exploring subjectivities in political blogs,* in *In In Proceedings, 4th International Conference on Data Mining* (2008) pp. 647–653.

[16] B. Liu, *Sentiment Analysis and Opinion Mining* (Morgan & Claypool Publishers, 2012).

[17] A. Kennedy and D. Inkpen, *Sentiment classification of movie and product reviews using contextual valence shifters,* in *Proceedings of FINEXIN-05, Workshop on the Analysis of Informal and Formal Information Exchange during Negotiations* (2005).

[18] A. Balahur and R. Steinberger, *Rethinking sentiment analysis in the news: from theory to practice and back,* Proceeding of WOMSA **9** (2009).

[19] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, *Lexicon-based methods for sentiment analysis,* Comput. Linguist. **37**, 267 (2011).

[20] P. Anand, M. Walker, R. Abbott, J. E. F. Tree, R. Bowmani, and M. Minor, *Cats rule and dogs drool!: Classifying stance in online debate,* in *Proceedings of the 2Nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis*, WASSA '11 (Association for Computational Linguistics, Stroudsburg, PA, USA, 2011) pp. 1–9.

[21] D. Sridhar, J. Foulds, B. Huang, L. Getoor, and M. Walker, *Joint models of disagreement and stance in online debate,* in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (Association for Computational Linguistics, 2015) pp. 116–125.

[22] Y. Sim, B. D. L. Acree, J. H. Gross, and N. A. Smith, *Measuring ideological proportions in political speeches,* in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing* (Association for Computational Linguistics, Seattle, Washington, USA, 2013) pp. 91–101.

[23] M. Iyyer, P. Enns, J. Boyd-Graber, and P. Resnik, *Political ideology detection using recursive neural networks,* in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (Association for Computational Linguistics, 2014) pp. 1113–1122.

[24] S. M. Gerrish and D. M. Blei, *Predicting legislative roll calls from text,* in *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11 (Omnipress, USA, 2011) pp. 489–496.

[25] D. Preoţiuc-Pietro, Y. Liu, D. Hopkins, and L. Ungar, *Beyond binary labels: Political ideology prediction of twitter users,* in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (Association for Computational Linguistics, Vancouver, Canada, 2017) pp. 729–740.

[26] R. Oshikawa, J. Qian, and W. Y. Wang, *A survey on natural language processing for fake news detection,* CoRR **abs/1811.00770** (2018).

[27] B. Stein, M. Potthast, J. Kiesel, J. Bevendorff, and K. Reinartz, *A stylometric inquiry into hyperpartisan and fake news,* in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers* (2018) pp. 231–240.

[28] S. Park, M. Ko, J. Kim, Y. Liu, and J. Song, *The politics of comments: Predicting political orientation of news stories with commenters' sentiment patterns,* in *Proceedings of the ACM 2011 Conference on Computer Supported Cooperative Work*, CSCW '11 (ACM, New York, NY, USA, 2011) pp. 113–122.

[29] D. X. Zhou, P. Resnick, and Q. Mei, *Classifying the political leaning of news articles and users from user votes,* in *ICWSM* (2011).

[30] M. Gamon, S. Basu, D. Belenko, D. Fisher, M. Hurst, and A. C. König, *Blews: Using blogs to provide context for news articles,* in *2nd AAAI Conference on Weblogs and Social Media (ICWSM 2008)* (American Association for Artificial Intelligence, 2008).

[31] W.-H. Lin, T. Wilson, J. Wiebe, and A. Hauptmann, *Which side are you on?: Identifying perspectives at the document and sentence levels,* in *Proceedings of the Tenth Conference on Computational Natural Language Learning*, CoNLL-X '06 (Association for Computational Linguistics, Stroudsburg, PA, USA, 2006) pp. 109–116.

[32] V. Kulkarni, J. Ye, S. Skiena, and W. Y. Wang, *Multi-view models for political ideology detection of news articles,* in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (Association for Computational Linguistics, 2018) pp. 3518–3527.

[33] Y. Jiang, J. Petrak, X. Song, K. Bontcheva, and D. Maynard, *Team bertha von suttner at SemEval-2019 task 4: Hyperpartisan news detection using ELMo sentence representation convolutional network,* in *Proceedings of the 13th International Workshop on Semantic Evaluation* (Association for Computational Linguistics, Minneapolis, Minnesota, USA, 2019) pp. 840–844.

[34] V. Srivastava, A. Gupta, D. Prakash, S. K. Sahoo, R. R.R, and Y. H. Kim, *Vernon-fenwick at SemEval-2019 task 4: Hyperpartisan news detection using lexical and semantic features,* in *Proceedings of the 13th International Workshop on Semantic Evaluation* (Association for Computational Linguistics, Minneapolis, Minnesota, USA, 2019) pp. 1078–1082.

[35] K. Hanawa, S. Sasaki, H. Ouchi, J. Suzuki, and K. Inui, *The sally smedley hyperpartisan news detector at SemEval-2019 task 4,* in *Proceedings of the 13th International Workshop on Semantic Evaluation* (Association for Computational Linguistics, Minneapolis, Minnesota, USA, 2019) pp. 1057–1061.

[36] B. Pang, L. Lee, and S. Vaithyanathan, *Thumbs up?: Sentiment classification using machine learning techniques,* in *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP '02 (Association for Computational Linguistics, Stroudsburg, PA, USA, 2002) pp. 79–86.

[37] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, *Distributed representations of words and phrases and their compositionality,* in *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13 (Curran Associates Inc., USA, 2013) pp. 3111–3119.

[38] J. Pennington, R. Socher, and C. Manning, *Glove: Global vectors for word representation,* in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (Association for Computational Linguistics, 2014) pp. 1532–1543.

[39] S. Baccianella, A. Esuli, and F. Sebastiani, *Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining,* in *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)* (European Languages Resources Association (ELRA), 2010).

[40] P. J. Stone, D. C. Dunphy, M. S. Smith, and D. M. Ogilvie, *The General Inquirer: A Computer Approach to Content Analysis*, Stone et al. (MIT Press, Cambridge, MA, 1966).

[41] M. I. A. G. James W. Pennebaker, Cindy K. Chung and R. J. Booth, *The development and psychometric properties of liwc2007,* .

[42] V. Pérez-Rosas, B. Kleinberg, A. Lefevre, and R. Mihalcea, *Automatic detection of fake news,* in *Proceedings of the 27th International Conference on Computational Linguistics* (Association for Computational Linguistics, Santa Fe, New Mexico, USA, 2018) pp. 3391–3401.

[43] H. Rashkin, E. Choi, J. Y. Jang, S. Volkova, and Y. Choi, *Truth of varying shades: Analyzing language in fake news and political fact-checking,* in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (Association for Computational Linguistics, Copenhagen, Denmark, 2017) pp. 2931–2937.

[44] Y. Kim, *Convolutional neural networks for sentence classification,* in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (Association for Computational Linguistics, 2014) pp. 1746–1751.

[45] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, *Hierarchical attention networks for document classification,* in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (ACL, 2016) pp. 1480–1489.

[46] C. Zhang, A. Rajendran, and M. Abdul-Mageed, *UBC-NLP at SemEval-2019 task 4: Hyperpartisan news detection with attention-based bi-LSTMs,* in *Proceedings of the 13th International Workshop on Semantic Evaluation* (Association for Computational Linguistics, Minneapolis, Minnesota, USA, 2019) pp. 1072–1077.

[47] A. Ahmed and E. P. Xing, *Staying informed: Supervised and semi-supervised multi-view topical analysis of ideological perspective,* in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10 (Association for Computational Linguistics, Stroudsburg, PA, USA, 2010) pp. 1140–1150.

[48] J. Eisenstein and E. Xing, *The CMU 2008 Political Blog Corpus*, Tech. Rep. (Carnegie Mellon University, 2010).

[49] T. Yano, P. Resnik, and N. A. Smith, *Shedding (a thousand points of) light on biased language,* in *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk* (Association for Computational Linguistics, Los Angeles, 2010) pp. 152–158.

[50] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, *Deep contextualized word representations,* in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (Association for Computational Linguistics, 2018) pp. 2227–2237.

[51] D. Cer, Y. Yang, S.-y. Kong, N. Hua, N. Limtiaco, R. St. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, B. Strope, and R. Kurzweil, *Universal sentence encoder for English,* in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations* (Association for Computational Linguistics, Brussels, Belgium, 2018) pp. 169–174.

[52] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *Bert: Pre-training of deep bidirectional transformers for language understanding,* arXiv preprint arXiv:1810.04805 (2018).

[53] C.-L. Yeh, B. Loni, and A. Schuth, *Tom jumbo-grumbo at SemEval-2019 task 4: Hyperpartisan news detection with GloVe vectors and SVM,* in *Proceedings of the 13th International Workshop on Semantic Evaluation* (Association for Computational Linguistics, Minneapolis, Minnesota, USA, 2019) pp. 1067–1071.

[54] M. Honnibal and I. Montani, *spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing,* To appear (2017).

[55] M. Fares, A. Kutuzov, S. Oepen, and E. Velldal, *Word vectors, reuse, and replicability: Towards a community repository of large-text resources,* in *Proceedings of the 21st Nordic Conference on Computational Linguistics, NoDaLiDa, 22-24 May 2017, Gothenburg, Sweden*, 131 (Linköping University Electronic Press, Linköpings universitet, 2017) pp. 271–276.

[56] R. Řehůřek and P. Sojka, *Software Framework for Topic Modelling with Large Corpora,* in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks* (ELRA, Valletta, Malta, 2010) pp. 45–50, http://is.muni.cz/publication/884893/en.

[57] T. Wilson, J. Wiebe, and P. Hoffmann, *Recognizing contextual polarity in phrase-level sentiment analysis,* in *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05 (Association for Computational Linguistics, Stroudsburg, PA, USA, 2005) pp. 347–354.

[58] M. Hu and B. Liu, *Mining and summarizing customer reviews,* in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04 (ACM, New York, NY, USA, 2004) pp. 168–177.

[59] J. Hooper, *On Assertive Predicates*, Indiana University Linguistics Club (Indiana University Linguistics Club, 1974).

[60] G. Thompson, Language in Society **37**, 138 (2008).

[61] L. Karttunen, *Implicative verbs,* Language **47**, 340 (1971).

[62] J. Graham, J. Haidt, and B. Nosek, *Liberals and conservatives rely on different sets of moral foundations,* Journal of personality and social psychology **96**, 1029 (2009).

[63] D. Fulgoni, J. Carpenter, L. Ungar, and D. Preoţiuc-Pietro, *An empirical exploration of moral foundations theory in partisan news sources,* in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)* (European Language Resources Association (ELRA), Portorož, Slovenia, 2016) pp. 3730–3736.

[64] Y. Chen and S. Skiena, *Building sentiment lexicons for all major languages,* in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (Association for Computational Linguistics, Baltimore, Maryland, 2014) pp. 383–389.

[65] V. Jijkoun and K. Hofmann, *Generating a non-english subjectivity lexicon: Relations that matter,* in *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '09 (Association for Computational Linguistics, Stroudsburg, PA, USA, 2009) pp. 398–405.

[66] H. He and E. A. Garcia, *Learning from imbalanced data,* IEEE Trans. on Knowl. and Data Eng. **21**, 1263 (2009).

[67] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, *Smote: Synthetic minority over-sampling technique,* J. Artif. Int. Res. **16**, 321 (2002).

[68] G. Lemaître, F. Nogueira, and C. K. Aridas, *Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning,* Journal of Machine Learning Research **18**, 1 (2017).

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

[70] S. Mullainathan and A. Shleifer, *The market for news,* American Economic Review **95**, 1031 (2005).

[71] M. Gentzkow and J. M. Shapiro, *Ideological Segregation Online and Offline*, NBER Working Papers 15916 (National Bureau of Economic Research, Inc, 2010).

[72] C. Budak, S. Goel, and J. M. Rao, *Fair and balanced? quantifying media bias through crowdsourced content analysis,* Public Opinion Quarterly **80**, 250 (2016).

[73] Y. Jiang, X. Song, J. Harrison, S. Quegan, and D. Maynard, *Comparing attitudes to climate change in the media using sentiment analysis based on latent dirichlet allocation,* in *Proceedings of the 2017 EMNLP Workshop: Natural Language Processing meets Journalism* (Association for Computational Linguistics, 2017) pp. 25–30.

[74] P. R. Center, *News media and political attitudes in the netherlands,* (2018).

[75] R. Snow, B. O'Connor, D. Jurafsky, and A. Ng, *Cheap and fast – but is it good? evaluating non-expert annotations for natural language tasks,* in *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing* (Association for Computational Linguistics, Honolulu, Hawaii, 2008) pp. 254–263.

[76] E. Vincent and M. Mestre, *Crowdsourced measure of news articles bias: Assessing contributors' reliability,* in *Proceedings of the 1st Workshop on Subjectivity, Ambiguity and Disagreement in Crowdsourcing, and Short Paper Proceedings of the 1st Workshop on Disentangling the Relation Between Crowdsourcing and Bias Management (SAD 2018 and CrowdBias 2018) co-located the 6th AAAI Conference on Human Computation and Crowdsourcing (HCOMP 2018), Zürich, Switzerland, July 5, 2018.* (2018) pp. 1–10.

[77] V. S. Sheng, F. Provost, and P. G. Ipeirotis, *Get another label? improving data quality and data mining using multiple, noisy labelers,* in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08 (ACM, New York, NY, USA, 2008) pp. 614–622.

[78] Z.-H. Zhou, *A brief introduction to weakly supervised learning,* National Science Review **5**, 44 (2017), http://oup.prod.sis.lan/nsr/article-pdf/5/1/44/24164438/nwx106.pdf .

[79] B. Frenay and M. Verleysen, *Classification in the presence of label noise: A survey,* IEEE Transactions on Neural Networks and Learning Systems **25**, 845 (2014).

[80] N. Natarajan, I. S. Dhillon, P. K. Ravikumar, and A. Tewari, *Learning with noisy labels,* in *Advances in Neural Information Processing Systems 26*, edited by C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (Curran Associates, Inc., 2013) pp. 1196–1204.

[81] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao, and B. Xu, *Attention-based bidirectional long short-term memory networks for relation classification,* in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (Association for Computational Linguistics, Berlin, Germany, 2016) pp. 207–212.

[82] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, *Attention is all you need,* in *Advances in Neural Information Processing Systems 30*, edited by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Curran Associates, Inc., 2017) pp. 5998–6008.

A

APPENDIX

A.1. SURVEY QUESTIONS

Q1-Q3 were asked per article; Q4 was asked per annotator. A translation into English follows the original questions.

A.1.1. ORIGINAL DUTCH QUESTIONS

- **Q1**: Over het algemeen, hoe bevooroordeeld vindt u dit artikel? Een artikel dat bevooroordeeld is, is voor of tegen een persoon of groep. N.B. Een artikel kan gaan over controversiële onderwerpen, zoals politiek, maar blijft redelijk neutraal.

  1. Onbevooroordeeld
  2. Redelijk onbevooroordeeld
  3. Enigszins bevooroordeeld
  4. Bevooroordeeld
  5. Extreem bevooroordeeld
  6. Onmogelijk om te bepalen

- **Q2**: Als u vindt dat dit artikel bevooroordeeld is, ten gunste van welke politieke richting vindt u dit artikel geschreven? (U kunt meerdere antwoorden kiezen)

  1. Links
  2. Rechts
  3. Progressief
  4. Conservatief
  5. Anders, namelijk: OPEN
  6. Niet van toepassing op dit artikel

7. Ik weet het niet

- **Q3**: Als u vindt dat het artikel bevooroordeeld is, pro of anti wie of wat vindt u dit artikel? (bijvoorbeeld pro-PVV, pro-conservatieven, pro-kapitalist, anti-Trump, anti-moslims, anti-atheïst). (U kunt meerdere antwoorden kiezen)

  – Pro:

  – Anti:

- **Q4**: Hoe zou u uw eigen politieke standpunt bepalen?

  1. Extreemlinks
  2. (gematigd-)links
  3. Neutraal
  4. (gematigd-)rechts
  5. Extreemrechts

A.1.2. ENGLISH TRANSLATIONS

- **Q1**: Overall, how biased is this article? An article that is biased is for or against a person or group. Note that an article can talk about contentious topics, like politics, but remains fairly neutral.

  1. Unbiased
  2. Fairly unbiased
  3. Somewhat biased
  4. Biased
  5. Extremely biased
  6. Not possible to decide

- **Q2**: If you find the article biased, which political direction do you find this article in favor of? (You can choose multiple answers)

  1. Left
  2. Right
  3. Progressive
  4. Conservative
  5. Others
  6. Not applicable to the article
  7. I don't know

- **Q3**: If you find the article biased, indicate who or what the article is biased in favor of ('pro') and/or against ('anti')? (for example pro-PVV, pro-conservative, pro-capitalist, anti-Trump, anti-Muslims, anti-atheist). (You can have multiple answers)

  - Pro:
  - Anti:

- **Q4**: How would you determine your own political position?

  1. Extreme-left
  2. (moderate)left
  3. Neutral
  4. (moderate)right
  5. Extreme-right

B

Appendix

Complete experiment results complementing chapter 5, 6, and 7.

B.1. Results of learning from publisher-level labels

Table B.1 and Table B.2 are the results of learning from publisher-level labels with a random split of publishers (Chapter 5) and non-overlapping split (Chapter 6).

B.2. Results of learning from article-level labels

Tables B.3 and B.4 show results of learning from articles-level labels on SemEval2019. Tables B.5 and B.6 show the same for dpgMedia2019. Due to the imbalanced data problem, we applied SMOTE (S) and cost-sensitive weighting (C) to have a more balanced decision boundary. For each feature and model, we show the better result from the two imbalance techniques (S or C), and use that for the final test.

Table B.1: Results of training with publisher-level labels. UP (unseen publishers) means that the validation set consists of different publishers from the training set (SemEval2019).

Features	Dim	Cross-validation F1-score		Test		
				Precision	Recall	F1-score
		LR	SVM			
N-gram	517,484	90.54	90.61	43.69	94.74	59.80
	50,000	90.38	90.41			
N-gram (UP)	517,484	59.59	59.72	39.06	95.79	55.49
	50,000	65.39	65.40			
Word2vec	300	78.25	78.36	43.13	95.79	59.48
Word2vec (UP)	300	55.35	55.48	43.13	95.79	59.48
Writing style	30	64.54	68.24	51.70	80.00	62.81
Writing style (UP)	30	49.81	50.63	51.70	80.00	62.81
Topic	20	75.37	75.54	44.12	94.74	60.20
	30	76.81	76.91			
	40	77.28	77.33			
Topic (UP)	20	52.24	52.59	45.18	93.68	60.96
	30	54.55	54.84			
	40	54.94	54.77			
Lexicon	89	71.74	71.92	47.40	86.32	61.19
Lexicon (UP)	89	54.69	54.75	47.40	86.32	61.19
Combined	459	80.66	81.44	43.81	96.84	60.32
Combined (UP)	459	56.16	55.43	43.27	94.74	59.41

Table B.2: Results of training with publisher-level labels. UP (unseen publishers) means that the validation set consists of different publishers from the training set (dpgMedia2019).

Features	Dim	Cross-validation F1-score		Test		
		LR	SVM	Precision	Recall	F1-score
N-gram	520,874	86.88	86.96	35.39	77.78	48.65
	50,000	86.34	86.36			
N-gram (UP)	520,874	57.88	57.95	35.56	79.01	49.04
	50,000	58.67	58.53			
Word2vec	100	71.71	71.72	32.64	77.78	45.99
Word2vec (UP)	100	57.28	57.32	32.64	77.78	45.99
Writing style	28	65.79	70.58	34.25	61.73	44.05
Writing style (UP)	28	55.22	56.04	34.25	61.73	44.05
Topic	20	66.15	65.30	33.81	87.65	48.80
	30	65.61	65.62			
	40	66.96	67.12			
Topic (UP)	20	52.66	52.73	33.81	87.65	48.80
	30	53.33	53.72			
	40	53.89	54.02			
Lexicon	7	67.18	59.84	27.03	98.77	42.44
Lexicon (UP)	7	51.12	50.81	32.81	83.95	46.74
Combined	175	75.61	77.58	34.57	69.14	46.09
Combined (UP)	175	57.67	60.51	34.57	69.14	46.10

Table B.3: Results of training with article-level labels using LR (SemEval2019).

Features	Dim	Cross-validation F1-score	Test Precision	Test Recall	Test F1-score
N-gram	517,484	62.29 (C)	71.43	57.89	63.95
	50,000	66.36 (C)			
Word2Vec	300	67.84 (S)	57.26	74.74	64.84
Writing style	30	66.70 (S)	60.48	66.32	63.32
Topic	20	65.21 (S)	57.76	70.53	63.51
	30	61.36 (S)			
	40	65.90 (S)			
Lexicon	89	67.43 (S)	57.02	72.63	63.89
Combined	459	70.68 (C)	67.88	77.89	72.55

Table B.4: Results of training with article-level labels using SVM (SemEval2019).

Features	Dim	Cross-validation		Test	
		F1-score	Precision	Recall	F1-score
N-gram	517,484	53.96 (C)	0.00	0.00	0.00
	50,000	53.96 (C)			
Word2Vec	300	68.74 (S)	77.63	62.10	69.01
Writing style	30	64.27 (S)	64.37	58.95	61.54
Topic	20	64.09 (S)	66.67	63.16	64.86
	30	67.27 (S)			
	40	67.63 (C)			
Lexicon	89	67.27 (S)	69.57	67.37	68.45
Combined	459	69.97 (C)	80.26	64.21	71.35

Table B.5: Results of training with article-level labels using LR (dpgMedia2019).

Features	Dim	Cross-validation		Test	
		F1-score	Precision	Recall	F1-score
N-gram	520,874	36.94 (S)	50.94	33.33	40.30
	50,000	42.55 (C)			
Word2vec	100	57.71 (C)	55.43	62.96	58.96
Writing style	28	51.21 (S)	41.88	60.49	49.49
Topic	20	51.68 (C)	50.48	65.43	56.99
	30	55.31 (S)			
	40	54.65 (S)			
Lexicon	7	43.65 (S)	31.71	48.15	38.24
Combined	165	56.33 (C)	48.48	59.26	53.33

Table B.6: Results of training with article-level labels using SVM (dpgMedia2019).

Features	Dim	Cross-validation F1-score	Test Precision	Test Recall	Test F1-score
N-gram	520,874 50,000	41.45 (C) 41.45 (C)	0.26	1.00	41.75
Word2Vec	100	55.38 (S)	50.91	34.57	41.18
Writing style	28	56.60 (C)	46.43	48.15	47.27
Topic	20 30 40	55.09 (C) 55.61 (S) 53.61 (C)	53.75	53.09	53.42
Lexicon	7	43.26 (C)	33.33	50.62	40.20
Combined	165	58.36 (C)	53.09	53.09	53.09

C
Appendix

In Chapter 5, we stated that which publishers were included in the dataset affected the performance of classifiers trained with publisher-level labels. We concluded this due to some experiments we performed on the validation set of SemEval2019. As we explained in Chapter 4, Kiesel *et al.* [1] split the publisher-level part into training and validation set of different publisher when they created the dataset. We used only the training set for all our experiments. We can also treat the validation set as another training set that contains a whole new set of publishers, and redo all experiments. Here we show our results.

C.1. Training with validation set of SemEval2019

Table C.1: Results of training with publisher-level labels (SemEval2019-validation set).

Features	Dim	Cross-validation F1-score		Test Precision	Recall	F1-score
		LR	SVM			
N-gram	50K	90.38	90.41	40.72	**94.74**	**56.96**
Word2vec	300	75.33	75.36	39.11	92.63	55.00
Writing style	30	62.42	65.41	37.25	80.00	50.84
	20	68.72	68.71			
Topic	30	70.22	70.35	37.80	83.15	51.97
	40	70.22	70.39			
Lexicon	89	66.57	66.79	**41.92**	87.37	56.66
Combined	454	76.50	78.73	36.24	87.36	51.23

We show the result with random validation publishers in Table C.1 and with unseen publishers in Table C.2. The main difference from the result of the training set was that the F1-scores were consistently lower across all features. This means that the publishers in the validation set are in general more difficult to learn from. Possibly because that they are more inconsistent in their partisanship.

Table C.2: Results of training with publisher-level labels where validation F1-score is measured on unseen publishers (SemEval2019-validation set).

Features	Dim	Cross-validation		Test		
		F1-score		Precision	Recall	F1-score
		LR	SVM			
N-gram	50K	65.39	65.40	39.06	**95.79**	55.49
Word2vec	300	62.31	62.27	38.77	92.63	54.66
Writing style	30	61.73	59.97	35.24	77.89	48.52
Topic	20	61.70	62.21	37.80	83.15	51.97
	30	61.00	61.07			
	40	62.72	62.75			
Lexicon	89	59.84	59.49	**41.21**	86.32	**55.78**
Combined	449	64.12	63.54	37.61	92.63	53.50