Sentiment Analysis and Topic Classification of Newspaper Articles

Shayna Grose
sgrose@sfu.ca

Anmol Saini
ahsaini@sfu.ca

Patrick Nguyen
pna18@sfu.ca

Argenis Chang
kchangch@sfu.ca

Abstract

We will explore the task of sentiment analysis and topic classification of newspaper articles. For sentiment analysis we propose a lexicon based approach which compares the words in newspaper articles to a corpus of positive and negative words. We will experiment with comparing just the raw count of positive versus negative words in an articles contents, as well as their headlines. We will also experiment with using TF-IDF to process article data before performing sentiment analysis. For topic classification, we propose using non-negative matrix factorization (NMF) and support vector classifier (SVC) to classify articles into one of five topics: business, entertainment, politics, sports, and technology. We show that our model can correctly classify articles into their respective topics based on their lexical contents. These results are shown in a map-based application that allows users to filter according to article categories and sentiment.

1 Introduction

In our digital age, most news media is consumed online. With the wealth of information online, newspaper outlets have turned to sensationalizing and polarizing news articles headlines to grab readers attention. We would like to classify articles into positive and negative categories based on just their headlines, or just their contents, or both. This will allow us to see if the number of articles which are put into different categories based on their headline or contents are significant, which may show that headlines do not accurately represent an articles contents. As a complimentary task to sentiment analysis, we will also perform topic classification of the news articles, to see if there are any interesting trends related to the sentiment and topic of an article. An application for this would be potentially allowing users to filter the news they read based on the topic of an article.

So far in our analysis we have found that after running our model on the article data sets we cleaned, the opinion classification of the articles contents were more useful than the classification of article titles, which primarily resulted in Neutral as the final output for most article titles. The sentiment analysis of the article text was further split according to the news categories to show whether a specific topic is more inclined to be positive or negative. While business and political articles leaned more towards negative classification, entertainment, sports, and technology had more positively classified articles, as can be seen in Figure 5. We also repeated this experiment using TF-IDF and compared the results with just using raw counts, which is explored in more detail later.

For topic classification we observed all of the categories had over 80% of their articles correctly classified, with the exception of the technology category which had 40% of it's articles correctly classified. We think this may be because there is overlap between the technology, sports and entertainment categories.

2 Related Work

Some inspiration for our idea was drawn from a recent paper on Sentiment Analysis of News Articles (Taj et al., 2019). This paper explores two methods of sentiment analysis, the first being a lexicon based approach, and the second based on approaches of machine learning. They do sentiment analysis at a document level, using the wordNet lexical dictionary.

Another interesting paper on the topic of sentiment analysis titled ”Sentiment Analysis and Subjectivity” explores the problems of sentiment analysis, where the appearance of an opinion word in a sentence does not necessarily mean that a sentence expresses a positive or negative emotion (Liu, 2010). Due to our limited resources and expertise, we did not do sentiment analysis on a sentence by sentence basis, but rather on entire newspaper articles. Given more time, we may have expanded our approach to consider the sentiment of individual sentences, and classify an article as positive or negative, given if it had more positive or negative sentences.

3 Approach

For sentiment analysis, our primary approach to identifying whether an article is positive or negative will be comparing articles to existing negative and positive words corpora. We will analyze the ratio of negative words to positive words of the article title and the article's contents. The two corpus's of positive and negative words we are using are taken from a GitHub repository (Hu and Liu, 2004). The second approach was to use term frequency–inverse document frequency (TF-IDF) to weight words.

Furthermore, we will be classifying articles according to their topics by the use of Non-Negative Matrix Factorization (NMF) and Support Vector Classifier (SVC). The non-negative matrix factorization algorithm will be used for feature extraction to decrease the dimensionality of the articles we will take as input. The resulting matrix will then be used as input to the support vector machine to classify the news pieces into one of the following classes: business, entertainment, politics, sports, or technology. With a set of 2225 articles, the NMF algorithm will extract relevant words to reduce the files dimensionality and efficiently classify article topics. To prove the accuracy of our topic classifier, we will input articles with known categories and check if our algorithm will deliver the same classification.

4 Experiments & Results

4.1 Datasets

British Broadcasting Corporation Data

The first data set we are using is a collection of 2225 news articles from the British Broadcasting Corporation (BBC) from the years 2004-2005, which was originally provided for use as part of paper on diagonal dominance (Greene and Cunningham, 2006). The data set came in the form of individual text files for each article, separated into five categories: business, entertainment, politics, sports and tech. We took all the text files and combined them into one CSV file which we used in our experiments. We dropped duplicate articles based on title, which removed about 150 articles. We gathered some initial statistics about this dataset as seen below in Figure 1. As you can see, the distribution of the articles between the five categories is somewhat even, with each category making up 15-25% of the dataset.

Figure 1: BBC Dataset Statistics

Total number of articles	2225
Number of articles without duplicates	2096
Average article length	377 words
Average title length	5 words
Business articles	510
Entertainment articles	365
Politics articles	397
Sports articles	500
Technology articles	333

Web Scraped Data

The second data set we collected for topic classification contains 4730 news articles from 21 news outlets in the Lower Mainland. These articles were collected by using a web scraper that we wrote using the open source Python library, *newspaper3k* (Ou-Yang, 2013) . Some statistics about the dataset are shown below in Figure 2. The information that we are collecting about each article includes news outlet, article title, and article contents. We could not collect the topic of the news articles with our web scraper so this data has unlabelled categories. We did not end up using this data very much, due to the fact that it does not contain category labels, but it was still used for sentiment analysis.

Figure 2: Web Scraped Dataset Statistics

Total number of articles	4730
Average article length	461 words
Average title length	10 words

Positive and Negative word Corpora

For the sentiment analysis of our news articles, we use opinion lexicons to estimate the likelihood of an article identifying as positive or negative, by comparing the words in these corpora to words in a newspaper article. There are 4783 words in the negative corpus and 2007 words in the positive corpus (Hu and Liu, 2004). We found these corpora in an open source GitHub, that had been used as part of a paper on ”Mining and Summarizing Customer Reviews” (Hu and Liu, 2004). We also make use of a contraction dictionary from a GitHub repository we found, to expand contracted words into their full form before performing sentiment analysis and topic classification. (Shyam, 2014)

4.2 Sentiment Analysis

For our first task of sentiment analysis, we wrote an algorithm which takes in either of the BBC or web scraped data sets, and compares each articles contents and title to a corpora of positive and negative words. The number of occurrences of positive and negative words is tracked, and an article’s contents or title will be classified as ”positive” if the number of positive words is greater than the number of negative words, and vice versa. Neutral classifications occur when the number of positive and negative occurrences are equal (or both 0). After running this on both data sets we found that there tended to be more positively classified article contents, as shown in Figure 3.

Figure 3: Sentiment Analysis Classification based on Article Contents

Sentiment	BBC Data	Web Scraped Data
Positive	1226	3417
Neutral	134	306
Negative	736	1007

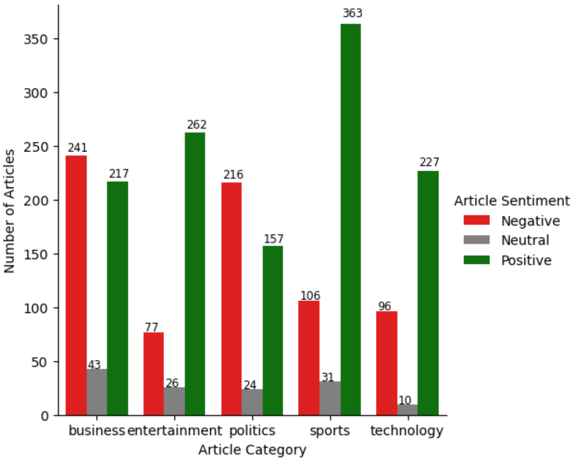
We also ran the sentiment analysis on just article titles, but the results were somewhat un-interesting. Since most titles were between 5-10 words, there would be no occurrences of a positive or negative word and the title would be classified as neutral. An example of this is shown in Figure 4. As you can see the majority of the article titles received a neutral classification. Originally, we were going to compare the classification of an article based on its contents, and the classification based on its title to see if there was any statistically important trends, but since most titles are neutral, we abandoned this idea.

Figure 4: Sentiment Analysis Classification based on Article Titles

Sentiment	BBC Data
Positive	452
Neutral	1406
Negative	367

Instead, we decided to look at which topics of news tend to be more positive or negative. To do this we used the BBC data which was given with the news split up into 5 categories. The results of this can be seen in Figure 5. From this graph we can see that articles in the business and politics news categories tend to be more negative, with 48% and 54% negative classification respectively. The other 3 categories are much more positive with 71%, 73% and 68% positive classifications for entertainment, sports and technology articles.

Figure 5: Sentiment Analysis Classification of News Categories using Raw Counts

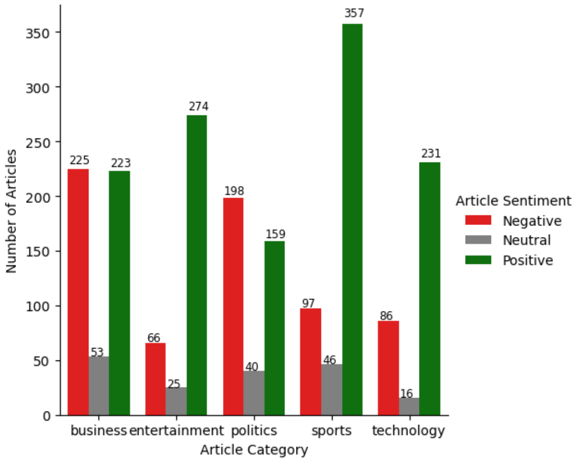


Our second approach to sentiment analysis was to make use of TF-IDF. TF-IDF is the term frequency-inverse document frequency, and measures how important a word is to a document in a collection or corpus. We used the TF-IDF function that is part of the *scikit-learn* library. We ran TF-IDF on all the article contents from the BBC articles, using unigrams and bigrams, keeping the top 5000 words as part of our dictionary. We ignored words with a document frequency over 60%, and words that appeared less than 10 times. After obtaining the collection of words deemed ”important” we performed sentiment analysis on the articles again, but this time only counted positive and negative words that TF-IDF labelled as important. This way, words that may be repeated many times, but happen to have a positive or negative meaning, don’t carry as much weight in the final classification.

Here we show a similar graph to Figure 5. This graph contains the results of the sentiment analysis using TF-IDF, per category.

Below in Figure 7, you can see a table containing

Figure 6: Sentiment Analysis Classification of News Categories using TF-IDF



the comparison of the total number of positive and negative classifications using the raw counts, or the TF-IDF counts. The numbers are not far off from each other, so one may be tempted to assume both methods perform the same, but we will demonstrate that they actually gave fairly different results.

Figure 7: Comparison of TF-IDF and Raw Counts Sentiment Classifications per Sentiment

Sentiment	Raw Counts	TF-IDF
Positive	1226	1244
Neutral	134	180
Negative	736	672

The graph in Figure 8 shows the number of articles which received a different classification using the two methods, which is hard to see from the previous graphs in Figures 5 and 6 since those just compared the total counts of articles in each sentiment and category.

As you can see, the majority of the articles received the same classification using both methods, but the number of articles that did not receive the same classification is not insignificant. Figure 9 is a table showing the same results but in terms of percentages. The percentage of articles that recived a different classification using the two methods was between 10-28% in each category.

Figure 8: Comparison of TF-IDF and Raw Counts Sentiment Classifications per Article

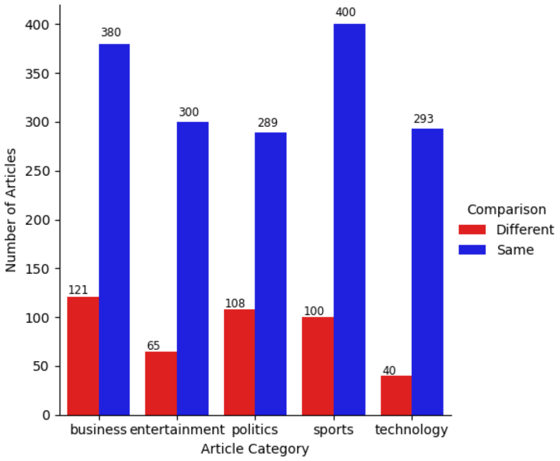


Figure 9: Comparison of TF-IDF and Raw Counts Sentiment Classifications by Category

Category	Same Sentiment	Different Sentiment	Percentage Different
Business	380	121	24%
Entertainment	300	65	18%
Politics	289	108	27%
Sports	134	180	20%
Technology	293	40	12%

This result was not unexpected, as we had hoped that the two methods would yield different results, to show that the importance of a word to a document does affect the sentiment an article receives. Since the sentiment analysis we are performing is on unlabelled data, we have no way of saying which of the two methods, using raw counts or TF-IDF data, are better. We believe that since both methods classified over 70% of the articles into the same sentiment, that there is some merit in these approaches.

4.3 Topic Classification

For this task we will be using the BBC data set since it is split into categories already. We use the BBC data set for fine tuning the parameters, and will check the accuracy of our predicted topic classifications, since the data is already labelled. We use two Non-Negative Matrix Factorization models, and an SVC model from the *scikit-learn* and *Gensim* Python libraries.

Briefly the pipeline works as follows. We train an NMF model on all the articles contents and find the optimal number of topics to classify the articles into. Once the best number of topics is found, the second NMF model takes in the TF-IDF processed article contents, and fits the model to the data, using the best number of topics from the previous model. This model is then used to transform the article data according to the model, into a document - term matrix. After transforming, the best 8 topic words that describe each topic from the previous NMF model are extracted by using the factorization matrix produced by the second NMF model. These 8 words are considered as the ”topic”, which are then passed into a SVC classifier. The SVC model is then used to predict which of the 5 categories each 8 word topic string falls into, effectively classifying newspaper articles. Here is a high level example.

Article: *”Hollywood stars Kevin Spacey and Kate Bosworth attended the British premiere of new film, Beyond the Sea, in London’s Leicester Square on Thursday. . . ”*
→NMF
→Topic: *Film, festive, star, movies, director, actor, cinema, oscars*
→SVC
→Category: *Entertainment*

To start, we performed pre-processing on the article contents that included tokenizing the article contents, converting all words to lower case, removing numbers, expanding word contractions, removing stop words, removing punctuation, removing words of length 1, and word stemming. We used a Porter stemmer from the *nltk* Python library, to reduce words to their base words. For example, the words likes and liked would all get replaced with like. Stemming is more of a crude process that chops off the ends of words to try and get the correct base form of a word to simplify our data and make it easier and faster to train. We also tested using a Snowball Stemmer, and Lancaster Stemmer (also known as the Paice-Husk stemmer), but ended up using the Porter Stemmer.

The Porter stemmer and Lancaster stemmer are rule based algorithms for removing suffixes of words, with the Lancaster stemmer being more aggressive. After testing our pipeline with all three different stemmers we found the Porter stemmer provided the best overall number of correctly classified articles, as seen in the table in Figure 10. One thing to note is that the Lancaster stemmer performed much better at classifying technology articles than the other two, but it also lowered the accuracy in other categories, probably due to overstemming of words.

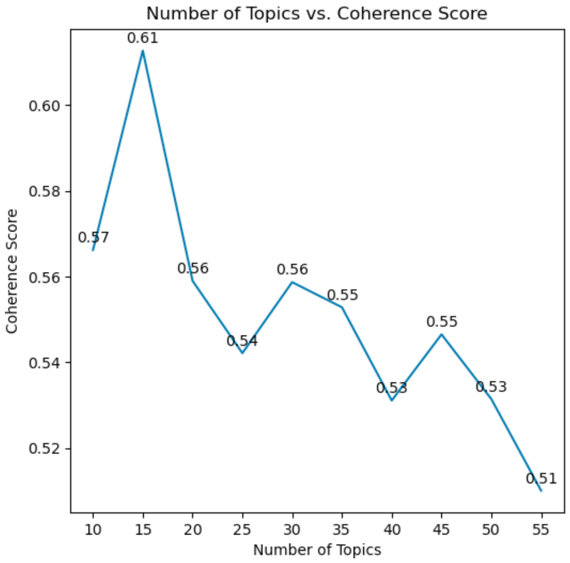
Figure 10: Comparison of Number of Correctly Classified Articles using Different Stemmers

Category	Snowball	Lancaster	Porter
Business	446	402	466
Entertainment	313	257	336
Politics	269	383	379
Sports	497	498	498
Technology	162	246	147
Total Percentage Correct	80.4%	85.7%	87.1%

Another approach which may have been better, but which we did not have time to experiment with would have been to use a Lemmatizer. Lemmatization is similar to stemming in that it reduces words to their base, but it does so using a vocabulary and morphological analysis of a word to return the base word, know as the lemma. Lemmatization may have returned better results given that it is a more sophisticated method.

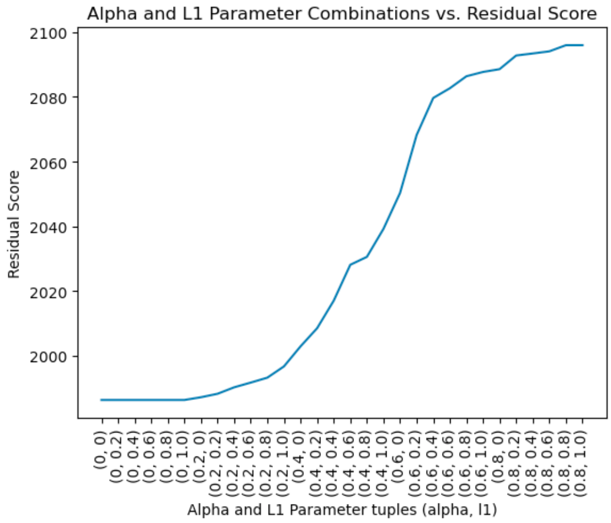
The first NMF model we trained was using the NMF implementation from the *Gensim* Python library, and the second from the *sci-kit learn* Python library. The first model was trained using different topic numbers ranging from 10, 15, 20 ... 55. We ranked the different models trained with different topic numbers based on the coherence score of the topics, which we used a Coherence model to obtain. Below in Figure 11 we show the coherence scores for each number of topics. From this data we picked 15 as the optimal number of topics to partition our data into, since it had the highest coherence score among the values we tested.

Figure 11: Comparison of the Number of Topics and their Coherence Scores



After getting the optimal number of topics, we trained another NMF model, this time experimenting with different alpha and l1 ratio values. We tested values from 0, 0.2, 0.4 ... 0.8 for alpha, and 0, 0.2 ... 1.0 for the L1 ratio. Alpha is a constant that multiplies the regularization terms, and the L1 ratio is the regularization mixing parameter. L1 of 0 means we are using an elementwise L2 penalty. Figure 12 shows graph of the different parameter combinations compared to the residual score. As you can see, the combination of alpha=0 and L1 ratio=0 had the lowest score, which means the parameters were a good combination, so we used those in our model.

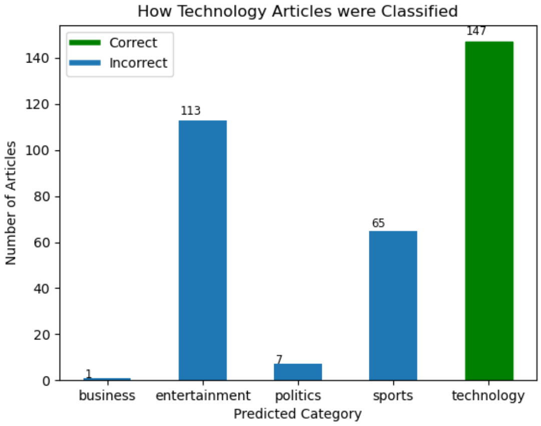
Figure 12: Comparison of Alpha and L1 Ratio Parameters and Corresponding Residual Scores



Finally, after training the NMF model with the optimal parameters, we used the model's results to assign a topic consisting of 8 keywords to each article, from a pool of 15 topics, the optimal number of topics as mentioned earlier. These 8 words are then passed into the SVC model, which then maps them to one of the 5 categories, business, entertainment, politics, sports or technology. The SVC model was trained on the article contents with their corresponding correct category labels. A Grid Search Cross Validation model was used to create a model for every combination of hyperparameters specified and then evaluate each model on the data to see which had the best estimator.

We explored different parameter sets with the grid search. We tested 3 different types of kernels, linear, polynomial and RBF (radial basis function). For each of these kernels we tested four C values: 0.0001, 0.001, 0.01, and 0.1. For the polynomial kernel we also tested it with degrees 3, 4 and 5. For the RBF kernel we tested it with gamma values 1, 10, and 100. Overall the grid search tested 28 different fits. We found that the best parameter set returned by the grid search was a linear kernel with C = 0.1. The gamma parameter defines how far the influence of a single training example reaches, and the C parameter acts as a regularization parameter.

Figure 13: How Technology Articles were Classified



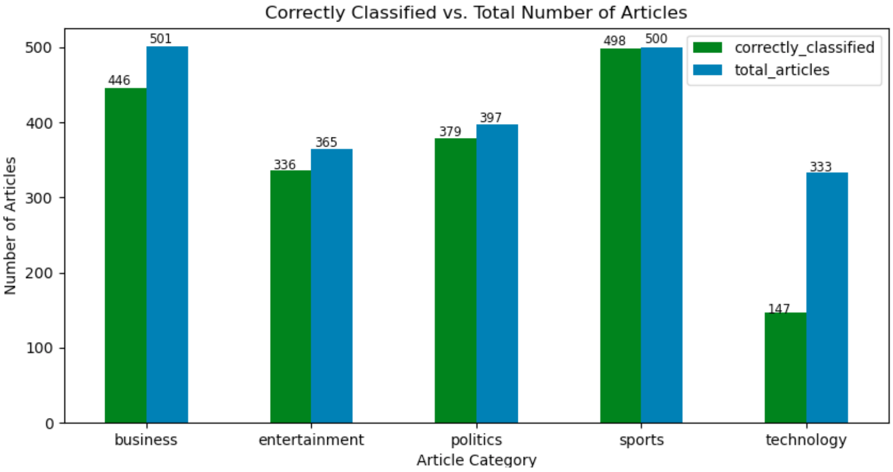
We have found that we have good accuracy with our model so far, but particularly the technology category has lower than average accuracy. As seen on page 7 in Figure 14, each category had a large portion of the articles correctly classified, with the exception of technology. Green represents the number of correctly classified articles, and blue is the total number of articles. As mentioned at the beginning of the report, we wanted to further investigate why the technology category had such a low correct classification rate. To gain further insight, we looked at which categories the technology articles were incorrectly predicted as.

As seen in Figure 13, the majority of the incorrectly classified technology articles were given either an entertainment or sports classification. This had been our prediction, since there are lots of common words shared between these categories such as ”game” and ”play”.

5 Geotagging Application

As a way to visualize our data, we created a map based user interface using Vue.js and an API called MapBox. To do this, we used the Python libraries *Spacy* and *geopy*. We used a pretrained Spacy model en_core_web_sm and scanned through an articles contents and used the spacy model to label

Figure 14: Topic Classification by News Categories



(Since we are unfamiliar with Latex, especially this paper template which has double columns, we could not figure out how to get this image to be on page 7 rather than 8. We could only get it to stick to the top of this page and span both columns.)

each entity in the article. We then filtered out the Geopolitical Entity (GPE) tags (aka. locations), and counted which location word appeared the most. We then used geopy to get the coordinates for the most common location name for each article, and if an article did not contain a location, we assigned it to London, since this is where BBC is located. The results of this can be seen at our live deployment here.

6 Conclusion

While there are many different ways to perform sentiment analysis, we found that the lexicon based approaches we used produced results which were interesting to analyze. It was observed that the Politics category tended to have more negative articles than other categories. The results from our topic classification methods were much better than we had expected, with over 80% accuracy in all categories with the exception of the Technology category. Future work described below will explain how we plan to further improve our methods and models.

7 Work Division

Our team distributed the work for the project quite evenly. Shayna and Anmol found the datasets we used, and cleaned them for processing. Anmol implemented the sentiment analysis with raw counts and Shayna used the TF-IDF method to weight the words. Patrick and Argenis trained the NMF and SVC models and implemented topic classification of news articles, and Shayna helped with searching for optimal hyper parameters. Patrick and Argenis also created the map-based user interface application. We all created the slide presentation for the poster session, and Shayna created the video and voice over. Anmol wrote the python notebook to document our analysis, as well.

8 Future Work

Currently, our text is preprocessed using the stemming method and to improve the accuracy of our system, in the future, we plan to use lemmatization which uses a morphological approach to modify the word to the appropriate base form.

We have plans to replace the word-based sentiment analysis to a sentence-level structure for future tasks to improve the accuracy of the sentiment analysis since it would take into account the semantic information of a sentence and train the content of the sentences to recognize the polarity. This can fix the cases such as when a word is preceded by a negation for which it is incorrectly counted as a positive sentiment. We will also work to minimize the number of articles classified as ’neutral’ by exploring larger opinion corpora which includes more variance of words, or adding relevant words to the existing lexicons.

Our plans for future work regarding the topic classification is to find a better method to classify technology articles so they are not incorrectly categorized as sports and entertainment based articles. Also, with more time in the future, we hope to further explore developments using operations from TF-IDF, NMF, and SVC to minimize the training time for the models, and to optimize the topic classification model, overall.

Currently, our front-end application that maps the data according to topics and/or sentiments is not optimized and unpolished so we would like to improve this in the future.

Furthermore, we will explore if and how our results from the sentiment analysis impact the economy and the society. We formulated this objective from a similar project that used a method to study the impact of news articles on the prices of stocks (Stockl, 2019). Using the news articles, we could explore the topics they were classified as to see if the sentiments had any impact on the field of the respective topics. For example, how did the current situation of COVID-19 impact the sports events that were scheduled for the year? Did major sports associations lose revenue compared to the previous year? These type of queries can show a cause-effect connection between our results and the effects on the real world. Answering these questions with our model will make it more beneficial.

References

Derek Greene and Padraig Cunningham. 2006. Practical solutions to the problem of diagonal dominance in kernel document clustering. *International Conference on Machine Learning*.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Bing Liu. 2010. Sentiment analysis and subjectivity. *Handbook of Natural Language Processing*.

Lucas Ou-Yang. 2013. Python library: newspaper3k. *GitHub repository*.

Neal Shyam. 2014. Expand common english contractions. *GitHub repository*.

Andreas Stockl. 2019. “newstrace” — a method to analyze the impact of news articles on stock prices. *Medium*.

Soonh Taj, Baby Bakhtawer Shaikh, and Areej Fatemah Meghji. 2019. Sentiment analysis of news articles: A lexicon based approach. *2019 International Conference on Computing, Mathematics and Engineering Technologies*.