Universiteit Leiden
The Netherlands

Opleiding Informatica & Economie

Differentiating Commercial and Editorial Content

Timo Kats

Supervisors:
Peter van der Putten & Jasper Schelling

BACHELOR THESIS

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

09/07/2021

Abstract

This thesis centers around differentiating commercial and editorial content in the form of advertorials and articles. Note that an advertorial is essentially a commercial message written and formatted as an article, making them hard for readers to recognize as commercial content. That's why this research aims to differentiate the two using a machine learning model, and a lexicon derived from that machine learning model. This was accomplished by scraping 1000 articles and 1000 advertorials from four different Dutch news sources and classifying them based on their text. With this setup our most successful machine learning model had a test accuracy of just over 90% and our most successful lexicon had a train accuracy of just over 98%. Thus, we conclude that machine learning can successfully differentiate commercial and editorial content and we can also successfully derive a lexicon from that model that aims to complete the same task.

Contents

1 Introduction

Advertising is a major part of journalism's business model. In fact, the presence of ads in the news relates to one of journalism's most essential principles: the clear distinction there should be between the news' editorial and commercial content. In journalism this is even commonly referred to as their separation of church and state [1].
However, the practical application of this separation has become vastly more difficult in the digital age. Websites can attract a larger group of readers and allow for more advertisement space than traditional print. By opening up this additional channel for news organizations to attract revenue from, their business model changed. For example, in the annual report from The New York Times for 2019 [2] revenue from digital advertising ($ 260,454,000) almost amounts to the same dollar figure as advertising on print ($ 270,224,000). This radical change in revenue streams would have been unthinkable 20 years ago and it's partly created by the rise of a new form of digital advertising: the advertorial.
An advertorial is essentially a commercial message formatted as an article, often branded by a small disclaimer. This type of advertising is currently experiencing a meteoric rise in the media. In fact, a study conducted by Enders Analysis expected that the usage of advertorials in Western Europe would increase with 156% between 2015 and 2020 [3]. This would make 52% of all digital advertising an advertorial.
The reason advertorials experience this rise and popularity with news sources is largely because of how successful they are at targeting consumers. This is especially true when their performance is being compared with traditional banner advertisements, which used to dominate the digital advertisement space. Those type of advertisements have become less important over the years because they prompt a less positive response from consumers than advertorials [4].
But how exactly do advertorials accomplish that? The answer to that is simple: most people don't even know that they're reading a commercial message. According to a research conducted by the university of Georgia [5] only 8% of participants recognized advertorials as advertising. This creates a major problem. Because if readers can't separate commercial and editorial content, is there still a clear commercial and editorial divide?
No, there isn't. That's why this research will focus on differentiating commercial and editorial content in the form of advertorials and articles with machine learning through answering two different research questions. By completing our research we not only strive to answer these research questions, but also to showcase how machine learning and AI can be a solution, not a problem, in the modern media landscape.

1.1 Research questions

As mentioned, this research is about differentiating advertorials and articles. We strive to accomplish this by stating and answering two research questions. These research questions are:

1. To what extent can we differentiate commercial and editorial content by using machine learning?

2. Can we derive a lexicon from our machine learning model that differentiates commercial and editorial language?

We aim to answer the first research question with a machine learning model (and it's parameters). The second research question will be answered by the performance of the lexicon that we derived from that model. The lexicon itself will also be delivered as a result from this research.

1.2 Thesis Overview

This thesis is structured as follows. The first part is the introduction and the statement of the research questions. The second part explores related work, in particular from a journalistic perspective. After that we state our research goals and give credit to contributions other people have added to this research.
From this point we'll start the process of answering our research questions stated earlier. This is done in four steps. Firstly we'll show our data acquisition. After that we share our experiments and results, followed by a discussion. Finally we take our results and draw our conclusions.

2 Related work

The discussion around the usage of advertorials in journalism is broader than this research alone. That's why in this section we discuss the findings of other researchers who have done work in this field. Firstly, we explore the change of journalism's business model in the digital age, and how that relates to advertorials. Secondly we'll discuss advertorials a bit more in to detail, especially from a psychology and marketing perspective. The third and final part of this section is about how lexicons have been used in the past by other researchers in classification problems that are comparable to ours.

2.1 The change of journalism's business model in the digital age

The rise of the internet has had a lot of effect on journalism. On one hand it opened up a whole new medium for news organizations to broadcast to, on the other hand it diminished the news on other mediums. The latter is especially true for newspapers, since the need for print has decreased dramatically due to the creation of the internet. In fact, according to a research conducted by the OECD (Organization for Economic Co-operation and Development) [6] this industry in the Netherlands has declined by 6% between 2007 and 2009 (in the US it was even more with 30%).
This change has put the business model of traditional news organizations in a radically different position. Historically newspapers largely rely on two revenue streams: income from ads on printed newspapers and income from newspaper sales [6]. Both of these revenue streams have become less profitable over the last couple of years. According to a research conducted by the European Union [7] revenue from advertising on print and the circulation of newspapers have decreased for every year between 2006 and 2010. The gap that this leaves has been filled up by online advertising, which has grown spectacularly over the last couple of years.
Online advertising is now one of the largest income sources for newspapers. Even as early as 2014 research from the Newspaper Association of America [8] showed that the revenue from advertising on print was declining whilst the revenue from digital advertising was increasing. Moreover, some news corporations nowadays exclusively rely on digital advertising and only have an online presence.

An example of this would be Buzzfeed, which attracted more than 7 billion monthly global views in 2016 [9] without the usage of print, radio or TV. These type of news companies are an important segment of the news nowadays and their existence relies wholly on the internet. This would have been unthinkable 20 years ago and it's emblematic for how journalism's business model has been completely redefined in the digital age.

2.2 The usage of disguise and disclosure in advertorials

To reiterate, advertorials are commercial messages formatted as editorial articles. The way advertorials are presented makes a huge difference in the way they are perceived. As mentioned, in a research conducted by the University of Georgia [5] only 8% of participants recognized advertorials as advertising. This means that 92% of disclosures went by unnoticed. What this shows us is that the usage and recognizability of disclaimers is minimal to say the least, even though it's a legal requirement to have them.
Naturally, the more serious media outlets such as major newspapers will make more of an effort to let you know certain content is sponsored. Nonetheless, that research conducted by the University of Georgia that showed only 8% of respondents recognized advertorials as advertising was "based on a real U.S.-based midsized market newspaper" [5]. Illustrating that also more mainstream news sources probably overestimate the ability for readers to recognize commercial content.
The effect of disclaimers and how they are placed has been explored by researchers from the University of Antwerp [10]. In their research they found that small changes can make a substantial difference in disclosure recognition. For example the placement of the disclaimer; if a disclaimer is placed at the top of the text recognition is lower than when a disclaimer is placed in the middle or at the bottom of the text. By using techniques like this news sources can actively try to disguise sponsored content instead of disclose it. Which could be the reason recognizability among readers is so low.

2.3 The usage of lexicons in classifying text

As mentioned, for our second research question we aim to create a lexicon that separates editorial and commercial language. The concept of using lexicons to classify text has been used before, often to separate positive and negative language. This is known to as "sentiment analysis".
In a lot of ways "sentiment analysis" is similar to our classification problem, since we also want to classify text in two different categories. The only difference is that our categories are commercial and editorial instead of positive and negative. Hence we can look at the work that has already been done in the field of "sentiment analysis" as a template for the creation of our lexicon.
An example of a lexicon used to do sentiment analysis would be "The Semantic Orientation CALculator (SO-CAL)" [11], which to our knowledge is the first research of this sort. This calculator has a score per word (a lexicon) that it uses to determine whether a piece of text is generally positive or negative and to what degree. It was trained and tested using a movie-review data set on which it scored around 80% accuracy [11], showing the potential of lexicon based classification models.
This potential combined with the fact that lexicons can be published and used without also needing to publish the training and test data of this research is the reason why our second research question aims at creating a lexicon of this sort for editorial and commercial language.

3 Research goals and contributions

In this section we state our research goals and what our research contributes to both science and society. The latter refers to our deliverables as well as the void within science we want to fill.

3.1 Research goals

As stated in the introduction our main goal is to differentiate editorial and commercial content using machine learning. We compartmentalize this by firstly differentiating the two with a machine learning model and secondly differentiating the two with a lexicon that is derived from that model. Besides that main goal however, there are also some other things we like to achieve with this research. Firstly, we hope this research can showcase how machine learning and AI can be a solution, not a problem, in the modern media landscape. Secondly, we hope this research spreads awareness about the importance of the divide there should be between editorial and commercial content in the news. Thirdly, we aim to better our understanding of how the text types differ in terms of the language and topics they use.
Finally, we also aim to fill what we see as a void in scientific literature. Since most research about this subject seems to predominantly come from a psychology and marketing perspective we want to add a computer science perspective as well.

3.2 Contributions

In terms of the contributions this research makes (besides the answers to the research questions) we have our source code, our final model, our visualizations, our lexicon, and a draft for a blogpost that will be published on the website of Reverb Channel [12]. The settings of the final model can be found in this research's results section (see Table 13). The source code and lexicon are stored in this research's GitHub repository at: `https://github.com/TimoKats/bachelor_project`. Finally, our visualization can be found at: `https://timokats.github.io/network/`

4 Data acquisition

In order to make a model that answers the research questions mentioned earlier we needed to create a data set that has advertorials and articles. Reverb Channel (which is "a data driven research programme, that focuses on the impact of contemporary applications of data mining and machine learning in digital news media." [12]) has a corpus that contains millions of articles, but no advertorials. Hence we had to acquire our own data for this research. In this section we explain this process and showcase the data set that we acquired.

4.1 Scraping the data

As mentioned, the data that was required for this research was not available in the corpus from Reverb Channel. That's why the data had to be scraped directly from news sources. This process consisted of two separate steps.

The first step was getting the URLs that link to the advertorials and articles per news source. The second step was to use those URLs as input to scrape and format the data from the pages they link to. The implementation of this was done in Python using the popular web scraping library "BeautifulSoup" [13]. However, because of small differences between news sources the Python code for every news source differs. That's why in this research we represent the scraping algorithms in a more general from through pseudo-code.

4.1.1 Collecting the URLs

The first step in creating the data set is collecting the URLs that link to the pages we wanted to scrape data from. The great thing about collecting URLs from news sites is that (with most news sites) all pages are identified by a certain index number. The way this index is formatted or used in the URL differs per news site but it's application within a given news source is consistent. Hence iterating through these indices makes it possible to brute force a lot of URLs that link to both the advertorials and the articles that are stored on the news' website.
In our URL scraper we used this to iterate through all the possible URLs (see Algorithm 1). If a URL referred to a valid link the algorithm checks whether the page is an article or an advertorial. If the page is an advertorial the URL will be appended to a file for advertorials, else it would be appended to a file for articles.
However, not all newspapers allowed for this kind of brute-force web scraping. In those cases the advertorials were collected manually by going to the sponsor-pages and the articles we collected through the archive-pages from that news source. To iterate through the archive pages from a news source we used the publication date (see Algorithm 2). Because we want to spread out the dates from our articles and not collect everything from one publication date we implemented a threshold of articles our scraper could collect per day. In our research we used a threshold of 10.

4.1.2 Scraping data from the URLs

The second step is using the URLs generated by the URL collector to acquire the data. This is done by iterating through all the links that the URL collector provides and use it as input. For every URL that the program encounters every property shown in Table 1 is collected. This data is thereafter exported to a csv-file, which is also the final output of this program. Since the sponsor property can only be acquired for articles the program has to be run separately for articles. A simplified pseudo code version of this process is shown in Algorithm 3.

4.1.3 Cleaning and encoding the data

Most of the preprocessing of the data is done during the experiments instead of the data acquisition. This means that, with the exception of commas, all the punctuation and capital letters are kept unchanged in the data set. Commas are being removed directly in the data acquisition because they are used as separators in our csv files and therefore interfere with the columns. For NRC we also remove HTML-tags from the text using regular expressions. This is not a necessary step for other news sources since they store their text in a JSON-string instead of the HTML-code.
Finally, in terms of encoding, all the written output files are encoded with UTF-8 [14]. The input on the other hand (meaning the data we scrape from the URL before writing it to a file) is imported without encoding in a regular string.

Algorithm 1: URL collector through index (brute force)

```
Result: A file with links to the advertorials and a file with links to the articles.
index ← 0
while True do
    page ← get_page(index)
    if page exists then
        if page is an advertorial then
            append this index to advertorial file
        else
            append this index to article file
    index += 1
```

Algorithm 2: URL collector through dates (brute force)

```
Result: A file with links to articles.
index ← 0
threshold ← any arbitrary number
for year in years do
    for month in months do
        for day in days do
            pages ← get_pages(year, month, day)
            index = 0
            while index < threshold do
                append page[index] to article file
                index += 1
```

Algorithm 3: Data scraper

```
Data: A file with links to articles/advertorials.
Result: The data set (in csv format).
for URL in input file do
    ad ← scraper(URL) ;                    // Can also be article = scraper(URL)
    scrape all the data that belongs to this ad/article
    export this data to a csv-file
```

4.2 About the data

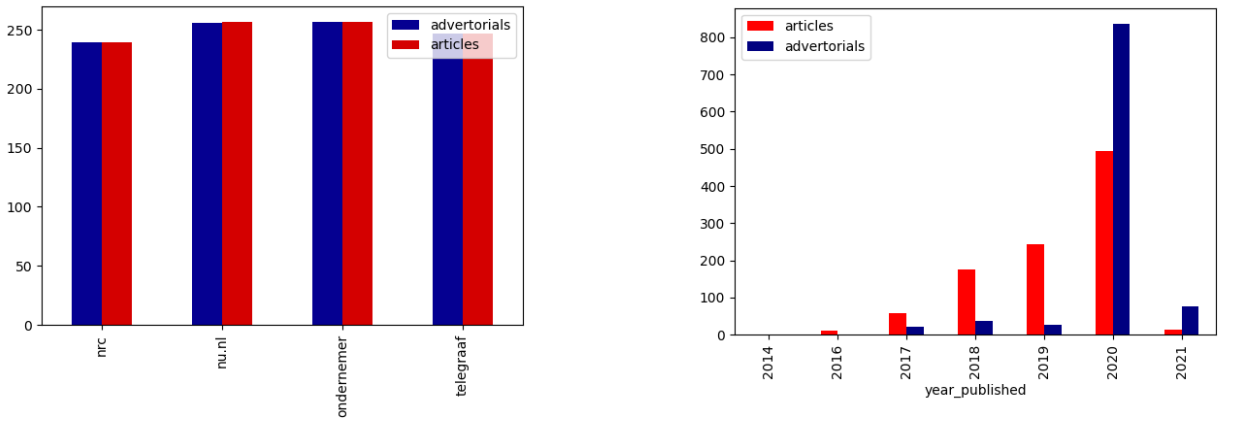
In order to practically use the data we acquired it had to be structured and explored. In this subsection we explain this by firstly highlighting the data itself and secondly by explaining how this data is structured.

4.2.1 Data exploration and results

The data set has 2000 entries in total, about half of these entries are advertorials. The reason only half of the data set belongs to advertorials is because we want to avoid incentivising the model to classify entries as advertorials simply because they are the majority of the data set. Even though that would probably increase the model's performance on our data set it would decrease it's performance on unseen data. Hence the even split between articles and advertorials.
Diversity of sponsors is also really important. We don't want the model to classify advertorials based on brand names that are unique to our data set, instead we want the model to recognize advertorials in a more general form. Therefore our data has 217 sponsors in total, of which the largest sponsor ONVZ has a 21.2% share (in terms of the amount of advertorials that it sponsors). Furthermore, for the same reason we diversified our sponsors, we also diversified our news sources (see Figure 1a). In total we've collected data from four different news sources, each adding around 250 advertorials and 250 editorial articles to our data set. The first news source we've collected data from is (online-only news) Nu.nl, the second news source is (politically conservative) Telegraaf, the third news source is (politically progressive) NRC, and finally the fourth news source is (B2B) Ondernemer. By having these four vastly different news sources we strive to create an unbiased data set that is representative of the Dutch media landscape as a whole.
Another important aspect in terms of diversification is the timestamp the entry originates from (see Figure 1b). For advertorials there is very little choice when it comes to this since most news sources remove advertorials after being online for a while. Thus, almost all advertorials in our data set are from 2020. For articles the situation is different because they get archived by news sources. However, to avoid a large mismatch in terms of timestamp we also mostly collected articles that are from 2020. The reason why not all articles are from 2020 (which is largely the case for advertorials) is related to bias. If a year has a major event (for example the covid-19 pandemic) then the articles from that year will disproportionately often be about that. This could result in our model having a bias, which is something we want to avoid.

Article property	**Description**
source	The news source the article/advertorial originates from.
date_published	The date the article/advertorial was published.
date_modified	The most recent date the article/advertorial was modified.
sponsor	The sponsor of the advertorial (states 'none' when the entry is an article).
title	The title of the advertorial/article.
introduction	The introduction of the advertorial/article.
body	The body of the advertorial/article.

Table 1: Properties of the data set



(a) Number of articles and advertorials per source

(b) Number of articles and advertorials per year

Figure 1: Metadata from the acquired data set

4.2.2 Layout of the data set

The data is stored in a csv-file format that has seven fields (see Table 1). With the exception of the field "sponsor" the application of the all fields remains consistent regardless of if an entry belongs to an article or an advertorial. This is because articles don't have a sponsor and therefore it can't be scraped either. Hence this field simply states "none" when an entry belongs to an article.

5 Experiments and results

After successfully acquiring the data we conducted our experiments. In this section we go through this process by first explaining our set up and then our results. Throughout this section we also share our decisions during the experiments and our motivations behind them.

5.1 Experimental set up

In order to create an experiment that can answer our research questions we needed to set up our experiment. In this subsection we discuss our experimental setup by firstly explaining our data selection and data representation. Secondly we discuss the preprocessing of the input. Thirdly we explain our experimental process. Finally we highlight how we record and evaluate our results.

5.1.1 Data selection and data representation

The data that we used as input for the machine learning process is the introduction of the text plus the body of the text (see Table 1). The reason why we used both is related to the formatting of (most) advertorials. A lot of newspapers store the actual advertisement part in the introduction in order to save space in the body for disclaimers and contact information. Hence we used both fields as input. We represented this input in two different ways; bag of words and tf-idf.

Bag of words bases itself on the vocabulary of the total data set, this is a list of all the words that exist in our data. Next it counts per article/advertorial the amount of times every word from that vocabulary appears in our text. This outputs an array of numbers that can be used for machine learning.
Tf-idf is similar to this process in the sense that it bases itself on the same vocabulary bag of words bases itself on. The only difference with tf-idf is that it also takes the "uniqueness" of a word into account. It does this by multiplying the frequency of every term (denoted by t in the formula below) in an article/advertorial by the inverse of the total frequency that word has in our data set. This process also outputs an array of numbers that can be used for machine learning and it's highlighted in the formula below.

$$TF = \frac{\text{term frequency in document}}{\text{total words in document}}$$

$$IDF(t) = \log_2\left(\frac{\text{total documents in data set}}{\text{documents with term}}\right)$$

$$\text{TF-IDF score for term t in data set} = \text{TF} \cdot \text{IDF(t)}$$

To reiterate, our problem (can we differentiate advertorials and articles) is a classification problem. Hence the target variable that we predict using the input explained earlier shows whether an entry is sponsored or not. We represented this target variable through the sponsor field (see Table 1). If this field states "none" then the entry is an article so our target variable is false (0), else it's an advertorial so the target becomes true (1).

5.1.2 Preprocessing the input

Our goal when making this model is that it's unbiased, generally applicable and efficient. However, when selecting a pseudo-random sample of media articles/advertorials it's of course very possible that biases and inefficiencies do occur. Hence, we removed terms that can create a bias in our model, also known as leaking variables.
Leaking variables in our model refer to words that trigger the model whilst being unique to our data set (see Figure 2). Examples of this would be sponsor names like "volvo" and "KPN" and words referring to disclaimers such as "gesponsord door" (sponsored by) and "valt buiten onze redactionele verantwoordelijkheid" (is outside our editorial responsibility). If the model uses these words as variables it would create a bias resulting in a model that could under perform on unseen data.
Next, we made all words lowercase. The reason why we convert words to lowercase is because capitalization doesn't change the meaning of a word. So if multiple words have the same characters in the same order but have different capitalization for those characters, the model should not treat them as unique words. Hence our model treats all versions of capitalization for a word the same by making them all lowercase before doing any machine learning.

5.1.3 Process

Since this is a classification problem (advertorial or article) we selected a diverse set of classification models to experiment with. These classification models are; SVM, linearSVC, decision tree, random forest, k-NN, SGD and naive bayes. We aim to find the best performing model (incl. parameter optimization) through narrowing down the search as the experiment progresses, taking the best performing preliminary results and continuing to optimize it. Also, for some models we needed to tweak the parameters in the setup in order to make them run. For SVM, SGD and linearSVC we increased the maximum amount of iterations to 5000 and for decision tree and random forest we set the max depth to "none".
A limitation that this process has is that (because we use preliminary results) we occasionally test on the same data we've trained on earlier, creating a slight effect of overfitting in our experiments.

5.1.4 Evaluation

We evaluate all our results using 10-fold cross validation with the exception of tables 4-10. These tables have a form of cross validation in the sense that they're trained on three news sources and tested on the fourth news source, which effectively creates a form of 4-fold cross validation with roughly a 75-25 train test ratio. The metrics that we evaluate our results with are accuracy, f1 score and AUC. All these metrics are shown along with their standard deviation. This means that for all metrics, the higher the score and the lower the standard deviation, the better.

5.2 Results

With the experimental setup as mentioned we got our results. In this subsection we go through these preliminary and final results for our machine learning model and the lexicon. Next, we show the results of exploring our data and our results.

5.2.1 Finding the right model

Before testing which model performs best on the data set we evaluated the effect of including and excluding stop words from the input. stop words are terms that are very common in a language and add very little meaning. Examples of stop words in Dutch would be "de" (the) "het" (it) and "Een" (a/an). The list of stop words that was used in our research was made by Gene Diaz [15] and it has 413 words in total.
The reason why we evaluate the effect of stop words is because machine learning is a computationally demanding task, making words that have very little effect are excess baggage. We evaluated the effect of our stop words in Tables 2 and 3. Table 2 in this case shows the performance of different models without removing stop words whilst Table 3 shows the performance of different models with the removal of stop words. The lack of difference between the results of these two tables showcase the lack of effect stop words have on the model. Hence in further experiments they were removed from the input.
After selecting the input we test which text representation gives the best results. The results from this experiment are shown in Table 3. What's apparent in this experiment is that tf-idf outperforms bag of words with every metric. For accuracy, f1 score and AUC tf-idf scores on average 0.80, 0.83 and 0.84 whilst bag of words scores 0.79, 0.80 and 0.83. This combined with the fact that the top

performing models (SVM, linearSVC and SGD) all seem to work best with tf-idf we selected it as the text representation to further our research with.
Thereafter we tested which model gives the best results. This is done in through testing each model with a form of 4-fold cross validation in newspapers explained earlier (see experimental set up). The results of this experiment per model are shown in Tables 4-10. The best performing model in this experiment is the SVM. Hence that's the model we'll continue experimenting with.
After selecting the model the next step was to optimize it's parameters. The first parameter that we've optimized was the amount of features (words) the model uses. Although this parameter isn't a part of the SVM itself it's an important parameter in the models' prepossessing, hence we chose to optimize it first.
The results of this part of the experiment are shown in Table 11 and in figures 4-6. These results show that for every metric increasing the amount of features only has a significant effect until 5000. Thereafter both the score and the standard deviation roughly stay equal or decrease slightly. This combined with the fact that increasing the maximum amount of features seriously burdens the model in terms of computation we decided to continue experimenting with 5000 features.
Finally we optimized the hyper parameters that the SVM has. The SVM has quite a lot of possibilities in terms of optimization but few apply in our case (most require specific preprocessing or data). Therefore we used exhaustive grid search with the parameters that are applicable to our model ("kernel" and "decision function shape") to find the best performing model.
These results are shown in Table 12. The first important thing to note is that the "decision function shape" has no effect on any metric, hence we'll leave that parameter on default (which is "ovr"). The kernel however does have an effect. From these the linear kernel has the best performance so we choose that one to complete our search.
In conclusion, we now have a model with optimized parameters. The settings from this model are summarized in Table 13. All the parameters that aren't mentioned in this table are left on default.

representation	**learning model**	**accuracy**	**f1 score**	**roc_auc**
bag of words	svm	0.87±0.05	0.86±0.06	0.93±0.04
bag of words	linearSVC	0.83±0.05	0.82±0.05	0.89±0.05
bag of words	decisionTree	0.82±0.07	0.82±0.08	0.83±0.06
bag of words	randomForest	0.87±0.07	0.88±0.06	0.93±0.05
bag of words	k-NN	0.79±0.06	0.78±0.07	0.83±0.07
bag of words	SGD	0.87±0.06	0.87±0.06	0.93±0.05
bag of words	naiveBayes	0.76±0.11	0.77±0.09	0.76±0.11
tfidf	svm	0.9±0.04	0.9±0.04	0.94±0.05
tfidf	linearSVC	0.92±0.04	0.92±0.04	0.95±0.04
tfidf	decisionTree	0.79±0.07	0.78±0.08	0.79±0.07
tfidf	randomForest	0.87±0.06	0.87±0.07	0.94±0.05
tfidf	k-NN	0.51±0.03	0.64±0.03	0.54±0.07
tfidf	SGD	0.91±0.05	0.91±0.04	0.95±0.04
tfidf	naiveBayes	0.76±0.09	0.76±0.08	0.76±0.09

Table 2: Results without the removal stop words

representation	**learning model**	**accuracy**	**f1 score**	**roc_auc**
bag of words	svm	0.85±0.04	0.85±0.05	0.93±0.04
bag of words	linearSVC	0.84±0.05	0.84±0.06	0.9±0.05
bag of words	decisionTree	0.78±0.08	0.78±0.08	0.79±0.07
bag of words	randomForest	0.88±0.06	0.89±0.07	0.94±0.05
bag of words	k-NN	0.57±0.14	0.63±0.12	0.58±0.16
bag of words	SGD	0.87±0.07	0.86±0.08	0.93±0.05
bag of words	naiveBayes	0.76±0.11	0.77±0.09	0.76±0.11
tfidf	svm	0.89±0.05	0.89±0.05	0.94±0.04
tfidf	linearSVC	0.91±0.05	0.91±0.05	0.95±0.03
tfidf	decisionTree	0.78±0.07	0.79±0.07	0.8±0.07
tfidf	randomForest	0.88±0.07	0.89±0.06	0.94±0.05
tfidf	k-NN	0.51±0.03	0.64±0.02	0.51±0.05
tfidf	SGD	0.9±0.05	0.9±0.06	0.95±0.04
tfidf	naiveBayes	0.76±0.09	0.76±0.08	0.76±0.1

Table 3: Results with the removal stop words

Train / Test	Nu.nl	NRC	Ondernemer	Telegraaf	Accuracy
Nu.nl	✗	✓	✓	✓	0.84
NRC	✓	✗	✓	✓	0.93
Ondernemer	✓	✓	✗	✓	0.76
Telegraaf	✓	✓	✓	✗	0.85
					0.85±0.06

Table 4: Cross validation with different newspapers using svm, stop words and tf-idf

Train / Test	Nu.nl	NRC	Ondernemer	Telegraaf	Accuracy
Nu.nl	✗	✓	✓	✓	0.84
NRC	✓	✗	✓	✓	0.95
Ondernemer	✓	✓	✗	✓	0.68
Telegraaf	✓	✓	✓	✗	0.84
					0.83±0.1

Table 5: Cross validation with different newspapers using LinearSVC, stop words and tf-idf

Train / Test	Nu.nl	NRC	Ondernemer	Telegraaf	Accuracy
Nu.nl	✗	✓	✓	✓	0.72
NRC	✓	✗	✓	✓	0.75
Ondernemer	✓	✓	✗	✓	0.54
Telegraaf	✓	✓	✓	✗	0.66
					0.67±0.08

Table 6: Cross validation with different newspapers using decision tree, stop words and tf-idf

Train / Test	Nu.nl	NRC	Ondernemer	Telegraaf	Accuracy
Nu.nl	✗	✓	✓	✓	0.83
NRC	✓	✗	✓	✓	0.82
Ondernemer	✓	✓	✗	✓	0.55
Telegraaf	✓	✓	✓	✗	0.84
					0.76±0.12

Table 7: Cross validation with different newspapers using random forest, stop words and tf-idf

Train / Test	Nu.nl	NRC	Ondernemer	Telegraaf	Accuracy
Nu.nl	✗	✓	✓	✓	0.52
NRC	✓	✗	✓	✓	0.52
Ondernemer	✓	✓	✗	✓	0.51
Telegraaf	✓	✓	✓	✗	0.46
					0.5±0.02

Table 8: Cross validation with different newspapers using k-NN, stop words and tf-idf

Train / Test	Nu.nl	NRC	Ondernemer	Telegraaf	Accuracy
Nu.nl	✗	✓	✓	✓	0.84
NRC	✓	✗	✓	✓	0.95
Ondernemer	✓	✓	✗	✓	0.65
Telegraaf	✓	✓	✓	✗	0.83
					0.82±0.11

Table 9: Cross validation with different newspapers using SGD, stop words and tf-idf

Test \ Train	Nu.nl	NRC	Ondernemer	Telegraaf	Accuracy
Nu.nl	✗	✓	✓	✓	0.72
NRC	✓	✗	✓	✓	0.81
Ondernemer	✓	✓	✗	✓	0.56
Telegraaf	✓	✓	✓	✗	0.73
					0.71±0.09

Table 10: Cross validation with different newspapers using naive bayes, stop words and tf-idf

learning model	**representation**	**features**	**accuracy**	**f1 score**	**roc_auc**
svm	tfidf	1	0.5773±0.0621	0.5733±0.0808	0.5772±0.0621
svm	tfidf	5000	0.8999±0.0579	0.8989±0.0583	0.9438±0.0393
svm	tfidf	10000	0.8974±0.0556	0.897±0.0549	0.943±0.0407
svm	tfidf	15000	0.8959±0.0547	0.8956±0.0534	0.9415±0.0415
svm	tfidf	20000	0.8964±0.0539	0.8958±0.0523	0.9411±0.0416
svm	tfidf	25000	0.8969±0.0532	0.8962±0.0515	0.9408±0.0417
svm	tfidf	30000	0.8964±0.0533	0.8955±0.0516	0.9405±0.0417
svm	tfidf	35000	0.8949±0.0531	0.8939±0.0516	0.9403±0.0417
svm	tfidf	40000	0.8949±0.0524	0.8938±0.051	0.9403±0.0417
svm	tfidf	45000	0.8944±0.0521	0.8934±0.0508	0.9403±0.0415
svm	tfidf	50000	0.8934±0.0521	0.8923±0.0508	0.9401±0.0415
svm	tfidf	None	0.8939±0.0503	0.8928±0.0494	0.9401±0.0414

Table 11: The effect of changing the max amount of features.

features	**kernel**	**decision function shape**	**accuracy**	**f1_score**	**roc_auc**
5000	linear	ovo	0.9029±0.0559	0.9003±0.0581	0.9495±0.0341
5000	linear	ovr	0.9029±0.0559	0.9003±0.0581	0.9495±0.0341
5000	poly	ovo	0.8094±0.0774	0.7755±0.1016	0.9167±0.0412
5000	poly	ovr	0.8094±0.0774	0.7755±0.1016	0.9167±0.0412
5000	rbf	ovo	0.8999±0.0579	0.8989±0.0583	0.9438±0.0393
5000	rbf	ovr	0.8999±0.0579	0.8989±0.0583	0.9438±0.0393
5000	sigmoid	ovo	0.9009±0.0563	0.8986±0.0585	0.9498±0.0339
5000	sigmoid	ovr	0.9009±0.0563	0.8986±0.0585	0.9498±0.0339

Table 12: The effect of tweaking the parameters with svm

learning model	**features**	**text representation**	**kernel**	**max no. of iterations**	**accuracy**
svm	5000	tf-idf	linear	5000	0.9029±0.0559

Table 13: Settings from the final model. Parameters that aren't mentioned are left on default.

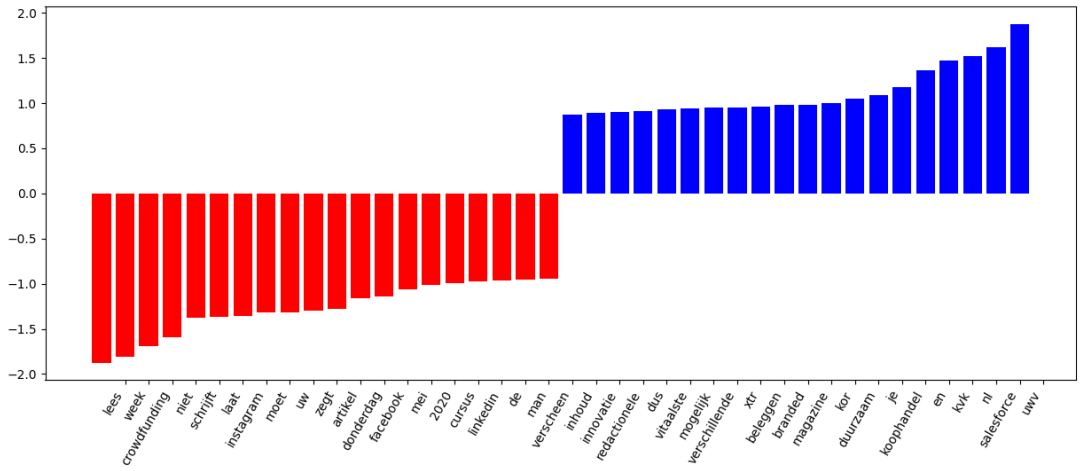


Figure 2: Top features without the removal of leaking variables or stop words

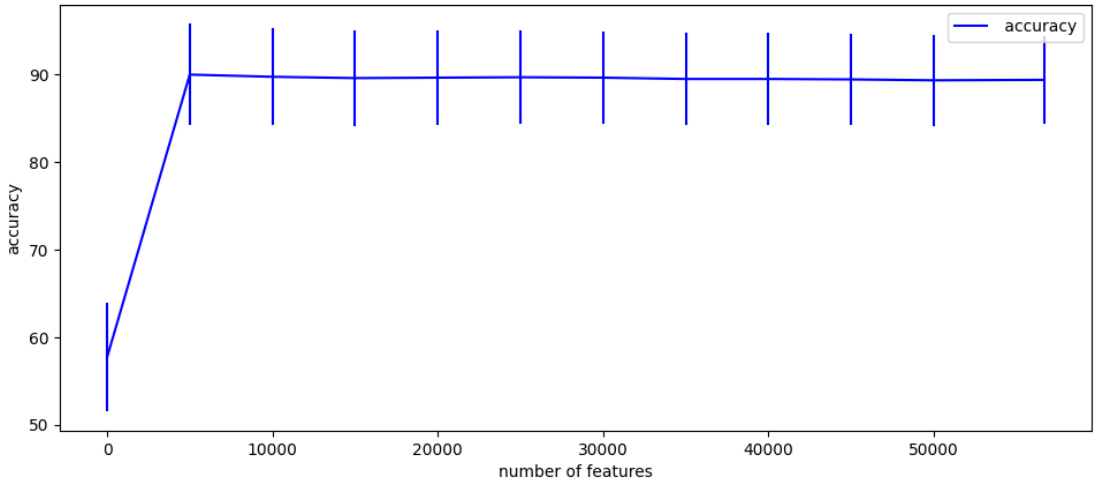


Figure 3: The effect of changing the amount of features in terms of accuracy.

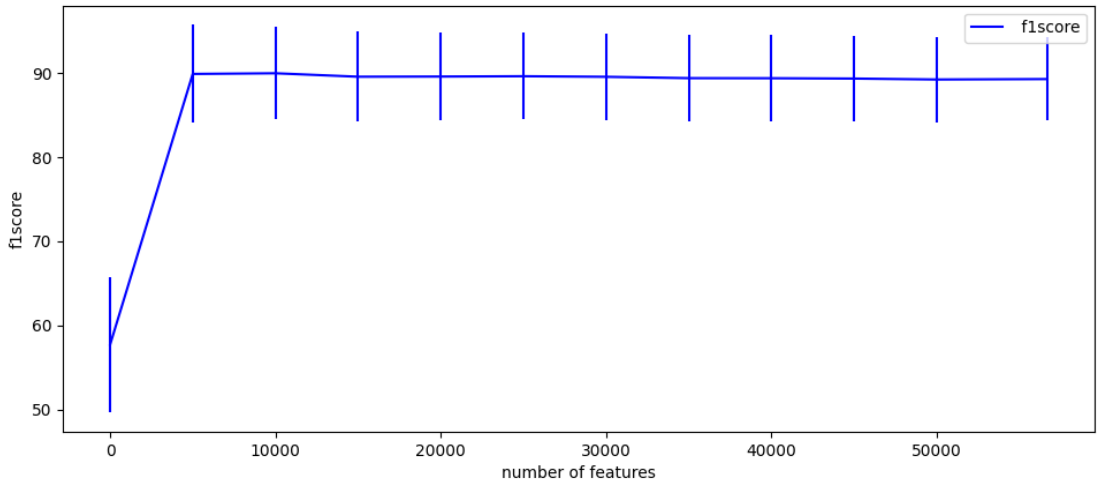


Figure 4: The effect of changing the amount of features in terms of the f1 score.

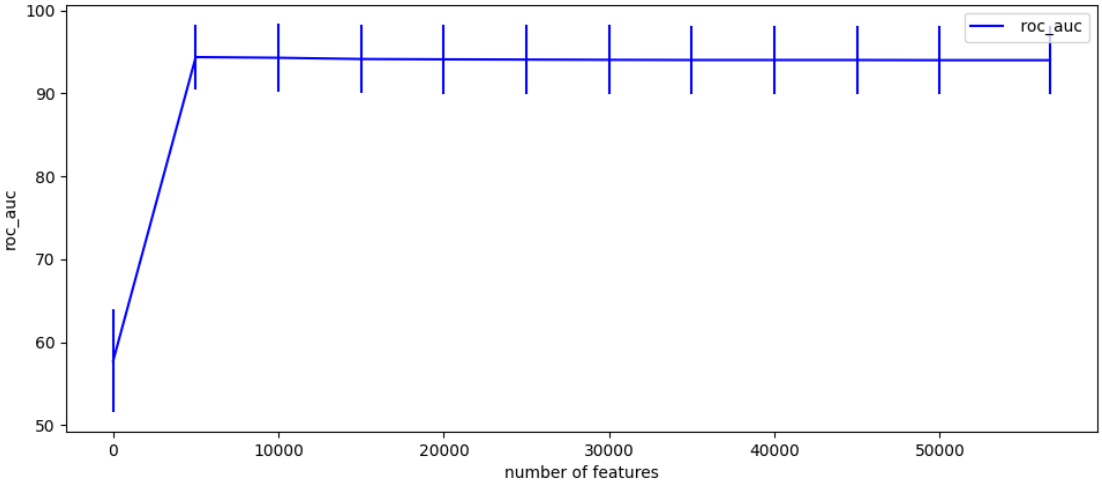


Figure 5: The effect of changing the amount of features in terms of roc auc.

5.2.2 Deriving the lexicon

The lexicon adds something to our research (even though it serves the same purpose as our model) because it can be published without needing to publish the data as well. This isn't the case for our model since it requires our training and testing data to be developed, which we can't publish due to legislation around intellectual property. This is because our complete data set contains journalistic content that officially belongs to news sources instead of the public domain. For example, NRC has a statement on their website that states that all of their journalistic products are copyrighted [16]. Thus, we could only publish an anonymized data set without the content of the articles, making a lexicon a good way to still publish our results.
Based on the settings of the machine learning model we configured through the experiments mentioned earlier (see Table 13) we made our lexicon. Because we wanted to base the lexicon on as much data as possible we trained it on the entire data set. Next, since our model has 5000 features our lexicon also automatically consists of 5000 terms and scores. These are equal to the feature terms and weights the SVM has calculated for it (see Figure 6). An overview of the distribution of all the scores in the lexicon can be found in Figure 7.
What the scores in effect represent is how editorial or commercial a word is according to our model. So for finding out how commercial or editorial individual words are this lexicon alone is already sufficient. However, it's also possible to use this lexicon to classify larger bodies of text.
To do this we developed a simple algorithm/formula (see formula below) that bases itself on the tf-idf or bag of words representation of the text. The vocabulary in this case consists of the 5000 words in the lexicon, so every piece of text is represented as a 5000 feature long array.
Thereafter, all features in this array are multiplied by their score (denoted by l in the formula) in the lexicon. The sum of all these scores result in a total score for the entire text. If this score is greater than 0, the text is classified as an advertorial, else it's an article.

$$score = \sum_{word}^{text} word \cdot l$$

We tested the performance of this formula on our data set by making it solve the same classification problem our machine learning model solved (to reiterate; predicting whether an entry is an advertorial or an article based on its text). This resulted in a 98% train-set accuracy when using a bag of words representation and a 98.5% train-set accuracy when using a tf-idf representation.
Note, even though these scores are very high it must be taken into account that the lexicon was tested on the same data as it was trained on, whilst our model was evaluated on a 80/20 train-test split. Hence, the performance of the model can't be compared with the lexicon based on this experiment.
Finally, since every entry has a score, we can't only see what entries are being classified as, but also with what certainty. Hence all the scores, differentiated per news source, are summarized in a histogram in Figure 8.

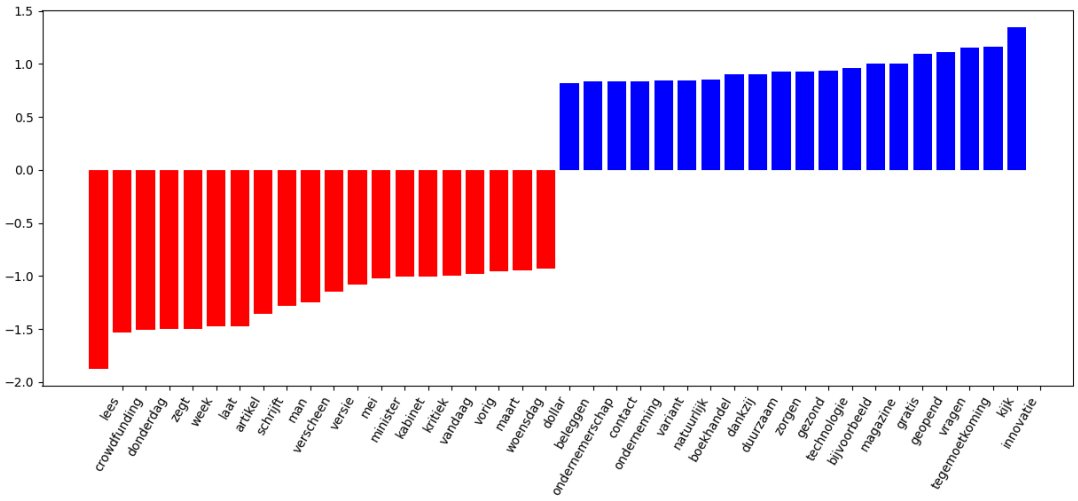


Figure 6: Top features of the final model.

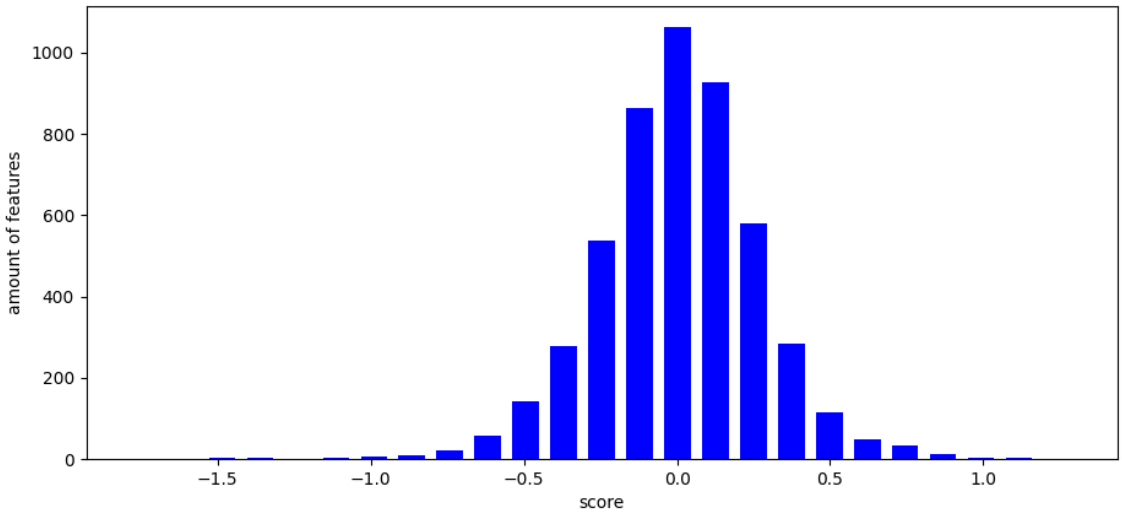
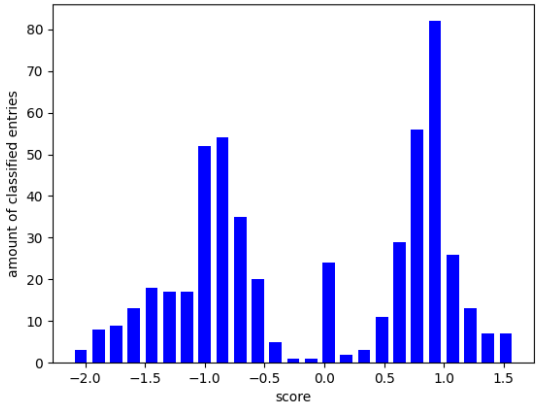
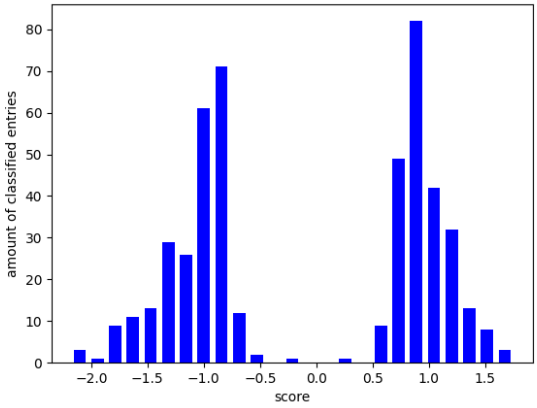


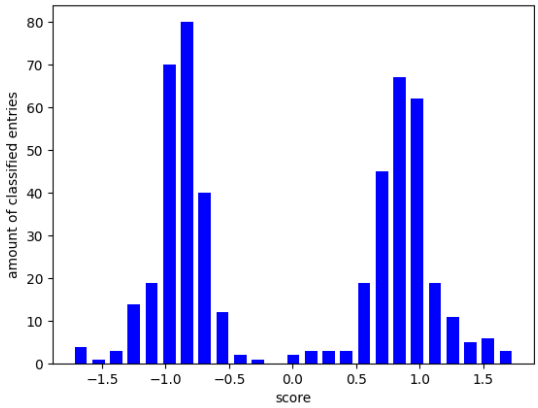
Figure 7: The distribution of the scores from the features in the lexicon.



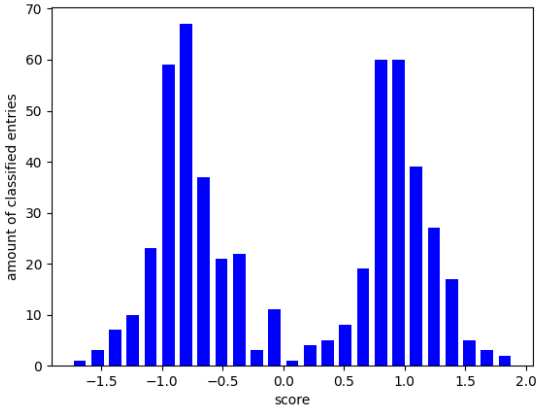
(a) Nu.nl



(b) NRC



(c) De Telegraaf



(d) De Ondernemer

Figure 8: The distribution of the predicted scores from the entries using the lexicon per newspaper.

5.2.3 Exploring the data and results through visualization

To get a better insight into the data we used and how it relates to our results we explored it through visualization. This visualization was made with an association graph based on the text and the terms from the lexicon.
We calculated these associations based on how often the terms from the lexicon appear in the same sentence, divided by the terms' total occurrences (in sentences) for the total data set. For example, in our data set every time the word "artificial" appears in a sentence, 76.47% of the time that sentence also has the word "intelligence" in it. Hence the weight of the edge between these two nodes would be 76.47%.
Using this method we created a graph where the nodes are the terms from the lexicon and the edges are the association between them. The graph is thereafter filtered with a minimum 75% cutoff percentage, only keeping the strongest associations. Furthermore, if two terms have a different percentage towards each other we take the smallest percentage as the association value, not the average.
Since the graph based on the entire lexicon is too large and complex to display on print we only the show parts of the graph (subgraphs) that are applicable to this research in the discussion section of this thesis. The complete graph can be found at: `https://timokats.github.io/network/`.
Finally, this graph is structured using Yifan Hu Proportional [17], which is a force-directed graph drawing algorithm that structures the graph in a way that allows us to easily find important relations, nodes and clusters.

5.2.4 Exploring the models' behavior through T-sne graphs

Just like our data and lexicon, we can explore the behavior from our model as well. In our research we did this through differentiating the model per newspaper and visualizing it's behavior with T-sne graphs. T-sne graphs [18] are a way to represent multi-dimensional models (in our case a 5000-dimensional model) in a two-dimensional graph. Thus, this method can visualize the separate results of the model (the predicted articles and advertorials) in relation to each other.
Our T-sne graphs can be found in Figure 10, along with their accuracy and standard deviation. Every graph here shows the entries classified as an article or an advertorial in a two-dimensional scatter plot. The points in the scatter plot represent the classified entries whilst the axes are the T-sne's representation of that 5000-dimensional point in a two-dimensional form.
Using these graphs we can see the differences between the newspapers and how it's points are classified. Furthermore, since these points are trained and tested within the same newspaper, so we can also get an overview of how the models' behavior changes per newspaper.

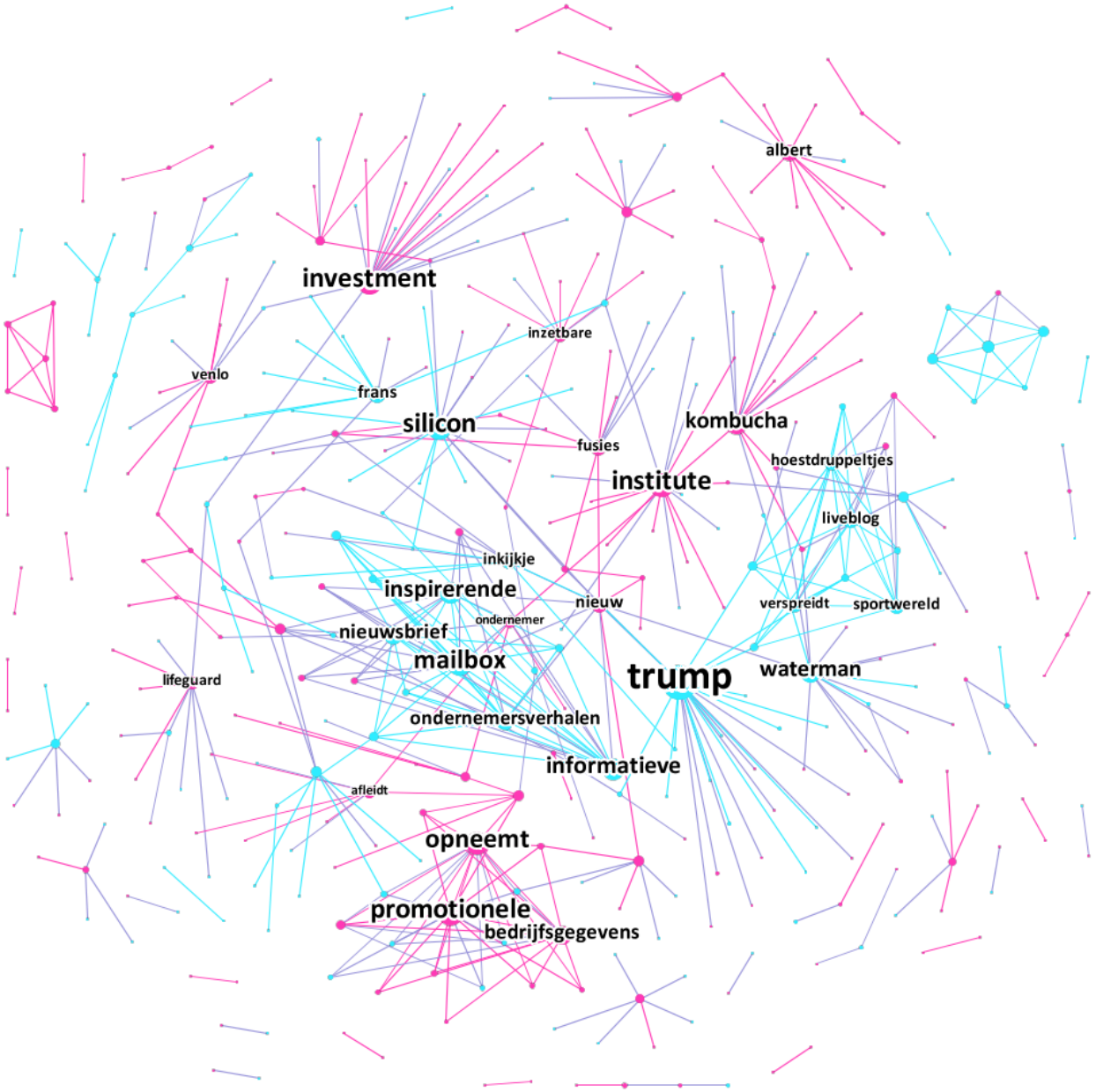
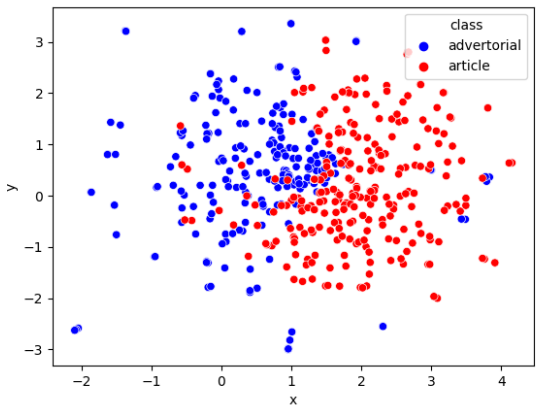
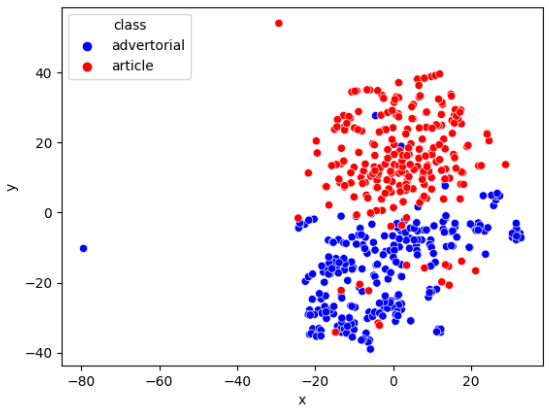


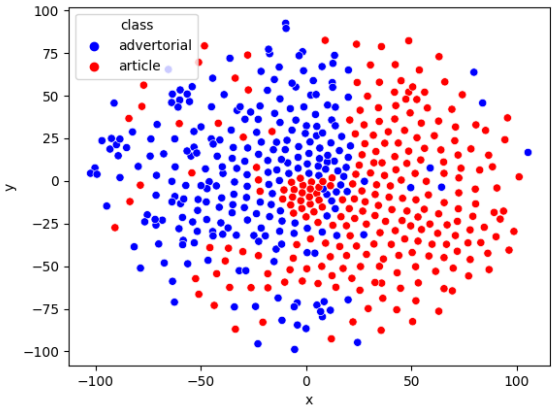
Figure 9: Overview of the total graph. Complete version can be found at: timokats.github.io/network/



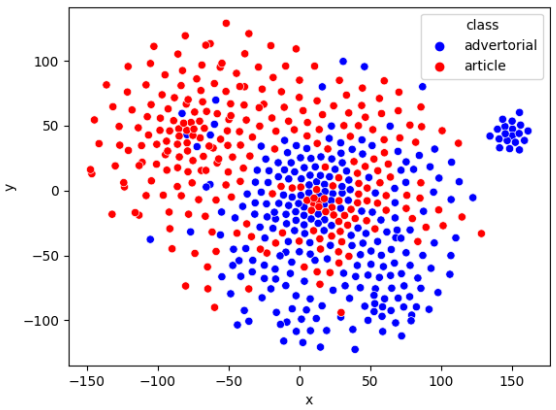
(a) Nu.nl (0.9219±0.0464)



(b) NRC (0.9518±0.0656)



(c) De Telegraaf (0.9109±0.0395)



(d) De Ondernemer (0.8544±0.0633)

Figure 10: T-sne scatterplots on the SVM trained and tested within the same newspaper

6 Discussion

In this section we discuss the results from our experiments and how they fit in a broader setting. Firstly we'll discuss the limitations our experiment has. Next we interpret the results and what they imply, and finally we'll give our recommendations for further research in this subject.

6.1 Limitations

Our research and the way it's set up has some limitations. In this subsection we discuss what these limitations are and what their effect is.

6.1.1 Text representation

In our experiment we used tf-idf and bag of words to convert the text to numbers. Those methods, although effective, have some limitations.
Firstly they can only convert text that's explicitly written. Anything implicit that readers can still catch onto these methods will probably miss. Hence the complete meaning of the text might get lost a bit using these methods.
Secondly, for both methods the order of words is not being taken into account. This means that texts with a completely different meaning but the same words (for example "This is very bad. Not good!" and "This is not bad. Very good!") can be translated to the exact same array of numbers.

6.1.2 False positives due to topic bias

In our lexicon we can observe the words that our model sees as commercial and editorial. A lot of the words that lean commercial center around the topics and buzzwords that advertisers use to sell their products. Examples of this would be words like "innovatie" (innovation), "technologie" (technology) and "duurzaam" (sustainable).
However, although these words might be common in advertorials, they're not exclusive to advertorials. In fact, it's totally possible for an article to be about "innovatie" (innovation) or "technologie" (technology) without having any commercial incentives. Hence, there is a topic bias in our model. We observed this bias in Tables 14 and 15. These tables each show ten different false positives and false negatives from our model. To reiterate, false positives in our model refer to articles that are classified as advertorials whilst false negatives are advertorials classified as articles.
Firstly, when looking at the false positives (see Table 14) it's clear that a couple of themes stand out. Articles about new and innovative technology ("Windmolens voor kunstmest" from NRC and "Winnaar Privacy Award: persoonlijke data veilig bij gadgets Tijmen Schep" from Ondernemer) in particular can be a result of our topic bias. But also more literal examples such as "Eerste details van nieuwe elektrische SKODA bekendgemaakt" from Nu.nl shows that articles about products (that might've also been marketed in advertorials) tend to wrongly get classified as an advertorial.
Next, when looking at the false negatives (see Table 15) the topic bias is less clear. What is clear however is that a lot of the titles have leaking variables in them. To reiterate, these would be brand names and disclaimers that are unique to our data set. Removing these words from the next might be the reason these advertorials have been wrongly classified as an article. Nevertheless, this is not the same as a topic bias.

6.1.3 Time frame bias

Most of our advertorials and articles are from 2020 (see Figure 1). Because 2020 was quite an atypical year, the text from our data set differs from other years. Subjects like for example covid-19 and the US elections are more common when sampling articles that are largely from 2020 then any other year in the past, and probably also in the future.
Moreover, the uniquely high occurrence of these subjects influence the model and the lexicon. Examples of this would be the fact that the lexicon has ten terms that directly refer to the covid-19 pandamic as well as the US 2020 election with terms like "Biden" and "Trump". This creates a problem, because these terms might not be as applicable in future articles as they were in 2020. Hence our model has a time frame bias towards the subjects that were common in 2020.

6.2 Interpretations

After getting our results we can interpret them. Since this research has multiple results and visualizations we also have multiple things to do this for. Firstly we interpret our model and lexicon. Secondly we interpret our visualizations and finally we interpret our T-sne graphs.

6.2.1 Interpreting the model and lexicon

For this research we have two different results that are closely linked together. The first result is the performance from our best performing model and our second result is the performance from the lexicon that is derived from that model. Both of these are based on the same data set, which was acquired specifically for this research and therefore has not been used by other researchers. This makes it hard to contextualize our results with that of previous work. Moreover, most of the research that has been done in the field of advertorials came from a marketing and psychology perspective instead of a computer science perspective.
However, there are results on how well readers can differentiate articles and advertorials from research conducted by the University of Georgia [5]. To reiterate, this research found that only 8% of participants could differentiate editorial from sponsored content. This result is miles apart from our results, since our best performing machine learning model had a test accuracy of just over 90% (see Table 12) and our lexicon had a train accuracy of just over 98%.
Regardless, our results don't conflict with that or even the general notion that advertorials and articles are hard to differentiate for readers. This is because our results purely focus on the ability of machine learning being able to differentiate commercial and editorial content.

6.2.2 Interpreting the visualization of our data and results

The data from this research was explored through visualizing it with an association graph. To reiterate, the parts from the graph that are applicable to this research are highlighted in the appendix whilst the complete graph can be found at: `https://timokats.github.io/network/`. A partially-labeled overview of this graph can be found in Figure 1. This is because the complete graph is too large and complex (in total 413 nodes) to accurately display on print.
When looking at the graph there are a couple of things we can learn from it. Firstly the separation of editorial and commercial nodes/clusters can show where commercial and editorial language differs and where it overlaps. Secondly the labels from the nodes (especially the ones with a high

degree) can inform us about what topics our lexicon has and how they relate to each other. Finally, individual connections and smaller clusters can show what phrases and terms are common in our data set and what the model has attached itself to.
Firstly, the clusters. The largest editorial cluster's central node is "Trump", which is attached to two sub graphs. One of these sub graphs is a cluster of covid related terms and the other sub graph is more related to more (timeless) political terms (see Figure 11). The fact that "Trump" is such an important node whilst also being attached to a number of covid related terms relates back to our time frame bias for editorial terms mentioned earlier, since these terms are typical of the news in 2020.
Finding commercial clusters however is less clear (see Figure 12). There are a couple of semi large clusters that are linked together through the nodes "nieuw" and "nieuwe". These are both very common commercial words that are applicable to almost all goods or services. In fact, "nieuw" has 2814 occurrences (in sentences) and "nieuwe" has 1628 occurrences (in sentences). In our graph this is a lot. The type of terms that they link to are mostly related to products (like "kombucha") or business operations (like "fusies"). Hence, unlike editorial terms there aren't any clear categories or biases we can observe, which could be a result of filtering the leaking variables successfully.
Secondly, the nodes. The editorial node with the highest degree is "Trump". In fact, there are no nodes in the entire graph that have a higher degree than "Trump". It's huge, and again, just like the editorial cluster it's attached to, it shows the time frame bias our data has. For commercial nodes however the node degrees are more even. Some examples of commercial nodes that have a higher degree are "investment", "promotionele" and "kombucha". In conclusion, similar to the clusters, there aren't any real outliers or patterns we can observe on the commercial side.
Finally, some associations and smaller clusters in the graph that showcase important parts and associations in our data. When looking at the total graph there are a lot of associations and clusters that add value to our exploration. That's why in this thesis we will only highlight a couple just to give some examples. Firstly, for commercial nodes (middle left of the graph) there is a cluster where almost all the nodes refer to travel and tourism (see Figure 13). When examining these nodes it's clear that they come from all newspapers, so they're not linked just because they're from one advertorial. Hence, based on our graph we can conclude that tourism is a subcategory of commercial content in our data set.
Secondly, for editorial terms we can also find a cluster of similar size where all the terms refer to covid symptoms (see Figure 14). This reconfirmed the time frame bias talked about earlier both with "Trump" being a major node and general covid related terms being common in the data set according to our graph.

6.2.3 Interpreting the models' behavior through T-sne graphs

As mentioned in our results, we explored the predictions from our model with T-sne graphs (see Figure 10) and our lexicon with histograms (see Figure 8). Because both of these are segmented per newspaper and display the predicted entries they're complementary to each other. That's why we can use them both apart, and together, to explore and interpret our results.
Firstly, the T-sne graphs from the model. These scatterplots show the predicted data points from our model in a two-dimensional form, of which the blue dots refer to advertorials and the red dots refer to articles. What's fascinating to see is how different the shapes of the scatterplots are per news source and how that correlates to the results.

The clearest shape (in terms of the divide there is between editorial and advertorial points) is from NRC, which also automatically has the highest accuracy. The lowest score and most opaque graph is from Ondernemer. Which also has a cluster of advertorials completely outside the main cluster of the graph.
This pattern continues in the histograms from Figure 8. NRC shows a very clear divide with almost no ambiguous scores whilst the other newspapers (in particular Ondernemer again) paint a much more opaque picture.
In conclusion, what both figures show is that there are major differences per news source in our data set when looking at the results. It's therefore likely that the news sources in our data set also differ from the news sources that we've not included in our research. That's why our data set shouldn't be interpreted as a complete and holistic representation of the dutch media landscape and nor should our results.

6.2.4 Interpreting the results with the separation of church and state

All our (preliminary) results center around differentiating commercial and editorial content in the form of advertorials and articles. As mentioned, in journalism the divide between commercial and editorial content is also referred to as their separation of church and state [1]. Thus, due to this overlap we can use our results as an indicator for how the separation of church and state differs per news source in our data set. In fact, when designing our model we already found that in our preliminary results the performance differed per news source (see Tables 4-10). Please note that we don't make any claims regarding the intent that these news sources write with, we only interpret their performance based on our data and results.
Firstly, the T-sne graphs (see Figure 10). As mentioned, these scatterplots show the predicted data points from our model in a two-dimensional form, of which the blue dots refer to advertorials and the red dots refer to articles. Furthermore, this figure also shows the performance of the model when it's trained and tested within the same newspaper. So what these graphs effectively show is how successful our model is at differentiating advertorials and articles for every news source, which also can be interpreted as a proxy for their separation of church and state.
Given this interpretation it's apparent that NRC (with the highest score and clearest T-sne graph) has the best separation of church and state, since it's easier for our model to differentiate NRC's advertorials and articles. The performance of Nu.nl and Telegraaf is quite identical. The performance of Ondernemer however is significantly lower than the other newspapers, since it has the most opaque graph and lowest score.
Moreover, we see the same pattern when we look at the performance of our lexicon (see Figure 8). This histogram shows the predicted scores of the lexicon per news source. How this can be interpreted with regards to church and state is that if a predicted score lies closer to zero, it's more ambiguous to our lexicon. In other words, if there are a lot of predicted entries with values close to zero the separation of church and state for that news source is less clear. Given this interpretation we again see that NRC has the clearest divide and Ondernemer the least clear (whilst Nu.nl and Telegraaf are quite similar).
In conclusion, both figures point to the fact that in our experiment NRC has the best separation of church and state and Ondernemer the worst, with Nu.nl and Telegraaf being average and quite similar to each other. Note, we don't claim that this is also an effective representation of how the separation church and state is respected in the news sources' writing.

source	title
ondernemer	Waarom je je niet te veel op de uitvoering van je plan moet richten (en waar dan wel op)
telegraaf	Jorritsma tegen private equity: gedraag je netjes in coronatijd
ondernemer	Alles wat je wil weten over PR
ondernemer	Denktank Coronacrisis: 'Werkgevers vergeet kwetsbare mensen juist nu niet'
ondernemer	Dutch Nomad Couple: de Triple Double van de travelmarketing
ondernemer	Merendeel van het mkb maakt nog geen gebruik van crm-systeem
nrc	Windmolens voor kunstmest
nu.nl	Eerste details van nieuwe elektrische SKODA bekendgemaakt
ondernemer	Winnaar Privacy Award: persoonlijke data veilig bij gadgets Tijmen Schep
nu.nl	Maak je balkon of terras zomerklaar

Table 14: Ten different false positives with our model.

source	title
nu.nl	Kijk elke week een nieuwe film tijdens de Pathé Thuis Filmzomer
telegraaf	Vrouw die superlijm als haarspray gebruikte wil fabrikant aanklagen
ondernemer	Succesvol ondernemen volgens Yoni? Heb lef maar vooral: sta achter je missie
nu.nl	Zo werden de verwaarloosde honden David en Goliath geholpen
ondernemer	Noodloket voor werkgevers open: UWV verwacht tienduizenden aanmeldingen per dag
nu.nl	Groninger Krant verovert nieuwslandschap in Groningen
ondernemer	Zo hielp slimme technologie Picnic de crisisdrukte door
telegraaf	Voorkom terugbetalen: controleer de voorlopige aanslag
ondernemer	Zo ervaart de directeur van Down the Rabbit Hole een festivalloze zomer
nu.nl	Tour Wielerspel: is jouw team klaar voor de eerste etappe?

Table 15: Ten different false negatives with our model.

source	Disclaimer position(s)	Relative size (compared to each other)	Different font?
nu.nl	Right and Top	Smallest	False
telegraaf	Top	Medium	False
ondernemer	Top	Medium	True
NRC	Left	Largest	True

Table 16: Outline of the disclaimers for all news sources in our data set

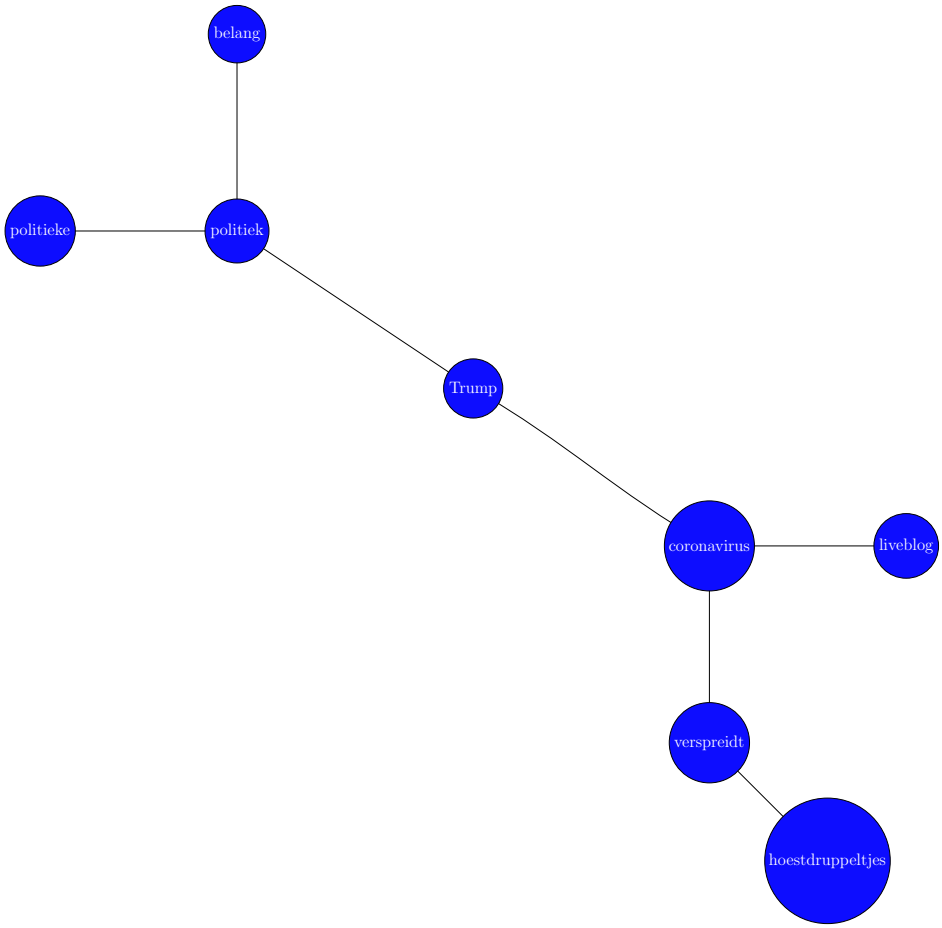


Figure 11: Important connections from the main editorial cluster

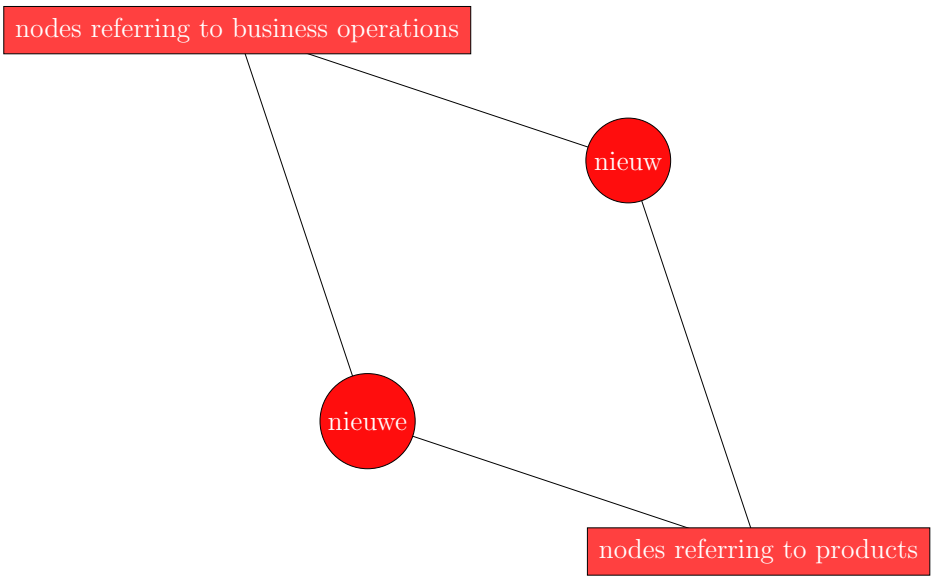


Figure 12: Important connections from the main commercial cluster

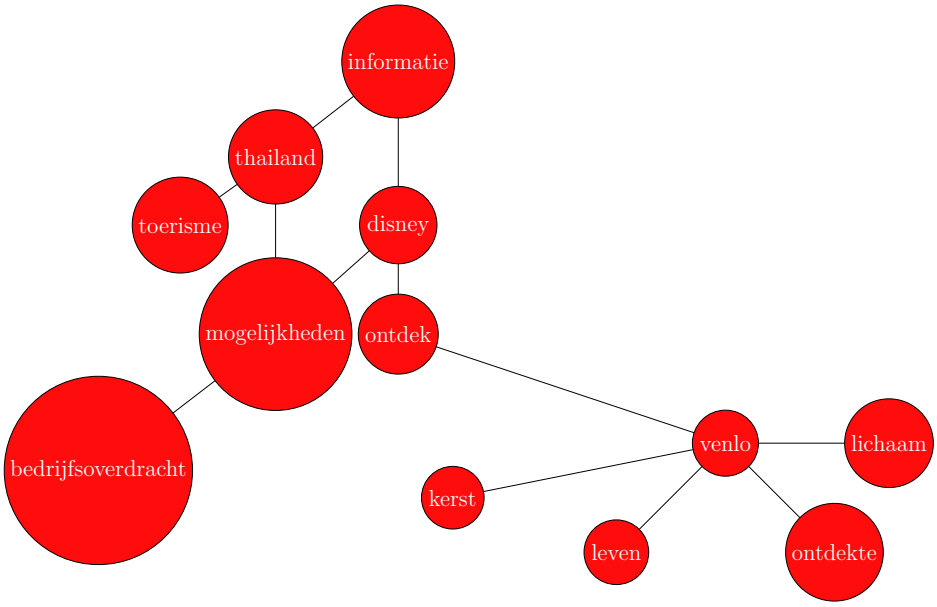


Figure 13: Commercial cluster that centers around tourism

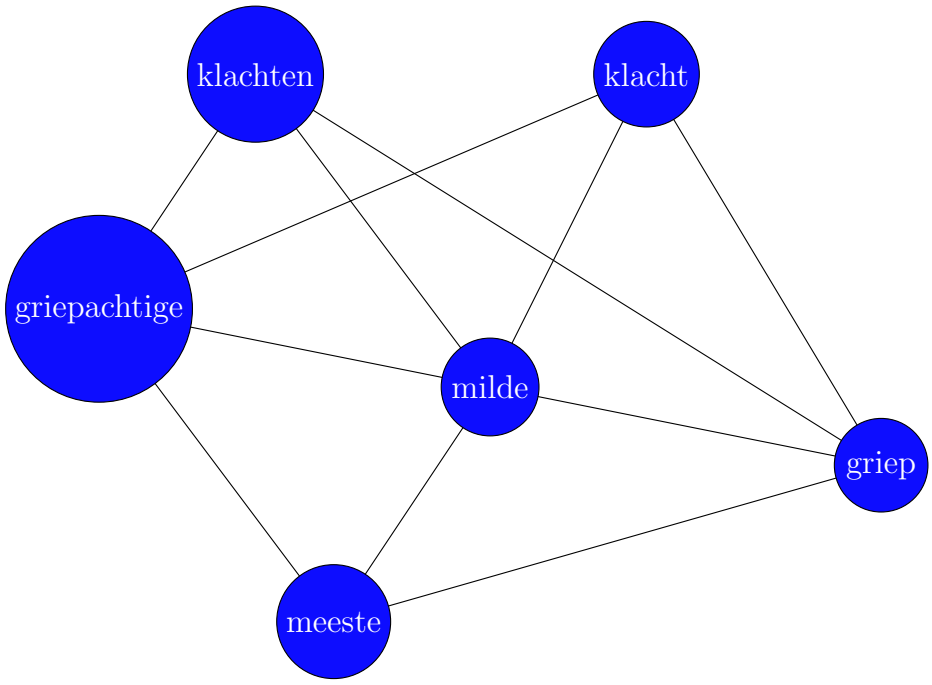


Figure 14: Editorial cluster that centers around covid symptoms

6.3 Further research and recommendations

For those that want to continue doing research in this field we have some suggestions and recommendations. These range from simple additions like a stemmer to more fundamental changes like the size of the data set or sentiment analysis.

6.3.1 Adding a stemmer

The data we used for our machine learning model didn't have any preprocessing besides from making all words lower case. This means that terms that (roughly) mean the same thing (for example "feature" and "featuring") are being read as unique and separate words. This can be avoided through using a stemmer, which is a form of preprocessing that converts words back to they're etymon. This would make words that mean the same also be the same, which is helpful for two reasons.
Firstly because there are multiple terms in the model that mean the same thing and therefore don't add much. In our lexicon the words "innovatie" (innovation), "innovatief" (innovative) and "innoveren" (to innovate) are all different terms and they all fill a spot in the lexicon when actually they refer to the same thing. When using a stemmer all these terms would be converted to their etymon (for example "innovation") and only one spot of the lexicon would have to be used.
Secondly it burdens the model with excess baggage. Just like stop words, small variations of terms that are already being used in the model don't add much. Adding them to the data regardless therefore burden the model unnecessarily, especially since machine learning is quite a computationally demanding task. Adding a stemmer would help to prevent this since converting multiple words to their etymon simplifies the input.

6.3.2 Increase the size of the data set

Our data set has a total of 1000 articles and 1000 advertorials (see Figure 1) from four different news sources. For data science and machine learning standards this is quite small. The reason we couldn't add more entries to our data set is because advertorials tend to get deleted after being online for a while, making them very hard to acquire. Moreover, since we don't want to incite the model to predict entries as articles just because they're the largest group in the data set we matched the amount of articles to the amount of advertorials. Limiting them to just 1000 entries total.
That's why we recommend future work to try to collect more data than we did. Moreover, we also recommend future work to actively acquire articles and advertorials from multiple years with comparable proportions (i.e. roughly the same amount of articles and advertorials per year).
This is because acquiring articles and advertorials predominantly from a single year increases the time frame bias, which we explained earlier. Furthermore, if the articles and advertorials are largely from different years there's a chance that a disconnect between them could occur in terms of their topics and vocabulary. This would then result in also partially classifying them based on "older" and "newer" content instead of just commercial and editorial content.

6.3.3 Sentiment analysis

Sentiment analysis is the process of classifying text as positive, negative or neutral. This fits into our research because we also classified text into different categories; commercial and editorial.
However, what future research can also examine is how the sentiment of these two categories differs. Meaning, whether commercial text is generally written positively or negatively than editorial text.

6.3.4 The effect of disclosing advertorials on readers

Our model and lexicon have shown to be able to differentiate advertorials and articles more successfully than readers have done in related work [5]. That's why we recommend future work to research if this model (or a model like this) can be used to assist readers in recognizing commercial content in the real world.
Furthermore, what future work could also examine is whether our model classifying certain content as commercial also leads to a more critical assessment of what is read (by the reader).

6.3.5 Research the effect of different disclaimers with machine learning

Related work has shown that the placement and formatting of disclaimers has a huge effect on the readers' ability to differentiate commercial and editorial content in the form of advertorials and articles [10]. Since our data only contained textual content we haven't explored this aspect of recognizing advertorials in our research, even though our news sources have vastly different ways of disclaiming their advertorials (see Table 16). That's why we recommend future work to implement these sorts of variables in the input of their machine learning models when researching advertorials.

7 Conclusions

This thesis aimed to differentiate commercial and editorial language based on two research questions. To reiterate, these questions were: To what extent can we differentiate advertorials and articles by using machine learning? And: To what extent can we derive a lexicon from our machine learning model that differentiates commercial and editorial language?
Both of these questions were answered with the results of our model and our lexicon. Based on the fact that our optimized model scored an accuracy of 90% and the lexicon derived from that model scored an accuracy of 98% (when using tf-idf representation) we can conclude that our model as well as our lexicon successfully differentiate commercial and editorial language.
However, it must be noted that our research has some limitations. Our lexicon was tested on the same data the model was trained on. This means that it has not yet been tested on unseen data. Furthermore, our model was trained on a data set of just 2000 entries, which is quite small for data science and machine learning standards.
Nevertheless, considering the fact that previous work has shown readers only recognize 8% of commercial content [5] we can still be very pleased with our results. Thus, our research still illustrates the potential of machine learning for differentiating commercial and editorial content, but has not quite reached it yet.
To accomplish that, we have a couple of recommendations that future studies could use. Firstly to

increase the size of the data set and train, test and tweak on separate parts of that data set. Also, adding a stemmer can help creating a more efficient and generally applicable model.
In conclusion, this research has successfully shown the potential of machine learning when differentiating commercial and editorial content. Moreover, it also showcases how machine learning can be a solution, not a problem, in the modern media landscape.

References

[1] Raul Ferrer Conill. Camouflaging church as state: A study of journalism's native advertising. 09 2015.

[2] Mark Thompson. The New York Times Company 2019 Annual Report. Technical report, The New York Times, 2020.

[3] Enders Analysis. Native advertising in europe to 2020. 02 2016.

[4] Bianca Harms, Tammo Bijmolt, and Janny Hoekstra. Digital native advertising: practitioner perspectives and a research agenda. *Journal of Interactive Advertising*, pages 00–00, 08 2017.

[5] Bartosz Wojdynski and Nathaniel Evans. Going native: Effects of disclosure position and language on the recognition and evaluation of online native advertising. *Journal of Advertising*, pages 1–12, 03 2016.

[6] Sacha Wunsch-Vincent and Graham Vickery. The evolution of the news and the internet. Technical report, The Organisation for Economic Co-operation and Development, 06 2010.

[7] Mijke Slot Andra Leurdijk and Ottilie Nieuwenhuis. The newspaper publishing industry. Technical report, The European Commission, 2012.

[8] Marc Edge. Newspapers' annual reports show chains profitable. *Newspaper Research Journal*, 35:66–82, 09 2014.

[9] Bikramjit Rishi, Aditya Mehta, Poulomi Banerjee, and Akshay Deepak. Buzzfeed inc: native advertising the way forward? *Emerald Emerging Markets Case Studies*, 8:1–18, 09 2018.

[10] Simone Krouwer, Karolien Poels, and Steve Paulussen. To disguise or to disclose? the influence of disclosure recognition and brand presence on readers' responses toward native advertisements in online news media. *Journal of Interactive Advertising*, 17:00–00, 10 2017.

[11] Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. Lexicon-based methods for sentiment analysis. *Computational Linguistics*, 37:267–307, 06 2011.

[12] Reverb Channel. Reverb channel. Available at: `https://www.aced.site/en/programmes/reverb-channel`.

[13] Leonard Richardson. Beautiful soup, 2004. Available at: https://www.crummy.com/software/BeautifulSoup/bs4/doc/.

[14] F. Yergeau. Utf-8, a transformation format of unicode and iso 10646, 1996. Available at: https://www.hjp.at/doc/rfc/rfc2044.html.

[15] Gene Diaz. Stopwords dutch (nl), 2016. Available at: https://github.com/stopwords-iso/stopwords-nl.

[16] NRC. Auteursrecht- en databankenrecht voorbehoud. Available at: https://www.nrc.nl/auteursrecht-databankrecht/.

[17] Yifan Hu. Efficient and high quality force-directed graph drawing. *Mathematica Journal*, 10:37–71, 01 2005.

[18] Laurens van der Maaten and Geoffrey Hinton. Viualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 11 2008.