Universiteit Leiden
The Netherlands

Opleiding Informatica

Knowing the difference between news and opinion

An explorative research project into classifying news and opinion

David Bouter

Supervisors:
Peter van der Putten & Suzan Verberne

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl

20/01/2020

Abstract

Many people often misidentify news articles as opinion articles, or the other way around. This thesis explores multiple techniques to use classifiers to identify these articles automatically. Additionally we present the patterns the classifiers found in these articles. The goal of this research is to explore the possibilities of using classification to distinguish news from opinion articles. We conclude that the best functioning classifier is the Support Vector Machine algorithm with a radial kernel combined with a tf-idf word representation. We also show that the classifiers can find patterns in news and opinion articles. The strongest patterns in news articles are keywords that reference a time indication and those announcing a quote. The strongest patterns in the opinion articles announced a proposition, argumentation and conclusion. Lastly we explore the possibilities of using BERTje for word embedding. We find that BERTje can be a very powerful tool but its use was limited by the scope of this thesis and hardware requirements. The usage of BERTje is very promising but will need further research.

Contents

1 Introduction

Journalism has changed over the years. In the past the most important source of news was the newspaper. With the rise of technology a lot of news sources can now be found on television, the radio or on the internet. Anyone with an internet connection can write an article or express their opinion and this creates a problem: how can we be sure that the article we are reading is meant to report factual news or tries to voice an opinion? Sources do not always label their articles as news or opinion and a large portion of people admit to having a hard time making the correct distinctions between the two, especially for online content. Currently according to the Pew Research Center over 70% of American adults incorrectly identify an opinion piece as factual news (Mitchell, 2018). This can result in falsely changed views of the world, whether it was intended by the writer or not, which can also lead to false interpretation of facts. The existence of these problems are confirmed by Loker (2018).

We do not mean to argue that opinion pieces are problematic but we want to showcase the impact of incorrectly thinking an opinion is a fact. In the same way we also do not want readers to think facts stated in news articles are opinions because that could also cause problems. Especially on social media these problems can be additionally treacherous because content that is recommended is often based on the previous behaviour of the consumer (Bright, 2008). This means that when the consumer often reads articles from sources that do not label their articles properly they will likely encounter more of such articles. This does not prove the existence of a filter bubble, which is a disputed term because it suggests that readers are not susceptible for any opposing information. It merely implies that the personalisation on media websites can cause readers that read articles with vague news and opinion labeling to encounter more of those articles.

Most articles people read on the internet are provided by recommendation algorithms that aim to increase the time spend of the user on the website (Schelling et al., 2020). This means those algorithms prioritise providing articles that keep people's attention over providing articles that have high quality of information. This thesis branches from the project Reverb Channel, created by ACED, which tries to reverse the role of these algorithms. For this thesis we want to give these algorithms an informing role to help people understand what they are reading, increasing the quality of that information. This brings us to our research questions which are: to what extent can we classify Dutch text documents as news or opinion using machine learning? And: can we find patterns in the results with which we can explain the behaviour of the classifier? We will further discuss these questions in the experiments and results section.

In the past news and opinion classification has already been researched but the emphasis has mostly been on trying to achieve a high accuracy. Our contribution to the research in opinion and news classification will be that we compare the performance of many different classifiers, use an approach without lexical features to keep the models as general as possible and then try to explain the behaviour of the classifiers. Then we compare its behaviour to what we expect to see in the real world when someone tries to separate news from opinion. In this way we hope to broaden our understandings of what the key elements of news and opinion articles are so that those understandings can be applied in future research or applications.

1.1 Thesis overview

In the remainder of this paper you will first find a "related work" section where the history of opinion and news journalism is explored and some past work done in opinion classification is discussed. This is followed by an "experiments" section. Here our approach and the results of all our experiments will be elaborated in more detail. In the discussion section that follows we will try to put the results into context and explain the behaviour of the classifiers. Lastly there will be a conclusion to recap our findings.

This thesis is written as a bachelor end project at Leiden University Faculty of Science with supervision from Peter van der Putten and Suzan Verberne. Our database was provided by ACED. As mentioned before, this thesis is a project in the context of the Reverb Channel which was founded by ACED. The Reverb Channel is a research program that explores the impact of the use of data mining and machine learning in digital news media.

2 Related work

In this section we will analyse previous research done which is related to news and opinion classification. First we will define the difference between news and opinion, then we will review the history of news and opinion. Lastly, we will examine previous news and opinion classifiers that have been created by other researchers.

2.1 News versus opinion

The news we see on TV, in the paper or on social media often shapes our views on the outside world. It is therefore important that we know what kind of media we are consuming. Especially with the rise of the internet where everyone is constantly fighting for our attention, it is important to know what kind of actions or thoughts these articles are trying to impose on us because it decides what information is used to change our understandings and perspectives. That is why we are examining the distinction between news and opinion journalism.

First let us discuss the difference between news and opinions. In theory, news attempts to inform the news reader about a recent event while opinion journalism tries to convince the reader of a point of view. In practice however things tend to be different because news and opinion are not completely binary concepts. Usually, it involves a scale with a grey area. An example of such a grey area would be news analysis. A news analysis article often shows an expert's interpretation of a recent event meaning it provides facts and evidence but also conclusions drawn from those facts by the expert. These conclusions do not have to be opinions because they are based on facts, but because the future is unknown they can turn out not to be true.

Though the existence of grey areas might make the distinction between these article types a bit harder, it is still important that readers know what they are reading: they must not think they are reading news when instead they are reading an opinion. Therefore articles should be labelled correctly to prevent confusion. This does not mean that opinion journalism is bad journalism. After all, it has a role in creating context and perspectives on subjects and challenge their reader's ideas. However for opinion journalism to be valuable it is important that the reader is well informed and open minded, but as mentioned before, this is often not the case.

2.2 The rise of internet throughout the history of journalism

The distinction between fact and opinion in the news used to be a lot simpler in the past. Social media did not exist yet and newspapers, radio and TV made a clear distinction between opinion journalism and news. They would use terms such as "op-ed", which is short for "opposite the editorial page". This tells the reader that the article expresses an opinion of an author usually not involved with the editorial board. With the rise of the internet the distinction between news and opinion has become increasingly vague for the reader. This lack of clarity allows writers of articles to take advantage of readers. It can be used to disguise perspectives as facts to make opinions more believable. This can encourage the reader to take specific actions or to be engaged with the article for a longer period of time - which can be used to make revenue with ad sales. Also important to note is that not all internet journalism has to be done by professional journalists. Amateur writers can make up their own news and post it on different websites. On websites such as Facebook or Twitter anyone can post anything, meaning articles from non-verified sources, written by non-professionals, can be written.

According to Fortunati et al. (2009) the role of journalists has changed with the rise of the internet. They claim that internet journalists generally seem to have weaker ethics when it comes to fact checking and that they seem to partially sacrifice accuracy for speed, making their journalism less reliable. This behaviour can be very different on an individual level and on a group level, such as certain countries or news sources that have certain policies about journalism. It is, however, argued that this is a general trend we are seeing in internet journalism.

Not only internet journalism has changed over the years. According to the American Press Institute (Loker, 2018), television journalism faces similar problems. According to their survey only 57% of the participants thought it was easy to distinct news from opinion when watching broadcast television. Even less people were able to make this distinction when it came to radio journalism and the lowest scoring medium was online news websites with only 43% of people that had no hard time making the distinction. To conclude, we see that over the years and especially online the distinction between news and opinion keeps getting harder to make.

2.3 Opinion classification

Having seen the importance of recognising opinion and news, the question rises whether we can make an automatic process that can distinguish between the two. To answer that question, we will be looking at earlier research in opinion classification, done by J. Wiebe et al. (Wiebe et al., 1999), (Wiebe et al., 2000), (Wiebe et al., 2004). Wiebe and colleagues worked on a method to distinguish between factual and opinion journalism. Their goal was to classify subjectivity for text categorisation and information extraction. Wiebe et al. reported that there is a link between lexical features and the subjectivity of texts on both a sentence level and a word level. They concluded that opinion classification could be automated and be applied in real time internet usage.

Yu & Hatzivassiloglou (2003) found accuracy scores of up to 97% when classifying for opinions. This is a good sign for this thesis as it shows that this type of classification can yield good results. The features they used to classify for news and opinion included properties such as polarity. The features also included the words of the documents, those features could be prone to overfitting. In our research we found that the classifier tends to train on leaking variables such as authors

or publishers, if it is still present in the articles. This could very well be the case for Yu and Hatzivassiloglou because their accuracy score is very high however, there was no mention of any analysis of these problems in their paper. For us this means we should be extra careful when analysing our features and checking them for noise.

A much more recent paper about news and opinion classification found very compelling results using deep learning models and Support Vector Machine algorithms with a radial kernel (Alhindi et al., 2020). From now on we will use the abbreviation SVM and if no kernel is specified we assume the radial kernel. The researchers found that these techniques are very powerful for natural language processing and found accuracy scores of up to 0.99. They concluded that with a small corpus of argumentative types of sentences, one can train a sentence classification model and use the argument component predictions to generate argumentation features for classifying news and opinion. In our research we will be showing some of the argumentative structures that were found by our classifiers. Especially in the opinion features these structures can become very clear. We have also found that the SVM classifier yields the best results so we will be doing most of our analysis on the results of that classifier.

Lastly we want to discuss the use of BERT in this thesis. BERT stands for Bidirectional Encoder Representations from Transformers. It is said to be the state of the art on many natural language processing tasks (de Vries et al., 2019). It was the same model the research from T. Alhindi et al. used. In this research we will use a Dutch pretrained BERT model called BERTje, which was designed by de Vries et al. In the research of Alhindi and colleagues, BERT performed exceptionally good so this technique is very promising. When working with BERT there are two options. You can train your own BERT model on your own corpus and fine tune it to fit it to your own classification task (de Vries et al., 2019). You can then chose to either use BERT as a classification model or as an embedding. For this thesis we chose to use the Dutch pre-trained model called BERTje and use it as a word embedding. We will not be creating our own model because it is very hardware intensive and does not fit our goals of comparing the same classifiers. BERT is a very promising piece of technology for natural language classification, however it did come with some problems. It requires a lot of computational power and is a very large topic to discuss in a bachelor thesis. That is why our experiments regarding BERTje will remain brief and meant mostly as a first step in exploring the possibilities of this technology.

The research that was mentioned mostly focused on getting a high accuracy score. This paper further analyses the results for characteristics that can help us understand how opinion differs from news. We will also be working with Dutch articles rather than English ones.

2.4 Our research goals

This research does not focus on lexical properties such as sentence length or punctuation but instead only considers the words in the document as features. It explores different techniques and compares the results. We chose this rather than working with lexical properties to make our approach as broad as possible. We are hoping that with this broad approach documents that do not perfectly abide to those features cam still be classified by the model. This research is mostly meant to be explorative and that is why the parameters of our experiments do not necessarily match those of a real world situation. As mentioned, one of our main goals is to find patterns in the results of the classifiers and learn what kind of structures a classifier can find in the text. These results, combined

with the knowledge of what other techniques perform well in news and opinion classification, can be useful to create a good functioning real world classifier. We will further speculate what the uses of such a classifier could be in the "future implications" section at the end of this paper.

2.5 Contributions

The code used for these experiments was entirely written by the author of this paper with the exception of two functions. This code was written by V. Kumar on stackoveflow.com and by N. Panwar on Medium.com. Detailed information about license and the links to their websites will be in the "contributions" subsection of the appendix.

3 Experiments and results

In this section we will present our methods we used and our findings. In the experimental setup we will present how we performed our experiments and why we made certain choices. In the results section we will present a structured overview of our findings.

3.1 Research question and objective

To reiterate, the questions we will be trying to answer are: to what extent can we classify Dutch text documents as news or opinion using machine learning? And: can we find patterns in the results with which we can explain the behaviour of the classifier? This means that we want to find out what methods can achieve good accuracy scores and we want to present an analysis of what the key elements of news and opinion articles are.

3.2 Experimental set up

For the data set we are limiting ourselves to articles from a major mainstream newspaper which were provided by the ACED institution for design, art and journalism. This data set is already categorized, but is limited by the amount of opinion articles. We have access to hundreds of thousand of news articles but the opinion articles are just over 18.000 entries. This means the largest training set we can train on will have a total of 36.000 entries if we want to use a balanced training set. This will be further reduced after cleaning the text and splitting the set in test and training sets.

A balanced set is a good place to start but it is not necessarily representative for the distribution of news and opinion articles in the average newspaper. In our database 5,43% of the total news and opinion articles are opinion articles and 94,57% are news articles. This is not necessarily a perfect example of how articles are divided in all newspapers because different labelling practices can be used by different newspapers. It does however tell us that there is a majority of news articles compared to opinion articles when reading this newspaper. We chose to not have the same distribution in of news and opinion articles in our training set as the database because that could cause the model to always predict news and still achieve a high accuracy score. Then we would have very low F1 scores on for the opinion labels and we would not be able to learn anything about

our training set. One of our goals is to explain the behaviour of the models and relate them to journalistic properties. We can not do this if the model just predicts news all of the time. If the training set has balanced labels it is more likely that we can find the patterns in the data that we are looking for.

Choosing to train in a balanced data set does mean that the model does not immediately translate to a real world application. Before the model could be used in such an application additional testing with unbalanced distributions of opinion and news articles should be done. Alternatively a balanced training set could be used and a cutoff point could be defined for when an article gets classified as news or opinion. For example, we can say an articles has to be at least 95% likely to be news before it gets classified as news. This way we can work with a balanced training set and still incorporate this distribution in our tests. Because making a practical application of the model falls outside the scope of this thesis we will not be doing such experiments.

The order of the data set will be randomised every time the program is run and randomly split in a training set which contains 80% of the entries and a test set that contains 20% of the entries. However, when using cross-fold validation methods the split will be dependant on the amount of folds. We chose this randomisation to keep the model balanced. If we would just draw the first N news articles from the database we might end up with articles only published in the same year or written by the same writer. By randomising the data set we hope we can prevent overfitting to a certain extent. An example set of the distribution of the years in a training set can be found in the appendix in table 19.

We want to create a model that works well over a long time period. That is why we will be using an additional test set when testing our model. This test set will contain only articles published later than the articles that were in the training set. This way we can simulate how techniques perform on articles that would be written years after the model was created.

The program will have the following setup. Parameters can be given to the program that decide the training set sizes, what algorithms should be used, and what other features should or should not be used. For each given training set size all classifiers will be trained. The training and test set will be reduced in size after and shuffled each iteration. This allows us to first train on the largest data set available and still train on smaller data sets to compare the results.

3.2.1 Exploration

To get a feel for the data set we will discuss the dimensions of the data set in this section. When training on articles, each unique occurrence of a word counts as a feature and will increase the dimensionality. In this section we visualised the dimensions from a randomly drawn sample and made a word cloud of the data sets. The sample set size is based on the largest training set we can draw from the database while keeping the labels balanced, currently 28,460 articles total. This amount is after splitting on test and training sets. We see in the word clouds that some of the words occur more than once. This happened because while generating the word clouds ‘collocations‘ where set to true, meaning the software tried to make word groups in the clouds that occurred in the text together.

Figure 1a represents all words in a randomly drawn set with the label ‘news‘, figure 1b represents all words in the same training set labelled ‘opinion‘. The word clouds are generated by joining all words in a random training set in a string. Than the ‘wordcloud‘ and ‘generate‘ functions

(a) News (b) Opinion

Figure 1: Word clouds, news and opinion

(a) News (b) Opinion

Figure 2: Word clouds with stopwords removed

produce a word cloud. Besides some image size settings, all settings are set to default. It is clear that most words are 'De', 'het' and 'van'. This is to be expected because much of our language exists out of these words. The other words are mostly adverbs. This is also to be expected because regardless of what an article is about, most of them will need adverbs.

Things get more interesting when we look at figure 2a and 2b. These represent all words in a random training set after a standard list of Dutch stopwords has been removed. The Dutch stopwords list was created by Xia (2016). This leaves us with a more differentiated data set which can already give us a slight idea of the difference between the contents of news and opinion articles. Many news words correspond with opinion words, such as 'Nederland' or 'gaat' which are both very common in Dutch articles. Though this visualisation can give us an idea of the contents of the data set we are training on, it is by no means a good representation of the models that will be produced by most classifiers. This is because the word clouds simply count the occurrence of each words and all words can occur in both the news and opinion set. They do not keep track of co-occurrence in any way or gives any to the difference between news and opinion texts. In the results and discussion section we will discuss the words that were classified as most likely to be in a news article or opinion article. Figure 3a represents the amount of articles, both news and opinion, and their total amount of words. The last column represents all articles that have more than the largest specified amount. These are grouped together to keep the graphs from becoming unreasonably long. Most articles are grouped around the 80-800 word range, only a small portion of articles have more than 900 words. Figure 3b represents the unique words, with most articles having between 30 and 400 unique words. Note that both these graphs were based on one randomly drawn set after splitting on test and train data. This data set is the largest we could get while keeping the labels balanced, currently 28,460 articles total and is made up of articles from 2008 to 2018. The graphs were generated after all cleaning and removal of stopwords was done.

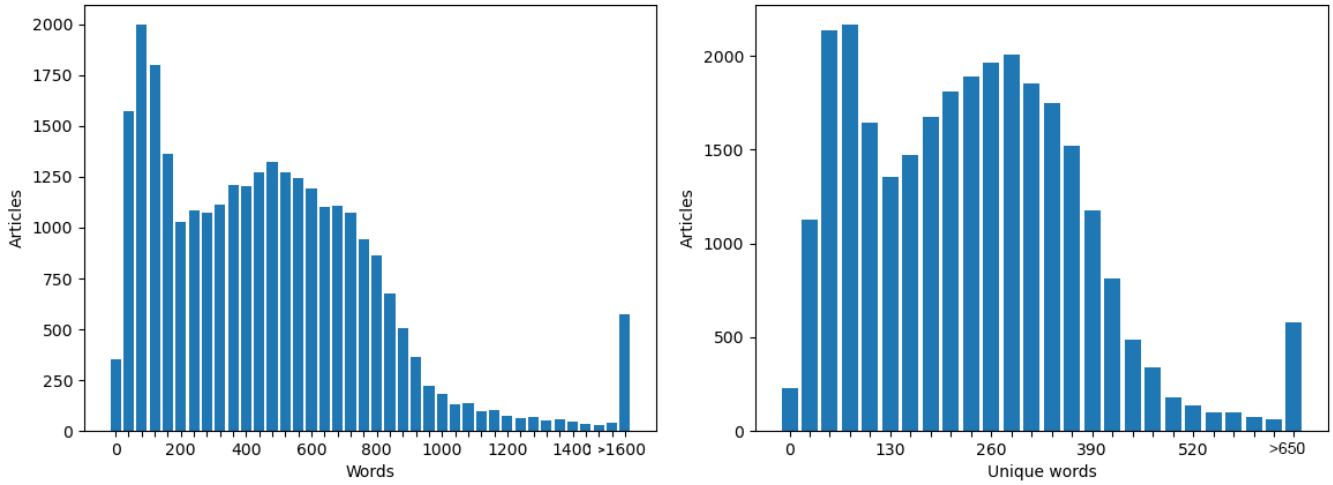
We also see two new shapes in the data when we create separate graphs for news and opinion. We see a Gaussian distribution for opinion articles and a non-Gaussian distribution for news. These graphs can be found in, figures 4a, 4b, 5a and 5b. This tells us that if in the future we would want to create a model with the goal to perform as good as possible in a real world scenario, document length might be a very interesting feature to look at for this specific news source.

Most importantly when we analysed the sizes of the training set, we saw during our tests between 200,000 and 300,000 unique words. These amounts are no hard limits due to the random samples of articles that are drawn each time the program is ran. This amount might seem very large but keep in mind that each different spellings, conjugation, names, or any string with a unique order of characters will be counted as an unique word. For example, "Verkiezing", "Verkiezingen" and "Verkiezingsbeleid" are all considered different words. The total amount of words in the training sets have reached up to 1,500,000 words. This is not a strict upper limit either, but an observation of some of the tests we have run. Of course this amount is dependent on the length of the articles that randomly get put in the training set.

The further implications of these sizes are especially important when we use a BERT model which we will introduce in the next section.

3.2.2 Data representation and pre-processing

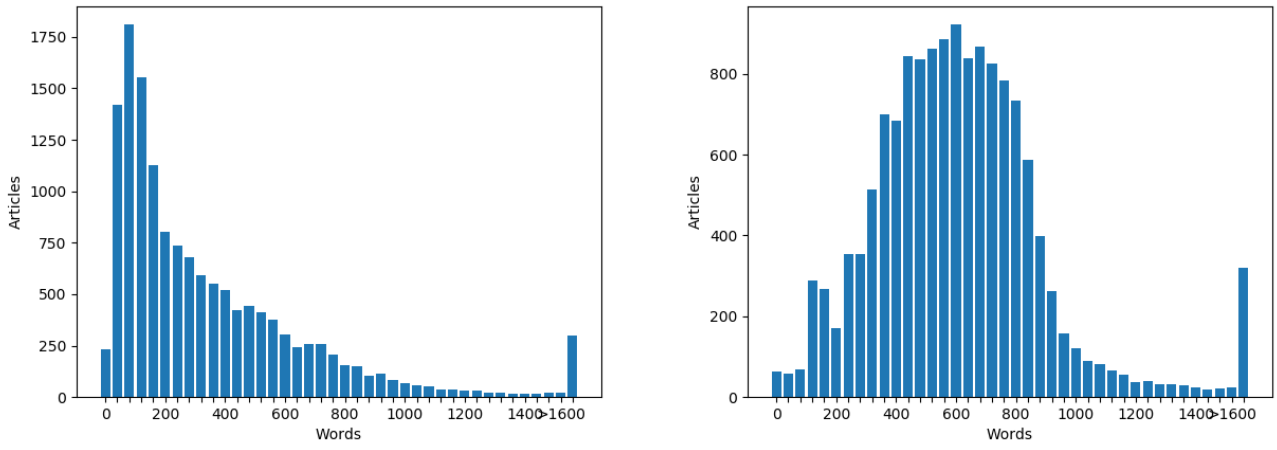
In this section we will present two different type of methods we use to represent words for training. These are tf-idf and BERT.



(a) All words

(b) Unique words

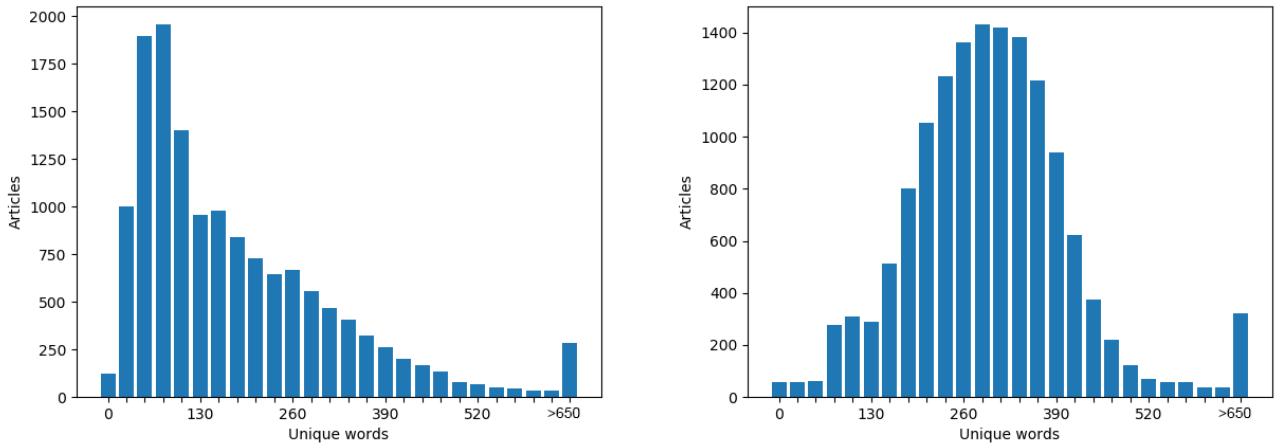
Figure 3: Representation of amount of words in a training set



(a) All words in news articles

(b) All words in opinion articles

Figure 4: Representation of total words in a news and opinion training set



(a) Unique words in news articles

(b) Unique words in opinion articles

Figure 5: Representation of the unique words in a news and opinion training set

First we will discuss tf-idf. Tf-idf, or term frequency–inverse document frequency is a numerical statistical method that is used to reflect the importance of a word to a document in a corpus. It represents the word frequency in a document, multiplied by a weight dependant on the total occurrence of that word in the corpus (Aizawa, 2003). Doing so will make words that generally appear in all documents, such as the word "de", less important. Tf-idf is a variation of bag of words, bag of words simply counts the words without any weight factor. We chose to use tf-idf for most of the testing because putting weights on the bag of words performed better than training without those weights, as shown in the results in figure 8a, 8b and table 3. In the table and figures we see that the accuracy scores of tf-idf are higher. Meanwhile it is still able to produce a model very quickly. Tf-idf is calculated as following:
t = A term, or in our document, a word
Nt = Number of times word t appears in a document.
Tt = Total number of words in the document.
D = Total number of documents.
Dt = Number of documents with word t in it.

$$tf - idf = TF * IDF$$
$$TF(t) = (Nt)/(Tt)$$
$$IDF(t) = log(D/Dt)$$

Second, we discuss BERT, which is the latest state of the art word encoder used for natural language processing, created by Google (Turc et al., 2019). We will use a Dutch variant pre-trained model called BERTje, or more specifically, bert-base-cased-dutch (de Vries et al., 2019). Details on how this model was fine-tuned can be found in the paper of de Vries et al. (2019). BERTje is used to map words in a vector space where the distance between vectors represents how often words are used together. There are two important things to keep in mind when training with BERTje. First, we used a maximum token length of 512 characters for each article. We chose 512 because it was

the default amount and our hardware was not powerful enough to run experiments with bigger token lengths. This is much smaller than the length of the majority of articles, as seen in figure 3a. Although we set this maximum length to 512 characters, we use even shorter sentences because BERT requires too much RAM if the sentences get too long or if we add to many articles to our training set. This means that we need to cut a significant piece of information from each article before training.

The experiments with BERTje happen as follows: we use BERTje to represent all articles in the train and test set in a BERTje embedding. This means we use the pre-trained model bert-base-cased-dutch to embed those articles. Then we proceed as normal, training our classifiers with the resulting training set and testing the models on the resulting test set. This means that in all tables with results using BERTje, the classifier that was trained will also be mentioned.

Overfitting is always a risk when training classifiers. Therefore we propose a few methods to reduce it as much as possible. This is important because we work with very large feature spaces where overfitting occurs often. Underfitting is currently less of a problem because it is usually caused by not having enough features (Smart, 2016).

During this thesis we will use the terms signals and noise. Signals are patterns in data that we want our model to learn. Noise represents irrelevant data and randomness that we do not want to include in our model (Smart, 2016). We added a few standard methods to keep the training sets clean. We made the text lower case, removed all punctuation and removed all Unicode, which contains all special characters. Cleaning the text this way removes a lot of information, such as the beginning and end of a sentence by removing all capital letters. For us this is not a problem because we do not plan on these kind, of features, as we mentioned in the related works, we will only train on the text as a whole. If we would still leave words with capital letters in the model could start making differences between “the” and “The”, increasing the feature space and therefore risking overfitting. Lastly we also removed most HTML tags and JavaScript code that were not meant to be in the text in the first place. We hope that in this way we have removed some of the noise from the model.

3.2.3 Stopword removal

Not all words left after the cleaning will be used when we train the model. We will remove some additional words if they meet certain criteria. First most we will try to remove as many names of authors or publishers as possible. This is not an easy task because our database does not contain a structured oversight of these names. We used a regular expression that found some of the names if there was a date and time mentioned in the first few words. This was always followed by a name so we removed those words.

Additionally we have a self defined list of words to remove. This list mostly contains html tags for cleaning the text. This list also contains the words “news” and “opinion” to prevent the model to train on the label directly. We also added the name of the newspaper which intends to make the model more general and fit better on articles from other news sources. We also removed the years 2000 to 2020 with the intention to achieve better results when testing an older model we created on newer data. In this thesis we will refer to the removal of these terms as ‘cleaning‘.

Lastly we added the option to remove Dutch stop-words from the data set. In this thesis when we refer to ‘removing stopwords‘ we mean only removing stopwords in the Dutch nltk stopwords

Parameter	Functionality
No-Stops	If set to True, the Dutch nltk stopwords will not be removed
Do-Bert	Disables tf-idf representation and enables BERTje embedding
Train flag	Specifies what classifiers to use.
Maximum data size	Specifies the amount of entries in the training set
Folds	The amount of folds used for N-fold cross validation
Validate-year	Creates a separate validation set of only instance from the specified year

Table 1: Parameters.

list. 'Cleaning' and 'removing stopwords' are done separately. These stopwords are pre-defined words that sometimes do not offer additional information when training on text. Removing these words can improve the performance of the software that trains the classifiers, without hurting the accuracy scores. This depends per project and we will be testing if this is the case for our research. When we are training with the BERTje embedding, removing the dutch nltk stopwords is less relevant because BERTje will make its own mapping of what words are relevant. The only downside of not removing stopwords when using BERTje is that the stopwords could take up space in the maximum amount of tokens when that space could also have been occupied by words that carry more relevance for news and opinion classification. We have tested whether this statement is true and will analyse the results of those tests in the discussion.

3.2.4 Evaluations

We will have several types of results. First of all, accuracy scores for each data set size and each algorithm which tells us how good a given classifier performs. We will also include a table of the F1 scores, recall and precision of the largest data sets, because they tend to give the best results. Furthermore, we will include the top N opinion features and the top N news features. A word cloud of the used data set and lastly the articles with the probability scores of whether they are opinion or news. Our validation techniques include cross-validation and a validation test on dates were not included in the training set.

The top N features are useful to give us insight on what the model actually did to reach its results. We can combine them with our probability scores and trace back how a sentence is actually build up. We expect to see many top performing features in articles that have a high probability score.

3.2.5 The parameters

In the following table you can find the most important parameters for the tests. There are a few other parameters which are mostly used for program diagnostics. More information on these parameters can be found in the readme.

Some attributes of the parameters must be mentioned before interpreting the results. No-Stops will have no effect on the removal of words in the self defined removal list. We made this decision because the presence of some words can have an adverse effect on the trained model and should always be removed, as mentioned in the "Stopword removal" section. If do-bert is set to False,

No Stops	True
Do Bert	False
Train flag	LogReg, SVM, linSVM, XGB, NB, RF
Max articles	6000, 14000, 22000, 28000, 36000
Folds	4
Validate year	2019

Table 2: Parameters used during runtime.

tf-idf embedding will be used instead. The available classifiers are logistic regression (LogReg), linear Support Vector Machine (linSVM), Naive Bayes (NB), random forest (RF), XG Boost (XGB) and Support Vector Machine with a radial kernel (SVM). Maximum data size can have multiple values. If multiple values are given then a model for each given size will be created and plotted for comparison. We use N-fold cross-validation. Setting "folds" to 0 will skip the cross-validation. The program will automatically calculate the test-train split for folds larger than 1. If the amount of folds is set to 1 or 0, the test train split will be 0.2. If validate year is specified no entries from that year and on wards will be used to train the model. The separate validation set will then be produced using only entries from the unused dates. The folds are never applied to the year validation set because 100% of the training set is already used to train and 100% of the validation set is already used to test. An important note is that we usually test both on N-fold cross-validation and a year-validation at the same time so we can run both experiments at the same time. We split the database before we train our model. This means that the model that gets validated with the year validation set is 1\Nth smaller than its maximum size, because that portion is already reserved as the first test fold.

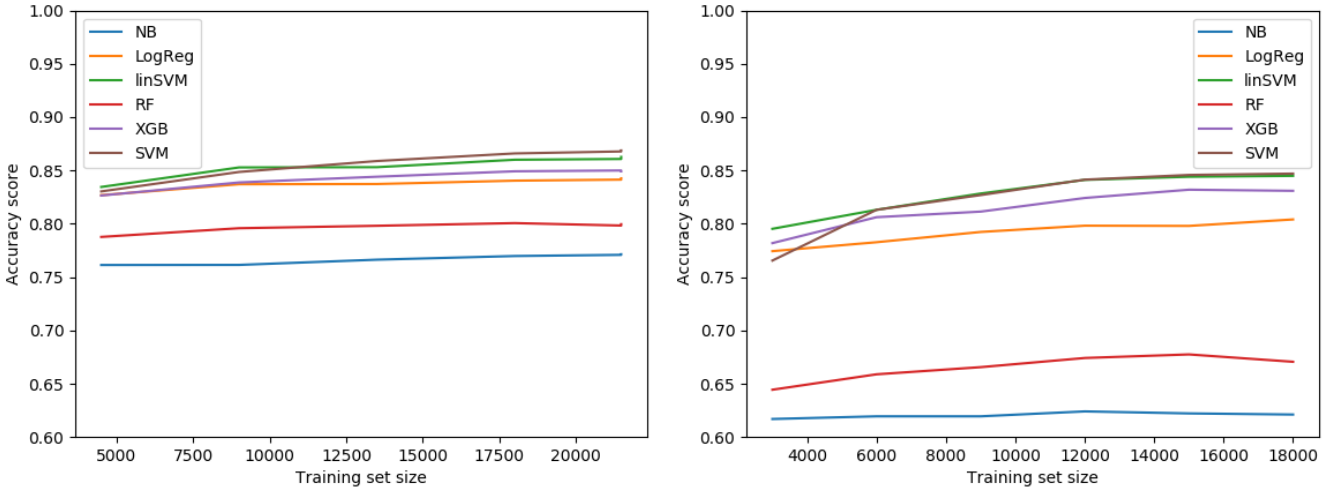
3.3 Results

In the results, when we talk about default settings we mean the settings as shown in table 2. We chose these settings as default settings because they are the simplest and can function as a good baseline for comparison. The self defined word removal list is the same for every train instance, it is however possible to add more words to this list if we expect those words to interfere with correct training results.

3.3.1 Results with default settings

In figure 6a we see the accuracy scores from different classifiers with different training sizes. Overall the classifiers score good but the best performing are SVM and linear SVM, as shown in the graph.

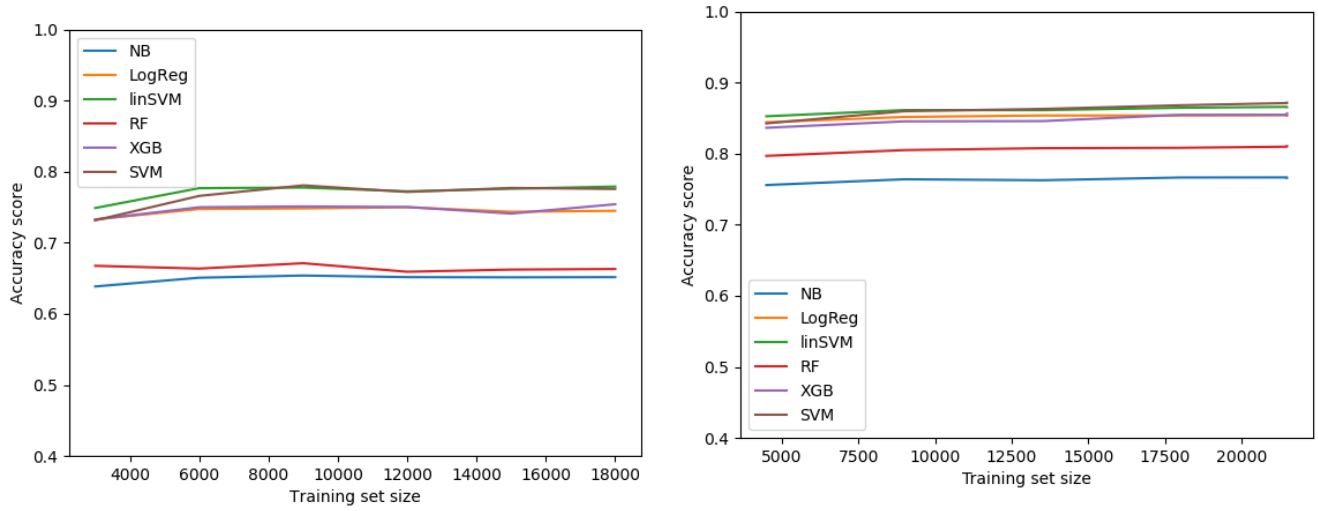
In figure 6b the performance of the same algorithms when tested on a test set drawn from a year on which the classifier did not train is shown. If we examine the additional test set we can see that on the short term (1 year) the model holds with 0.85($\pm$0.01) percent, losing only about 0.02 points accuracy. However, as time moves on the model starts to degrade, being about 0.09 percent less accurate on a model tested on data from 2016 - 2019 and trained on data before 2015, shown in figure 7a.



(a) Accuracy scores (default settings)

(b) Accuracy scores (train 2008-2018, validation 2019)

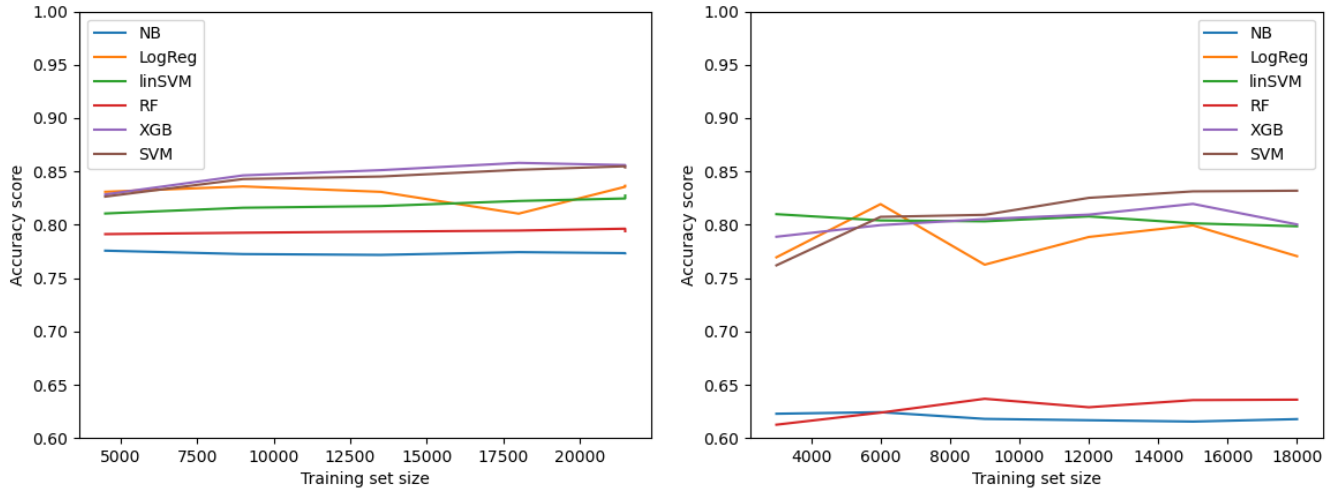
Figure 6: Accuracy scores for default settings and using tf-idf



(a) Accuracy scores (train 2008-2015, test 2016 - 2019)

(b) Accuracy scores after removing Dutch nltk stopwords (train 2008 - 2019)

Figure 7: Accuracy scores using tf-idf



(a) Classifiers trained with bag of words representation (b) Classifiers with bow representation, tested on 2019

Figure 8: Accuracy scores using bag of words

Settings	Label	Accuracy	Precision	Recall	F1	Train size
SVM bow	Nieuws	0.85 ±0.00	0.87 ±0.00	0.83 ±0.01	0.85 ±0.00	21,500
	Opinie	0.85 ±0.00	0.83 ±0.01	0.88 ±0.00	0.85 ±0.00	21,500
SVM 2019 validation bow	Nieuws	0.83	0.84	0.82	0.83	18,000
	Opinie	0.83	0.82	0.85	0.83	18000
SVM tf-idf	Nieuws	0.87 ±0.00	0.90 ±0.01	0.83 ±0.00	0.86 ±0.00	21,500
	Opinie	0.87 ±0.00	0.84 ±0.00	0.90 ±0.00	0.87 ±0.00	21,500
SVM 2019 validation tf-idf	Nieuws	0.85	0.87	0.82	0.84	18,000
	Opinie	0.85	0.83	0.88	0.85	18,000

Table 3: Bag of words compared to TF-IDF, with default settings.

Additionally, we see in figure 7b the accuracy scores of the training results after the Dutch nltk stopwords have been removed. This graph looks very similar to the graph with the results from the defaults settings. We will address the possible explanations of this in the discussion. Lastly in figure 8a and 8b the accuracy scores of of the classifiers trained on a bag of words representation are shown. An oversight of the results of the classifiers trained on a bag of words representation are shown in table 3, together with the results of the same classifier trained on a tf-idf representation for comparison. An oversight of all results can be found in table 4. Note that the last column "train size" refers to the total amount of articles used for training the model. This includes both news and opinion articles.

3.3.2 BERTje embedding

When running the experiments we ran into a few issues. The biggest one being that creating the BERTje embedding required a lot of RAM which often bottle-necked the process. To solve this issue we had two options: to reduce the size of the training set or reduce the size of the maximum length of each article. In the end we tested both for comparison. We also decided to test the influence of removing the Dutch nltk stopwords before creating the Bert model. This resulted in running four experiments, the results of which are in tables 15, 16, 17 and 18. The best performing results are also included in table 4.

The experiments with the reduced document lengths were trained on multiple different train sizes. Those results are also presented below in figure 9a and 9b. Note that even though the maximum length of the articles was reduced to 40 words, the biggest training set we could use was only 1200 articles in size. Increasing the sizes beyond those values was not possible due to hardware limitations.

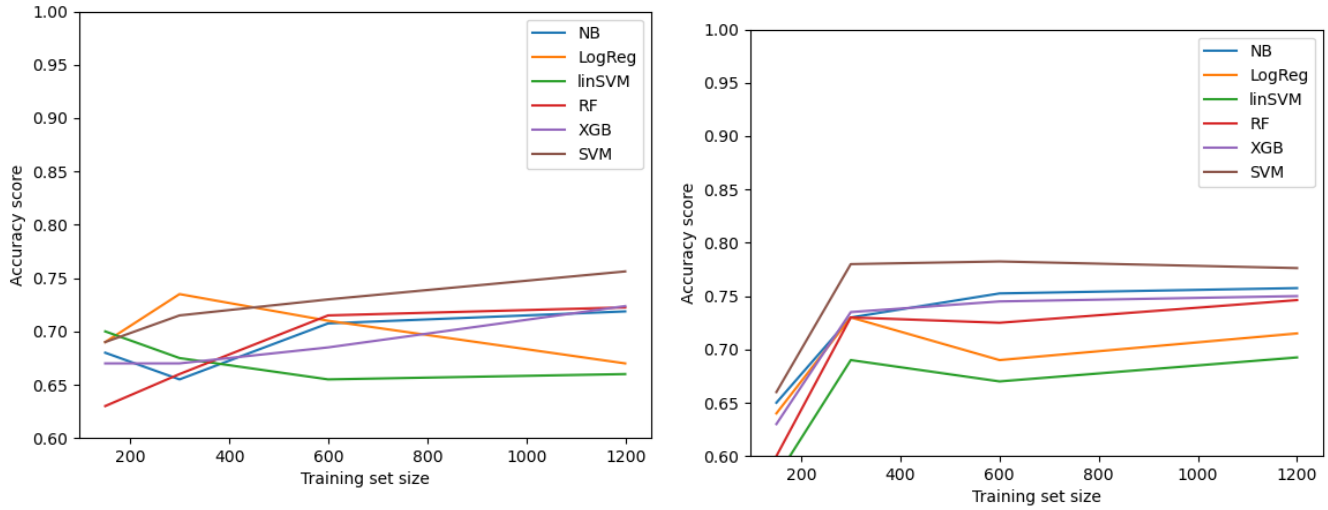
We have also done an experiment with only 300 articles. This allowed us to increase the maximum document length to 256 total words each. Because there seemed no point in training this model on less than 256 articles, there is no graph showing the results on different training set sizes.

On a general note, because we have been training with relatively small training set sizes we decided to double the relative size of the test sets, compared to the other tests we have run. This is in an attempt to make the results more accurate. Also, all standard deviations that are mentioned in all results coming from an experiment with BERTje were not made using cross fold validation. Due to a problem with the code, the cross-validation did not function with the BERTje embedding, so we run each experiment four times and calculated the standard deviation from those results. Figure 9a and 9b are not a weighted average, but are drawn from one of those experiments.

3.3.3 Feature importance analysis

In this section we will explain what the models have learned by examining the words that were marked as most likely to be used in either news or opinion articles. We will do this using the results from the linear SVM and the tf-idf features.

In figure 8 we see the top 50 words that were the most influential features for news articles (red) and the top 50 words that were the most influential features for opinion articles (blue). In table 8 and in table 5, 10 articles are presented that are classified as most likely to be news or most likely to be opinion, displayed by their title. Tables 6, 7, 9 and 10 contain the top 8 news and opinion



(a) Classifiers trained with BERTje embedding and without stop-words removal

(b) Classifiers with bow representation, tested on 2019

Figure 9: Accuracy scores

Settings	Label	Accuracy	Precision	Recall	F1	Train size
SVM 4-folds	Nieuws	0.87 ±0.00	0.90 ±0.01	0.83 ±0.00	0.86 ±0.00	21,500
	Opinie	0.87 ±0.00	0.84 ±0.00	0.90 ±0.00	0.87 ±0.00	21,500
SVM 2019 validation	Nieuws	0.85	0.87	0.82	0.84	21,500
	Opinie	0.85	0.83	0.88	0.85	21,500
test on 2016-2019**	News	0.78	0.81	0.72	0.76	18,000
	Opinion	0.78	0.75	0.83	0.79	18,000
Stopwords 4-folds	News	0.87 ±0.00	0.90 ±0.1	0.84 ±0.1	0.87 ±0.1	21,500
	Opinion	0.87 ±0.00	0.85 ±0.2	0.91 ±0.1	0.88 ±0.1	21,500
BERTje + SVM ***	news	0.85 ±0.02	0.91 ±0.01	0.79±0.05	0.85±0.03	300
	opinion	0.85 ±0.02	0.80 ±0.02	0.92±0.01	0.86±0.01	300

Table 4: Results of SVM classifier on largest possible data set.

* These are trained on the same set of data. When creating the model all articles from 2019 were left out and used for a test set. Other experiments are based on randomised data sets.
** The model was created using data only before 2016. The test set was created only using data of 2016 to 2019.
*** The model was trained with articles with no more than 256 words and with no stop-words removal

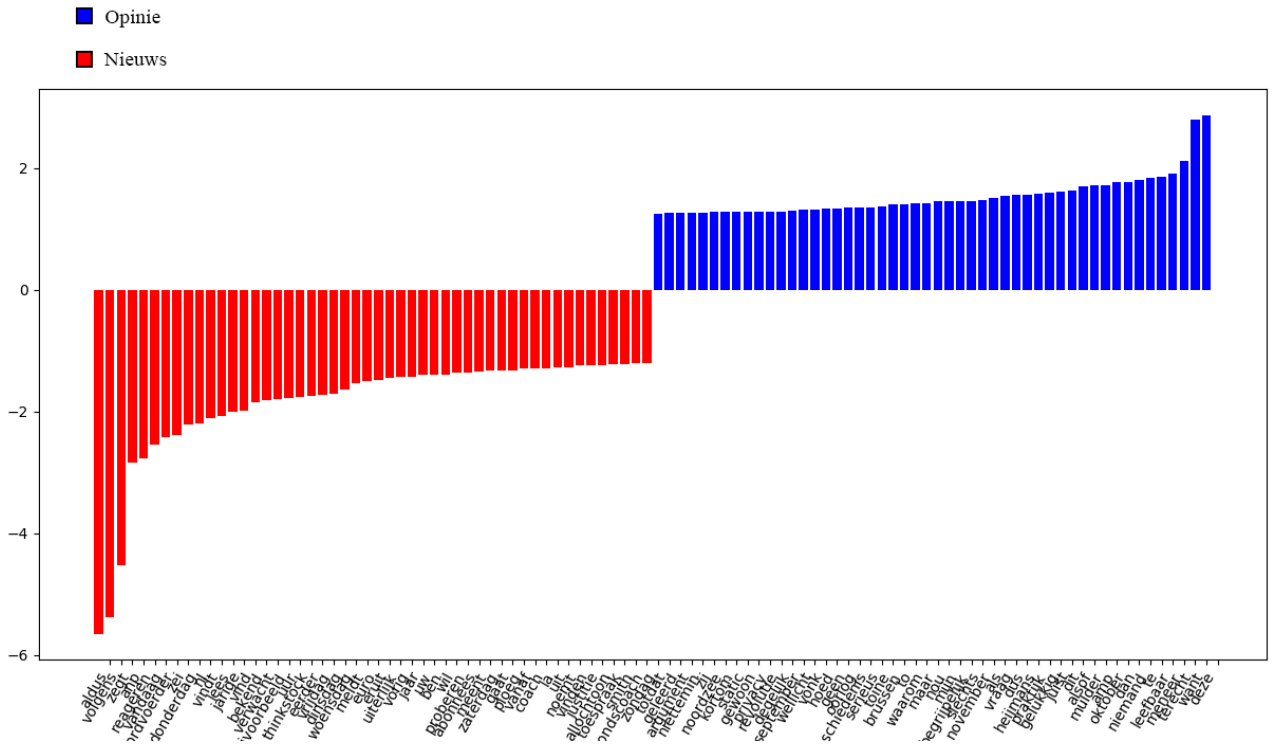


Figure 10: Top 100 feature importance (linear SVM)

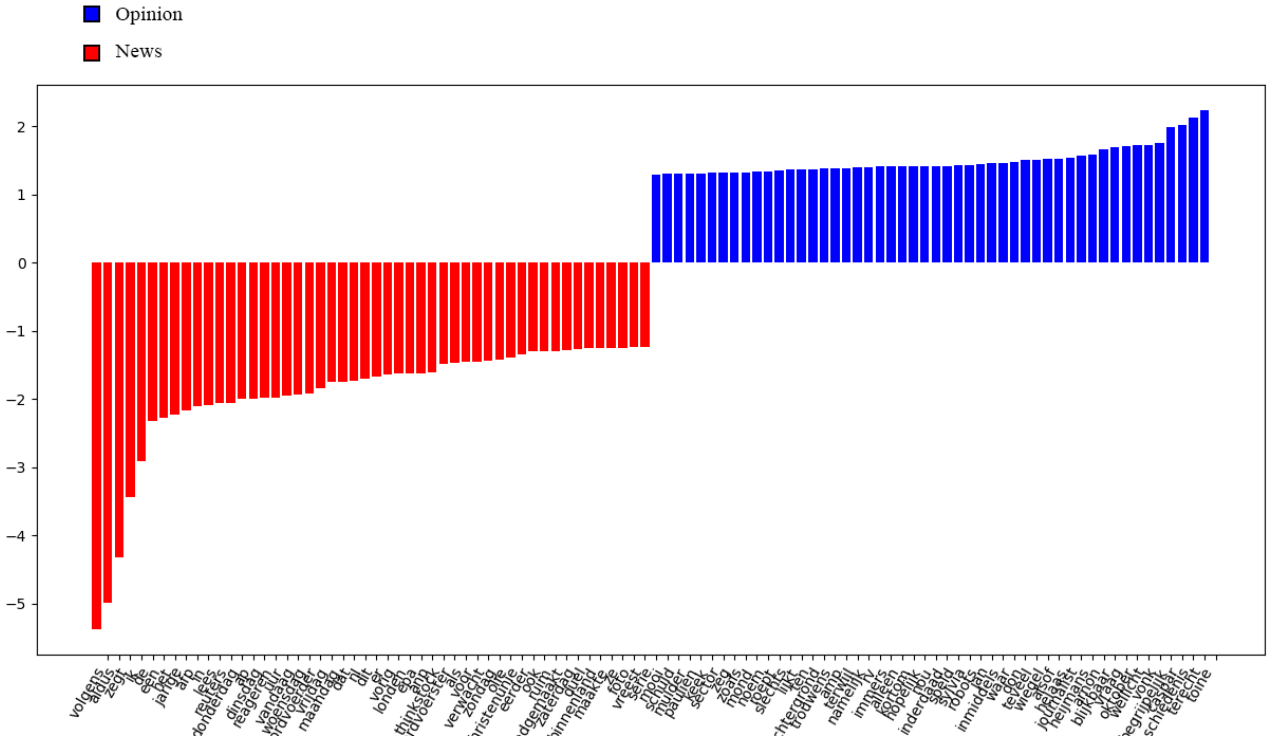


Figure 11: Top 100 feature importance with a bag of words representation (linear SVM)

Article number	Article title
1	120 Syrische agenten gedood in hinderlaag
2	Achenteh past uitstekend in onze offensieve voetbalvisie
3	Britse gevangenen eten beter dan patiˋenten
4	Duits leger overspannen
5	Japanse minister heeft zelfmoord gepleegd
6	Kabinet wil hulp chronisch zieken inkomensafhankelijk maken
7	Ministerie komt afspraken met uitgezette hongerstakers niet na
8	OVSE-waarnemers Oekraïne vrijgelaten
9	Twaalf verdachten betrokkenheid aanslag Parijs ondervraagd
10	Van der Sloot zit straf gewoon uit in Peru

Table 5: top 10 'most likely to be news' articles

-	aldus	volgens	zegt	anp	reageren	vandaag	woordvoerder	zei	sum
1	0.0	4.39	0.0	0.0	0.0	0.0	0.0	2.63	7.02
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	3.14	2.19	0.0	0.0	0.0	0.0	0.0	0.0	5.33
4	0.0	4.39	0.0	0.0	0.0	0.0	0.0	0.0	4.39
5	3.14	2.19	0.0	0.0	0.0	0.0	0.0	0.0	5.33
6	0.0	0.0	0.0	3.97	0.0	0.0	0.0	0.0	3.97
7	6.28	4.39	2.58	0.0	0.0	0.0	4.18	0.0	17.4
8	3.14	8.79	2.58	3.97	0.0	0.0	0.0	0.0	18.4
9	3.14	0.0	0.0	0.0	0.0	0.0	0.0	2.63	5.77
10	0.0	0.0	0.0	0.0	0.0	3.27	0.0	2.63	5.9
avg	1.9	2.63	0.52	0.79	0.0	0.33	0.42	0.79	7.35

Table 6: News articles with top 10 news features

features in those sentences. The numbers in the first column in tables 6, 7, 9 and 10 refer to the article numbers and titles in table 8 and 5. The cells in tables 6, 7, 9 and 10 are the occurrence of the feature in that column, in the article which is represented by the article number. The 'sum' column represents all values of a row summed up. The 'avg' row represents the average of all values in all cells above.

4 Discussion

In this section we will discuss and interpret the results shown above. We will also consider possible mistakes and speculate how to improve on this research and on how it could be used in practice.

-	deze	want	terecht	meneer	leefbaar	te	niemand	dan	sum
1	1.74	0.0	0.0	0.0	0.0	0.0	0.0	1.42	3.16
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.85
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	5.65	0.0	1.42	7.07
8	0.0	0.0	0.0	0.0	0.0	2.26	0.0	0.0	2.26
9	0.0	0.0	0.0	0.0	0.0	3.39	0.0	0.0	3.39
10	0.0	0.0	0.0	0.0	0.0	3.39	0.0	0.0	3.39
avg	0.17	0.0	0.0	0.0	0.0	1.58	0.0	0.57	2.32

Table 7: News articles with top 10 opinion features

Article number	Article title
1	'Gezond populisme' bij links én rechts is hoopvol
2	'Foute' stemmen in het publieke debat willen smoren, werkt averechts
3	Redelijk rechts vernietigt zichzelf
4	Wilders reduceert het woord tot frase
5	Wie welvaart 'wegbelast', smoort groei in de kiem
6	Wie moet hier de telefoon beantwoorden?
7	Wat hier op het spel staat, is mijn leven
8	Waarom ik begrip heb voor de Friese blokkeerders
9	Waarom grijpt de politiek niet in waar nodig?
10	Vragen over wetenschap en islam

Table 8: top 10 'most likely to be opinion' articles

-	deze	want	terecht	meneer	leefbaar	te	niemand	dan	sum
1	0.0	0.0	0.0	0.0	0.0	16.9	0.0	0.0	16.9
2	1.74	9.51	6.38	0.0	6.44	30.5	3.18	15.7	74.0
3	5.22	0.0	0.0	0.0	0.0	20.3	0.0	7.14	32.6
4	3.48	9.51	0.0	0.0	0.0	12.4	0.0	2.85	28.2
5	6.96	0.0	0.0	0.0	0.0	7.92	0.0	4.28	19.1
6	6.96	4.75	3.46	0.0	0.0	13.5	6.37	4.28	39.3
7	1.74	2.37	0.0	0.0	0.0	12.4	0.0	9.99	26.5
8	0.0	2.37	0.0	0.0	0.0	12.4	3.18	2.85	20.8
9	6.96	0.0	0.0	0.0	0.0	2.26	6.37	0.0	15.5
10	8.7	4.75	0.0	0.0	0.0	24.8	0.0	8.56	46.8
avg	4.18	3.33	1.04	0.0	0.64	15.34	1.91	5.57	29.27

Table 9: Opinion articles with top 10 opinion features

-	aldus	volgens	zegt	anp	reageren	vandaag	woordvoerder	zei	sum
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	2.58	0.0	0.0	0.0	0.0	0.0	2.58
3	0.0	2.19	2.58	0.0	0.0	0.0	4.18	0.0	8.95
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	4.35	0.0	0.0	0.0	4.35
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	3.14	0.0	0.0	0.0	0.0	6.54	0.0	5.26	14.9
avg	0.31	0.22	0.52	0.0	0.44	0.65	0.42	0.53	3.08

Table 10: Opinion articles with top 10 news features

4.1 Experiment with default settings

When we look at figure 6a and table 4 we see that the highest accuracy score we can achieve is 0.87 ±0.00 with a size of 22,500 articles, with balanced labels. We can see a minor increase in the accuracy scores when we increase the size of the training set. We can also see those scores plateau and that training on smaller training sets yield results that are close to the largest training set. All other results of each other data set can be found in the appendix, in table 11, 12, 13 and 14.

When examining the different classifiers, we see that random forest and Naive Bayes perform particularly worse than the others. For random forest this is no surprise because it uses decision trees which do not perform well on natural language processing (Breiman et al., 2011). Naive Bayes was a bit more of a surprise at first because the papers mentioned in the related works mostly worked with Naive Bayes. Naive Bayes performs the best when using features such as document length and punctuation. Naive Bayes can give those features probability scores, which is a more accurate way of classifying than just giving a probability score to each word (Rish et al., 2001). This is because most words that are likely to be opinion could just as well be used in news articles and the other way around. This may be the reason it performs relatively bad on our data set.

That leaves us with the best performing classifiers. These are logistic regression, XG Boost, SVM and Linear SVM. When we examined logistic regression, we found that it performs slightly worse than SVM but significantly better than Naive Bayes and random forest. Logistic regression also seems to be a bit more inconsistent when testing out of time. Overall, the results of logistic regression seem quite average. Regression has a hard time finding complex relations between features, which is somewhat required for natural language processing (Kleinbaum et al., 2002). Overall, our results are within the expectations of logistic regression. A large upside of logistic regression is that it trains very fast, so it could be used to retrain the model each time new data becomes available.

XGBoost is a new and popular algorithm that produces a model by combining multiple weak prediction models, often decision tress. It is quite slow but usually yields good results. In our project, XGBoost has performed above average in comparison with the other classifiers. We expected XGBoost to perform better than it did, and the most likely reason it did better than perform above average is because it is quite prone to overfitting (Chen & Guestrin, 2016). There is undeniably some form of overfitting in our data and having a cleaner data set might allow XGBoost to work to

its full potential. Because XGBoost trains quite slow so it does not fill a role of quick algorithm like logistic regression can (Chen & Guestrin, 2016).

SVM and linear SVM are the same algorithms with different kernels so we will cover those both in this part. Our expectation of these algorithms was that they would perform well because they handle high dimensional data very well (Schölkopf et al., 2000). Our data is very high dimensional and in the results SVM with a radial kernel performs among the best. This fits our expectation of SVM. We expect that the radial kernel performs slightly better because our data is not very easy to linearly separate. The radial kernel tends to perform the best on the most machine learning tasks so it is no surprise it does here as well. The largest downside of using SVM with a radial kernel is that the time to train the model with radial SVM takes significantly more time than all other classifiers. Due to the complexity of SVM we will not go further in depth on the working of the classifier. We do however see that SVM with the radial kernel outperforms the other classifiers in almost all stages.

4.2 Out of time tests

We performed two out of time tests with this model, the results of which are depicted in figure 6b and 7a. When we tested the model, trained on articles from 2008 to 2018, on articles from 2019. There was about a 2% percent drop in accuracy. This amount starts to increase when we shift the year to 2016. If we train the model on articles from 2008 to 2015 and test on articles from 2016 to 2019, we start to lose an additional 7% percent in accuracy. In these results we see a trend that we start to lose accuracy when we train a model on older articles and test it on newer articles. This fits with our expectations because some commonly used words can change over time. An example of this would be the name of a well known politician, this name would be used in the training of the model. When the model would be tested on a sample which contains articles written when the politician was retired, the model would likely perform worse. Also, we expected the model to perform better on the short term because some noise, such as author names, still exists in the model. We will further explain why we expected this in the known limitations section.

4.3 Stopwords

In this section, we discuss the effects of removing all words in the Dutch nltk stopwords list from the training set and test set before training our model. We chose to remove the stopwords from the test set as well because in a practical setting stopwords can always be removed from a test set before putting it through the model. The results from the experiment with stopwords are very similar to those without stopwords: the difference in accuracy is less than 0.01 percent point. The top 50 features with stopwords also look very similar to those of the model without stopwords. The table with the features of the model trained with stopwords can be found in the ‘results‘ section, figure 11. With our database there does not seem to be a strong indication that the removal of stopwords significantly improves the accuracy scores of the trained models. In this statement we do not mean to include the results produced using BERTje, those results will be discussed in the next section.

4.4 BERTje embedding

BERT has given us some very interesting results. Even though the implementation caused a large part of the text to be removed before being used to train, it still yielded results that were very close to the best classifier configuration without BERTje. Its highest accuracy score of 0.85 ±0.02 was reached with a training set of only 300 articles and only 256 words in each article. These results also maintained a good f1 score of 0.86 ±0.01 for opinion and 0.85 ±0.3 for news. This shows how powerful of a tool BERTje can be with this kind of learning tasks. Having seen these results, it is not surprising that Alhindi and colleagues reached such high accuracy scores when working with BERT(Alhindi et al., 2020).

The tests we run on BERTje with Dutch nltk stopwords removed were somewhat inconclusive. in table 15 and table 16, which represent the experiments with article lengths of 40 words, we see that the accuracy scores seem to be slightly better when we removed the stopwords. However, the standard deviations are also bigger when removing the stopwords, so in reality the accuracy could be the same or even slightly lower. Overall, it did not seem to matter a lot whether we removed the stopwords or not.

During the tests we run on articles with lengths of 256 words, shown in table 17 and 18, we found comparable results. When the stopwords were removed, the standard deviations were a bit larger. We do see that the accuracy scores where the stopwords were not removed performed slightly better, but here we also see that the with the standard deviation in mind it does not seem to matter too much if we remove stopwords or not.

To recap, there does not seem to be a major difference in accuracy scores with or without stopwords. However the standard deviations do seem to get bigger when using stopwords. From these results we can not confirm nor deny that the removal of stopwords has a negative effect on the accuracy scores if the articles are limited to having 256 or 40 words. Future research with bigger test sets and a cross-validation experiment with a sufficiently high amount of folds could give us more conclusive results.

In these results we can see that BERT can perform on small data sets. This was also what we have seen in the research by Alhindi et al. To see whether this is also the case in a real world scenario, more research with a proper out of time and out of source test should be done to confirm if those models created from limited information still hold. Our results have furthermore shown that it is more important to have a few articles with more content than having many articles with less content. Our results are however unclear in whether there is a balance of article length and training set size. Future research in finding such a balance between article length and training set size could optimise the usage of hardware while finding the best results.

Lastly, we want to mention the problems with discussing the patterns that the algorithms learn when we use BERTje as an embedding. Because all features are represented using vectors, it is hard to see which features are important for the model and which are not. It is also hard to visualise the relations between words that BERTje can find due to the high dimensionality in which the vectors are represented. That is why we will not further analyse the inner functionality of BERTje and the patterns it finds.

4.5 Bag of words and tf-idf

As mentioned in the experimental setup, tf-idf functions better on average than bag of words. In table 3 it shows that models built with a tf-idf representation score 2% better than those built on a bag of words representation. The results are not overwhelming but very consistent with a very small margin of error. This lead us to conclude that the usage of tf-idf over bag of words was an appropriate decision.

4.6 Explaining model behaviour

In this section we will analyse the behaviour of the model, using the results presented in section 3.3. We will start with discussing the results from figure 10 and work our way down through the results. In figure 10 the feature weights of 100 words are represented, 50 opinion weights and 50 news weights respectively. If we compare these words with what we initially saw in the word cloud (figure 2a and 2b), there is little to no resemblance. We already discussed the fact that the word clouds were just representations of the amount of words in the training sets without any other form of weights or selection on news or opinion. This means it is no surprise that the features in figure 10 are different kinds of words than those in the word clouds because they are assigned different kinds of weights besides occurrence by the classifier. The Naive Bayes top features are also different from those in the word clouds. We included those features to be able to see the feature differences between SVM and Naive Bayes. However, we will not be using them in the model behaviour analysis, simply because SVM performed much better. The feature words of Naive Bayes can be found in the appendix in the "Top features Naive Bayes" subsection.

In table 8 and in table 5, 10 sentences are presented that are classified as most likely to be news or most likely to be opinion. For most Dutch readers these articles are overwhelmingly clear in being news or opinion articles. Tables 6, 7, 9 and 10 show how the feature words appear in the articles. The rows represent those articles and the columns represent the top 8 most predicting words for news and opinions. We limited ourselves to only those 20 articles and 16 words to prevent the tables from using too much space. To calculate the value for each cell we took the term frequency of a given word in that article (TF) and multiplied it by the inverse document frequency (IDF). We used the tf-idf representation to get a fair representation of each word in those articles and its relevance. In the last column we added the sum of the all the other columns. The goal of this column is to provide a quick overview of the difference in results. The bottom row represents the averages of all values above for the word represented by that column. When examining the sum column we can quickly see that the news features are significantly more common in news articles than in opinion articles. In the same way, we see that opinion features are significantly more common in opinion articles than in news features. These results show us how the generated results fit what we see in the articles and to some extent shows us that the classifiers perform well.

4.7 Comparing model behaviour to journalistic properties

In this subsection we will first examine some journalistic concepts before moving on to the analysis of the model. Harold D. Lasswell created a model for the structure of communication in society. The model proposed that the following questions should be answered when communicating: *Who?, Says what?, In which Channel?, To whom?, With what effect?* (Lasswell, 1948). These

principles are still used in modern news journalism today. An additional question news articles often answer that is not mentioned in Laswells model is the question of "when", which is mentioned in the five W's (Singer, 2008). In the 5 W's Nordquist adds the questions "When" and "how". Opinion articles often have the structure of: "proposition", "argumentation" and "conclusion" (Aldisert, 2009). Having reviewed the criteria for news and opinion we will now examine our results.

Having considered Lasswells model, supplemented by the 5 W's, we can see that the features for news fit the model very well. If we consider the top predictors for news articles: (aldus 1), (volgens 2), (zegt 3), (reageren 5) and (woordvoerder 7), (zei 8) and (vindt 11) we see that all these words relate to a quote someone made. This partly covers the "Who" and "Says what?" questions. Some other high predicting words contain references to time, such as: (vandaag 6), (donderdag 9), (uur 18), (vrijdag 21), (dinsdag 22), (woensdag 23), (zaterdag 46) and (zondag 50). This answers the question of "When". The other questions in Laswells model are not answered in with the top 50 features which could be because these questions are harder to answer in a single word or simply because the patterns are not found by the classifier.

In the opinion articles there emerges a pattern as well, although they are a bit harder to see. The most clear one is a pattern of transition words. These are: (want 2), (terecht 3), (dit 14), (alsof 13), (als 16), (slechts 18), (kortom 46) and (desalnietemin 47). Also a smaller pattern of adverbs can be found in the list: (te 6), (dan 8), (dus 12), (juist 14) and (wellicht 36). There is a clear pattern of names of writers or publishers in the opinion results. Some example of these features are (amp 10), (Mulder 11), (Heijmans 14), (Toinne 27), (vkGeschiedenis 29) and (Vonk 33). These are most likely noise but can not be ruled out as important features just because they are names. We will discuss this further in the "known limitations" section.

The word "dus" refers to a conclusion and the words "want" and "desalnietemin" refer to a argumentation. Other transition words such as words that signal a prerequisite, difference or contradiction could be part of a theorem. This is however not a hard rule because they could be used in other contexts as well. Overall, we do see a pattern emerge of words used for propositions, argumentation and conclusions.

4.8 Known limitations

The foremost issue is the presence of words still left in the training set such as the names of writers and publishers. Although we tried to remove most of it we did not find fitting general methods for each word that should be removed. This will most likely cause the model to perform better on the short term and worse on the long term because sources, though very accurate predictors, can change over time. Also, a model that is trained on a lot of noise will collapse when given articles from other news sources because the names the model trained on will most likely not be found in other news sources. For future work, more general methods should be used to filter these words from the data sets. Lastly, out of time and out of source tests are very important to see if the results could be applicable in real world applications. If the model does not hold on other sources separate models should be produced for each source. If the model does not hold for different time periods those models should be regularly updated. This can however also lead to overfitting the model if the model would contain too many recent sources. Keeping the dates balanced in the data set will be important. The existence of articles from different years and sources can both cause implementation problems if a real world application would be made that performs news and opinion classification.

Training with BERTje has been limited by our hardware to some extent. Even though we used relatively small training set sizes and reduced the content of the articles significantly we quickly reached our limit of 16GB of RAM. We also trained using our CPU instead of using our GPU which can significantly decrease performance. This could be solved by just having better hardware. Another issue we already mentioned in the experimental setup is that the cross-validation functionality did not work correctly when using the BERTje embedding. In an attempt to still validate the results we ran the tests multiple times and took the averages of those results before putting them in the table and wrote down the standard deviation between those tests. We should also mentioned that the used BERTje model from de Vries et al. (2019) was created using cased texts even though our training data was uncased. This will not cause major errors because uncased text can still be embedded by a cased BERTje model, however it would have been a better practice to use an uncased BERTje model for an uncased training set.

For future research we could try to create a fairer split of test and train data. Rather than training on folds that are randomly chosen, we could make splits based on author and test those on splits based on other authors. Using this method could prevent overfitting on names of authors and publishers. Unfortunately reliable author data is often missing in our data set. Additionally, more could be done in terms of completely cleaning the text. Currently, some publisher names still remain in the text. This shows when analysing the opinion features.

We have already mentioned a lot of the future research that could be done with BERTje, but we will recap our remarks. For BERTje, tests with bigger train, test and cross-validation tests should be done. To do so requires better hardware and probably a better cleaning of the database because BERTje does not recognise words that are not in its corpus. Also out of time and out of source tests should be done. Lastly, an experiment should be done to decide which words from an article should be used in the embedding. For our implementation we used the first N words, but this does not have to be the best collection of words from an article. Knowing what section should be cut and kept might improve results. Lastly it could be interesting to create and fine-tune a BERTje model specifically for this learning task to find out if it could be a good way of distinguishing between news and opinion.

4.9 Future implications

A very interesting question we did not cover in this thesis is: "how could news and opinion classification be used in our everyday lives?", keeping in mind that the problem we are trying to solve is that people have trouble making a distinction between news and opinion articles. A practical application could be a web browser extension that would tell a reader: "This article is 90% likely to be news". A cutoff point could be defined where this pop up could apply. For example, if the model would only be 60% sure, a message with "could not classify article" should appear.
Additionally, the news and opinion classifier could serve as a proxy for other kind of classifications. For example, if we wanted to classify articles as being either click bait or not and we would find out that opinion articles are more likely to be click bait than news articles. Then we could use the results of the news and opinion classifier as features for the click bait classifier. Of course such a correlation should be proven first and until then this idea remains speculative.

5 Conclusions

This thesis opened with two questions: to what extent can we classify Dutch text documents as news or opinion using machine learning? And: can we find patterns in the results with which we can explain the behaviour of the classifier? In this section we will answer these questions and reflect on the outcomes of this research.

For the first question we have found that news and opinion classification is a reasonable task that can yield good results. We have found that the strongest classifier for this task is SVM with a radial kernel. It seems that the removal of Dutch nltk stopwords does not significantly reduce the accuracy score. This means that those words likely carry little importance when it comes to deciding whether an article is a news or opinion article. The experiments have also shown us that the quality of the model degrades over the years and a functional model will have to be updated from time to time. To verify whether this model can function in a practical scenario, an out of sample test should also be done to make sure that the model is not just completely biased to the paper it was trained on.

Experiments with BERTje have shown us that BERT is a very powerful tool with a lot of potential for news and opinion classification. We have seen accuracy scores that neared those of models trained without BERTje that had training sets more than 10 times larger and could use each words in an article. More research should be done on the exact use of BERTje to use it in practice. After all, there are still a lot of unknowns such as out of time tests and out of source tests.

Regarding the second question we have seen that classifiers are in fact able to reproduce important patterns in both news and opinion articles. The most clear patterns in the news articles were those of a time indication and the beginning of a quotation. The strongest patterns in the opinion articles were words that were strongly related to the structure of proposition, argumentation and conclusion.

In conclusion we have observed that classification is promising and can offer a lot of potential in the online media world. Deploying classifiers with the goal of making information more reliable and guiding people through their online media environment is a concept worth researching further.

References

Aizawa, A. (2003). An information-theoretic perspective of tf–idf measures. *Information Processing & Management*, *39*(1), 45–65.

Aldisert, R. J. (2009). *Opinion writing*. AuthorHouse.

Alhindi, T., Muresan, S., & Preoţiuc-Pietro, D. (2020). Fact vs. opinion: the role of argumentation features in news classification. In *Proceedings of the 28th international conference on computational linguistics* (pp. 6139–6149).

Breiman, L., Cutler, A., & Stevens, J. R. (2011). Random forests. In *Ensemble machine learning: Methods and applications* (p. 157-176).

Bright, L. F. (2008). Consumer control and customization in online environments: An investigation into the psychology of consumer choice and its impact on media enjoyment, attitude, and behavioral intention. *Texas ScholarWorks*.

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 785–794).

de Vries, W., van Cranenburgh, A., Bisazza, A., Caselli, T., van Noord, G., & Nissim, M. (2019). BERTje: A dutch bert model. *arXiv preprint arXiv:1912.09582*. `https://github.com/wietsedv/bertje`.

Fortunati, L., Sarrica, M., O'Sullivan, J., Balcytiene, A., Harro-Loit, H., Macgregor, P., ... De Luca, F. (2009). The influence of the internet on european journalism. *Journal of Computer-Mediated Communication*, *14*(4), 928–963.

Kleinbaum, D. G., Dietz, K., Gail, M., Klein, M., & Klein, M. (2002). *Logistic regression*. Springer.

Loker, K. (2018). *Confusion about what's news and what's opinion is a big problem, but journalists can help solve it.* `https://www.americanpressinstitute.org/publications/confusion-about-whats-news-and-whats-opinion-is-a-big-problem-but-journalists-can-help-solve-it/`. American Press Institute.

Mitchell, A. (2018). *Distinguishing between factual and opinion statements in the news*. Pew Research Center.

Rish, I., et al. (2001). An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence* (Vol. 3, pp. 41–46).

Schelling, J., van Eekelen, N., van Veelen, I., van Hees, M., & van der Putten, P. (2020). Bursting the bubble. *2nd Multidisciplinary International Symposium on Disinformation in Open Online Media (MISDOOM)*.

Schölkopf, B., Smola, A. J., Williamson, R. C., & Bartlett, P. L. (2000). New support vector algorithms. *Neural computation*, *12*(5), 1207–1245.

Singer, J. B. (2008). Five ws and an h: Digital challenges in newspaper newsrooms and boardrooms. *The International Journal on Media Management*, *10*(3), 122–129.

Smart, C. B. (2016). The signal and the noise in cost estimating. *2016 International Training Symposium*. `http://www.iceaaonline.com/bristol2016/`.

Turc, I., Chang, M.-W., Lee, K., & Toutanova, K. (2019). Well-read students learn better: On the importance of pre-training compact models. *arXiv preprint arXiv:1908.08962*.

Wiebe, J., Bruce, R., & O'Hara, T. P. (1999). Development and use of a gold-standard data set for subjectivity classifications. In *Proceedings of the 37th annual meeting of the association for computational linguistics* (pp. 246–253).

Wiebe, J., et al. (2000). Learning subjective adjectives from corpora. *AAAI/IAAI*, *20*.

Wiebe, J., Wilson, T., Bruce, R., Bell, M., & Martin, M. (2004). Learning subjective language. *Computational linguistics*, *30*(3), 277–308.

Xia, M. X. (2016). Node-nltk-stopwords. https://github.com/xiamx/node-nltk-stopwords/blob/master/data/stopwords/dutch.

Yu, H., & Hatzivassiloglou, V. (2003). Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of the 2003 conference on empirical methods in natural language processing* (pp. 129–136).

6 Appendix

6.1 Contributions

The contribution from V. Kumar can be found on:
https://stackoverflow.com/questions/52042843/splitting-coef-into-arrays-applicable-for-multi-class.
The permissions can be found here:
https://stackoverflow.com/legal/terms-of-service

The contributions of N. Panwar can be found here:
https://medium.com/naukri-engineering/text-classification-using-bert-sklearn-and-pytorch-7665433b56c7
The permissions can be found here:
https://policy.medium.com/medium-terms-of-service-9db0094a1e0f

6.2 Top features Naive Bayes

Top 50 feature words for nieuws (NB):
‘werd‘ ‘wel‘ ‘hun‘ ‘tegen‘ ‘nu‘ ‘dit‘ ‘na‘ ‘geen‘ ‘zo‘ ‘wordt‘ ‘of‘ ‘worden‘ ‘jaar‘ ‘zich‘ ‘ze‘ ‘tot‘ ‘dan‘ ‘meer‘ ‘hebben‘ ‘al‘ ‘was‘ ‘hij‘ ‘nog‘ ‘over‘ ‘heeft‘ ‘naar‘ ‘uit‘ ‘door‘ ‘als‘ ‘maar‘ ‘er‘ ‘bij‘ ‘om‘ ‘aan‘ ‘niet‘ ‘ook‘ ‘die‘ ‘voor‘ ‘met‘ ‘te‘ ‘zijn‘ ‘is‘ ‘dat‘ ‘op‘ ‘en‘ ‘een‘ ‘van‘ ‘het‘ ‘in‘ ‘de‘

Top 50 feature words for opinie
‘kunnen‘ ‘je‘ ‘deze‘ ‘was‘ ‘kan‘ ‘hun‘ ‘dit‘ ‘wel‘ ‘wordt‘ ‘nu‘ ‘tot‘ ‘ze‘ ‘worden‘ ‘hebben‘ ‘zich‘ ‘wat‘ ‘meer‘ ‘al‘ ‘geen‘ ‘naar‘ ‘heeft‘ ‘uit‘ ‘zo‘ ‘nog‘ ‘of‘ ‘bij‘ ‘over‘ ‘door‘ ‘dan‘ ‘er‘ ‘om‘ ‘ook‘ ‘aan‘ ‘als‘ ‘maar‘ ‘de‘ ‘voor‘ ‘van‘ ‘met‘ ‘zijn‘ ‘die‘ ‘niet‘ ‘op‘ ‘te‘ ‘is‘ ‘dat‘ ‘en‘ ‘een‘ ‘in‘ ‘het‘

6.3 Tables

Settings	Label	Accuracy	Precision	Recall	F1	Train size
NB	Nieuws	0.77 ±0.01	0.76 ±0.01	0.78 ±0.00	0.77 ±0.01	21,500
	Opinie	0.77 ±0.01	0.77 ±0.00	0.76 ±0.02	0.77 ±0.01	21,500
NB 2019 validation	Nieuws	0.62	0.66	0.49	0.56	18,000
	Opinie	0.62	0.60	0.75	0.66	18,000
Log Reg	Nieuws	0.84 ±0.00	0.86 ±0.01	0.81 ±0.01	0.84 ±0.00	21,500
	Opinie	0.84 ±0.00	0.82 ±0.01	0.87 ±0.02	0.84 ±0.00	21,500
Log Reg 2019 validation	Nieuws	0.80	0.82	0.78	0.80	18,000
	Opinie	0.80	0.79	0.82	0.81	18,000
lin SVM	Nieuws	0.86 ±0.00	0.87 ±0.00	0.84 ±0.00	0.85 ±0.00	21,500
	Opinie	0.86 ±0.00	0.84 ±0.00	0.88 ±0.00	0.86 ±0.00	21,500
lin SVM 2019 test	Nieuws	0.84	0.85	0.83	0.84	18,000
	Opinie	0.84	0.84	0.86	0.85	18,000
Random forest	Nieuws	0.8 ±0.00	0.85 ±0.01	0.71 ±0.00	0.78 ±0.00	21,500
	Opinie	0.8 ±0.00	0.75 ±0.00	0.88 ±0.01	0.81 ±0.00	21,500
RF 2019 validation	Nieuws	0.67	0.77	0.48	0.59	18,000
	Opinie	0.67	0.62	0.86	0.72	18,000
XGB	Nieuws	0.85 ±0.01	0.87 ±0.02	0.81 ±0.00	0.84 ±0.01	21,500
	Opinie	0.85 ±0.01	0.82 ±0.00	0.88 ±0.01	0.85 ±0.00	21,500
XGB 2019 validation	Nieuws	0.83	0.85	0.81	0.83	18,000
	Opinie	0.83	0.82	0.85	0.83	18,000
SVM	Nieuws	0.87 ±0.00	0.90 ±0.01	0.83 ±0.00	0.86 ±0.00	21,500
	Opinie	0.87 ±0.00	0.84 ±0.00	0.90 ±0.00	0.87 ±0.00	21,500
SVM 2019 validation	Nieuws	0.85	0.87	0.82	0.84	18,000
	Opinie	0.85	0.83	0.88	0.85	18,000

Table 11: Results from all classifiers trained with default settings

Settings	Label	Accuracy	Precision	Recall	F1	Train size
NB	Nieuws	0.79 ±0.01	0.77 ±0.02	0.81 ±0.00	0.79 ±0.01	21,500
	Opinie	0.79 ±0.01	0.80 ±0.00	0.76 ±0.02	0.78 ±0.01	21,500
NB 2019 validation	Nieuws	0.65	0.68	0.57	0.62	18,000
	Opinie	0.65	0.63	0.74	0.68	18,000
Log Reg	Nieuws	0.86 ±0.01	0.88 ±0.02	0.82 ±0.02	0.85 ±0.01	21,500
	Opinie	0.86 ±0.01	0.83 ±0.01	0.89 ±0.02	0.86 ±0.00	21,500
Log Reg 2019 validation	Nieuws	0.74	0.78	0.69	0.73	18,000
	Opinie	0.74	0.72	0.80	0.76	18,000
lin SVM	Nieuws	0.87 ±0.00	0.88 ±0.00	0.85 ±0.01	0.86 ±0.00	21,500
	Opinie	0.87 ±0.00	0.85 ±0.00	0.89 ±0.00	0.87 ±0.00	21,500
lin SVM 2019 validation	Nieuws	0.78	0.78	0.77	0.78	18,000
	Opinie	0.78	0.78	0.78	0.78	18,000
Random forest	Nieuws	0.82 ±0.01	0.87 ±0.01	0.75 ±0.01	0.80 ±0.00	21,500
	Opinie	0.82 ±0.01	0.78 ±0.01	0.89 ±0.01	0.83 ±0.00	21,500
RF 2019 validation	Nieuws	0.66	0.77	0.46	0.58	18,000
	Opinie	0.66	0.62	0.87	0.72	18,000
XGB	Nieuws	0.86 ±0.01	0.89 ±0.02	0.82 ±0.00	0.85 ±0.01	21,500
	Opinie	0.86 ±0.01	0.84 ±0.00	0.89 ±0.01	0.86 ±0.00	21,500
XGB 2019 validation	Nieuws	0.75	0.78	0.70	0.74	18,000
	Opinie	0.75	0.73	0.81	0.77	18,000
SVM	Nieuws	0.87 ±0.00	0.90 ±0.00	0.83 ±0.00	0.87 ±0.01	21,500
	Opinie	0.87 ±0.00	0.84 ±0.00	0.91 ±0.00	0.88 ±0.00	21,500
SVM 2019 validation	Nieuws	0.78	0.81	0.72	0.76	18,000
	Opinie	0.78	0.75	0.83	0.79	18,000

Table 12: Model trained on articles from 2009 - 2015 and tested on articles from 2016 - 2019

Settings	Label	Accuracy	Precision	Recall	F1	Train size
NB	Nieuws	0.76 ±0.00	0.74 ±0.00	0.80 ±0.00	0.77 ±0.00	21,500
	Opinie	0.76 ±0.00	0.78 ±0.01	0.72 ±0.00	0.75 ±0.00	21,500
Log Reg	Nieuws	0.85 ±0.00	0.87 ±0.01	0.83 ±0.00	0.85 ±0.00	21,500
	Opinie	0.85 ±0.00	0.84 ±0.01	0.87 ±0.01	0.85 ±0.00	21,500
lin SVM	Nieuws	0.86 ±0.00	0.88 ±0.01	0.84 ±0.01	0.86 ±0.00	21,500
	Opinie	0.86 ±0.00	0.85 ±0.01	0.88 ±0.00	0.86 ±0.00	21,500
Random forest	Nieuws	0.81 ±0.00	0.86 ±0.01	0.73 ±0.01	0.79 ±0.00	21,500
	Opinie	0.81 ±0.00	0.77 ±0.00	0.88 ±0.01	0.82 ±0.00	21,500
XGB	Nieuws	0.85 ±0.00	0.87 ±0.01	0.83 ±0.01	0.85 ±0.00	21,500
	Opinie	0.85 ±0.00	0.84 ±0.01	0.87 ±0.01	0.85 ±0.00	21,500
SVM	Nieuws	0.87 ±0.00	0.89 ±0.00	0.83 ±0.00	0.86 ±0.00	21,500
	Opinie	0.87 ±0.00	0.84 ±0.01	0.90 ±0.00	0.87 ±0.00	21,500

Table 13: Results of all classifiers with default settings and the removal of stopwords

Settings	Label	Accuracy	Precision	Recall	F1	Train size
NB	Nieuws	0.77 ±0.00	0.76 ±0.00	0.78 ±0.00	0.77 ±0.00	21,500
	Opinie	0.77 ±0.00	0.78 ±0.00	0.76 ±0.01	0.77 ±0.01	21,500
NB 2019 validation	Nieuws	0.62	0.66	0.49	0.56	18,000
	Opinie	0.62	0.59	0.75	0.66	18,000
Log Reg	Nieuws	0.83 ±0.00	0.84 ±0.03	0.83 ±0.03	0.83 ±0.00	21,500
	Opinie	0.83 ±0.00	0.83 ±0.01	0.84 ±0.03	0.83 ±0.01	21,500
Log Reg 2019 validation	Nieuws	0.77	0.79	0.74	0.76	18,000
	Opinie	0.77	0.75	0.80	0.78	18,000
lin SVM	Nieuws	0.82 ±0.01	0.81 ±0.00	0.84 ±0.01	0.83 ±0.01	21,500
	Opinie	0.82 ±0.01	0.83 ±0.01	0.81 ±0.01	0.82 ±0.00	21,500
lin SVM 2019 validation	Nieuws	0.80	0.79	0.81	0.80	18,000
	Opinie	0.80	0.81	0.78	0.80	18,000
Random forest	Nieuws	0.79 ±0.00	0.85 ±0.00	0.70 ±0.00	0.77 ±0.00	21,500
	Opinie	0.79 ±0.00	0.74 ±0.01	0.88 ±0.00	0.81 ±0.00	21,500
RF 2019 validation	Nieuws	0.64	0.76	0.40	0.52	18,000
	Opinie	0.64	0.59	0.88	0.71	18,000
XGB	Nieuws	0.85 ±0.00	0.88 ±0.00	0.82 ±0.01	0.85 ±0.00	21,500
	Opinie	0.85 ±0.00	0.83 ±0.01	0.88 ±0.00	0.86 ±0.01	21,500
XGB 2019 validation	Nieuws	0.80	0.84	0.74	0.79	18,000
	Opinie	0.80	0.77	0.86	0.81	18,000
SVM	Nieuws	0.85 ±0.00	0.87 ±0.00	0.83 ±0.01	0.85 ±0.00	21,500
	Opinie	0.85 ±0.00	0.83 ±0.01	0.88 ±0.00	0.85 ±0.00	21,500
SVM 2019 validation	Nieuws	0.83	0.84	0.82	0.83	18,000
	Opinie	0.83	0.82	0.85	0.83	18,000

Table 14: Results of all classifiers when using default settings and bag of words representation

Settings	Label	Accuracy	Precision	Recall	F1	Train size
NB	Nieuws	0.73 ±0.02	0.74 ±0.02	0.69 ±0.04	0.71 ±0.03	1200
	Opinie	0.73 ±0.02	0.72 ±0.02	0.76 ±0.01	0.74 ±0.02	1200
Log Reg	Nieuws	0.69 ±0.02	0.68 ±0.03	0.70 ±0.04	0.69 ±0.02	1200
	Opinie	0.69 ±0.02	0.70 ±0.01	0.68 ±0.04	0.69 ±0.03	1200
lin SVM	Nieuws	0.67 ±0.02	0.66 ±0.02	0.68 ±0.05	0.67 ±0.03	1200
	Opinie	0.67 ±0.02	0.68 ±0.03	0.67 ±0.01	0.67 ±0.01	1200
Random forest	Nieuws	0.72 ±0.00	0.71 ±0.01	0.74 ±0.01	0.72 ±0.01	1200
	Opinie	0.72 ±0.00	0.74 ±0.00	0.71 ±0.02	0.72 ±0.00	1200
XGB	Nieuws	0.72 ±0.01	0.72 ±0.02	0.72 ±0.01	0.72 ±0.01	1200
	Opinie	0.72 ±0.01	0.73 ±0.00	0.72 ±0.01	0.73 ±0.01	1200
SVM	Nieuws	0.75 ±0.01	0.76 ±0.02	0.73 ±0.01	0.75 ±0.01	1200
	Opinie	0.75 ±0.01	0.75 ±0.01	0.78 ±0.02	0.76 ±0.01	1200

Table 15: Default settings with BERTje embedding: results with maximum token length 40 and with no stopword removal

Settings	Label	Accuracy	Precision	Recall	F1	Train size
NB	Nieuws	0.74 ±0.01	0.74 ±0.03	0.73 ±0.03	0.74 ±0.02	1200
	Opinie	0.74 ±0.01	0.74 ±0.02	0.75 ±0.03	0.74 ±0.01	1200
Log Reg	Nieuws	0.67 ±0.08	0.65 ±0.09	0.80 ±0.12	0.71 ±0.01	1200
	Opinie	0.67 ±0.08	0.74 ±0.03	0.55 ±0.31	0.61 ±0.25	1200
lin SVM	Nieuws	0.68 ±0.01	0.67 ±0.02	0.70 ±0.03	0.69 ±0.02	1200
	Opinie	0.68 ±0.01	0.69 ±0.03	0.66 ±0.00	0.67 ±0.01	1200
Random forest	Nieuws	0.73 ±0.03	0.72 ±0.02	0.74 ±0.04	0.73 ±0.03	1200
	Opinie	0.73 ±0.03	0.74 ±0.05	0.72 ±0.02	0.73 ±0.03	1200
XGB	Nieuws	0.73 ±0.03	0.73 ±0.02	0.72 ±0.05	0.72 ±0.02	1200
	Opinie	0.73 ±0.03	0.72 ±0.04	0.73 ±0.02	0.73 ±0.03	1200
SVM	Nieuws	0.76 ±0.03	0.78 ±0.02	0.73 ±0.05	0.75 ±0.03	1200
	Opinie	0.76 ±0.03	0.75 ±0.05	0.79 ±0.02	0.77 ±0.02	1200

Table 16: Default settings with BERTje embedding: results with maximum token length 40 and with stopword removal

Settings	Label	Accuracy	Precision	Recall	F1	Train size
NB	Nieuws	0.83 ±0.02	0.87 ±0.01	0.80 ±0.07	0.83 ±0.03	300
	Opinie	0.83 ±0.02	0.80 ±0.03	0.86 ±0.04	0.83 ±0.00	300
Log Reg	Nieuws	0.81 ±0.03	0.86 ±0.07	0.76 ±0.06	0.81 ±0.03	300
	Opinie	0.81 ±0.03	0.77 ±0.04	0.86 ±0.09	0.81 ±0.03	300
lin SVM	Nieuws	0.82 ±0.02	0.84 ±0.04	0.79 ±0.05	0.82 ±0.04	300
	Opinie	0.82 ±0.02	0.78 ±0.03	0.84 ±0.07	0.81 ±0.02	300
Random forest	Nieuws	0.83 ±0.04	0.86 ±0.02	0.81 ±0.06	0.83 ±0.04	300
	Opinie	0.83 ±0.04	0.80 ±0.06	0.85 ±0.04	0.82 ±0.04	300
XGB	Nieuws	0.82 ±0.02	0.84 ±0.01	0.81 ±0.06	0.82 ±0.03	300
	Opinie	0.82 ±0.02	0.80 ±0.03	0.83 ±0.03	0.82 ±0.01	300
SVM	Nieuws	0.85 ±0.02	0.91 ±0.01	0.79 ±0.05	0.85 ±0.03	300
	Opinie	0.85 ±0.02	0.80 ±0.02	0.92 ±0.01	0.86 ±0.01	300

Table 17: Default settings with BERTje embedding: results with maximum token length 256 and with no stopword removal

Settings	Label	Accuracy	Precision	Recall	F1	Train size
NB	Nieuws	0.81 ±0.07	0.87 ±0.06	0.74 ±0.11	0.8 ±0.09	300
	Opinie	0.81 ±0.07	0.78 ±0.08	0.89 ±0.05	0.83 ±0.06	300
Log Reg	Nieuws	0.82 ±0.01	0.81 ±0.02	0.84 ±0.04	0.82 ±0.01	300
	Opinie	0.82 ±0.01	0.84 ±0.02	0.81 ±0.04	0.82 ±0.02	300
lin SVM	Nieuws	0.78 ±0.05	0.79 ±0.04	0.76 ±0.06	0.78 ±0.05	300
	Opinie	0.78 ±0.05	0.78 ±0.06	0.80 ±0.04	0.79 ±0.05	300
Random forest	Nieuws	0.82 ±0.04	0.83 ±0.03	0.79 ±0.05	0.82 ±0.04	300
	Opinie	0.82 ±0.04	0.81 ±0.05	0.85 ±0.03	0.83 ±0.04	300
XGB	Nieuws	0.80 ±0.02	0.81 ±0.03	0.77 ±0.02	0.79 ±0.03	300
	Opinie	0.80 ±0.02	0.79 ±0.03	0.83 ±0.03	0.80 ±0.02	300
SVM	Nieuws	0.84 ±0.04	0.90 ±0.05	0.76 ±0.06	0.83 ±0.05	300
	Opinie	0.84 ±0.04	0.80 ±0.05	0.92 ±0.04	0.86 ±0.04	300

Table 18: Default settings with BERTje embedding: results with maximum token length 256 and with stopword removal

Year	Articles (news)	Articles (opinion)	Percentage (news)	Percentage (opinion)
2009	1561	1275	10.91	8.89
2010	1661	1364	11.61	9.51
2011	1950	926	13.63	6.46
2012	2116	394	14.79	2.75
2013	2008	439	14.03	3.06
2014	1719	1258	12.02	8.77
2015	1187	1495	8.30	10.42
2016	938	1759	6.56	12.26
2017	711	1499	4.97	10.45
2018	455	3897	3.18	27.17

Table 19: Example Distribution of years from a randomly drawn data set