A fake news detection system

Timon Brudna
Stefan Hoffmeister
Luca Müller
Tom Tietz

ABSTRACT

Fake news are becoming a highly relevant topic in modern society, as the spread of false information is on the rise. To answer the question “Can a text analyzing system reliably detect fake news?”, we used a machine learning model In order to achieve this, we explored different methods for natural language processing to extract characteristics of fake and real news articles. These characteristics can then be used to differentiate between the two. With these features we built a machine learning model which is able to classify whether a given news article is real or fake.

1 INTRODUCTION

The number of articles which contain misleading or outright incorrect information is substantially increasing. Furthermore, these articles are becoming harder to differentiate from real news. To add to that social media is helping to spread them. As a result, more people are blindly believing the information disseminated by these news. The dimension of these consequences could also be seen in the recent history. Therefore it is important to be able to distinguish between trustworthy news and untrustworthy ones. To define what a fake news article exactly is, we referred ´ to the definition found in the Cambridge dictionary. It defines fake news as “false stories that appear to be news, spread on the internet or using other media, usually created to influence political views or as a joke.”[3] We will only focus on the first type of fake news, the ones “created to influence political views”. A system which can reliably detect these news, could help minimizing the necessary guesswork to be made by an individual. Thus, our goal was to create a model which can determine if an article is true or false.

2 BACKGROUND

We used the *Natural Language Toolkit* to generate features from a text, which can then be used by the model. We also determined the complexity of an article with the *textstat* library. Besides those two libraries we also used common libraries like *NumPy*, *Sklearn*, *Pandas* and *Matplotlib*. In this context we refer to “Detecting opinion spams and fake news using classification”, by Ahmed H., Traore I. and Saad S., who worked on this topic([1]).

3 METHOD

This part will focus on the data we used and the steps we went through to eventually create our machine learning model. It is structured chronologically regarding our work process. First we explain how we obtained the dataset. Following that we discuss the data in detail. Then, we elucidate about our approach on the exploratory data analysis on said data in order to extract our features. Finally, we present the classifier which resulted from the exploratory data analysis.

3.1 Data

Our first step was to find suitable data to work with. After some online research we found the "Fake and Real News dataset" on kaggle.com[2], which was assembled by the fact-checking organization Politifact and facebook. Each sample represents an article, with four features, "title", "text", "subject" and "date". The title and text represent the unchanged title and text of the article, the subject represents the category of the article, like "politics" and the date represents the release date of the article. The dataset contains 17,903 unique samples with the label "fake" and 20,826 unique samples with the label "true". Over the course of our work we also looked for more datasets yet, the datasets we found were of poor quality. One such example was the "Fake News Corpus", an open source dataset that we found on github[5]. Unfortunately, the associated csv file was malformed and contained many "None" values, so we decided to not incorporate them into our research.

3.2 Exploratory Data Analysis

After data collection was completed, we did exploratory data analysis to find characteristics of fake news in order to classify based on those characteristics. We did not just go with the features presented by the dataset, but rather look at different aspects of news articles in order to obtain meaningful information on fake news and how they function. Looking at the features, we first filtered, what we wanted to further investigate. The subjects as well as the dates of the articles were not relevant since these features do not give much information on the article and can be misleading. We also decided not to focus on the title of the article, because we assume that the small amount of words does not indicate, whether an article is true or fake. Thus, we only analyzed the contents of the articles, i.e. the texts.
For that we used several natural language processing techniques, like tokenization, extracting bigrams and word tags. From our findings we were able to extract three features to use for our classifier.

The first characteristic we observed was the readability of the article. For that we calculated the Flesch Reading Ease score[4]. It is calculated as follows:

$$206.835 - 1.1015\left(\frac{\text{total words}}{\text{total sentences}}\right) - 84.6\left(\frac{\text{total syllables}}{\text{total words}}\right)$$

The more difficult a text is to read, the lower its score. The difficulty depends on the average number of words per sentence and the average number of syllables per word. Our hypothesis was that fake news articles would be easier to read, as they intend to appeal to a wide audience. As a matter of fact, when we calculated the score for all of our samples, the fake news generally had a higher score, meaning they are in fact easier to read. The mean for the fake news articles was at around 43, while the mean for the real news

articles at just around 37. The median values were very close to the means. The Standard deviation for fake and real news were 16.85 and 12.52 respectively. The third quartile of the true news scores was also below the mean of the fake news scores, thus the readability seems to be a good indicator whether an article is real or fake.

Another characteristic we wanted to look at was words which regularly appear in fake news articles. After comparing single words in fake and real news articles, we found that, even after removing stop words, there were still many words, which do not convey a lot of meaning, like "said" and "also". We decided to instead look at bigrams, as they offer two main advantages. Firstly, they remove incidents of these words without much meaning, and secondly, by including words which accompany a certain word, they put it into more context. For all our samples, we extracted the 100 bigrams which appear most regularly in real and fake news articles respectively. Although there were 34 bigrams which appeared in both of these lists, their frequency mostly differed greatly. Looking at the bigram "hilary clinton" for instance, it was the second most appearing bigram in fake news articles, while for real news, it was only ranked 26th. In order to extract a feature from the bigram frequency, we came up with the so-called "bigram score", which works as follows: First, extract the n most frequent bigrams for real and fake news articles, where n can be any natural number, with a larger number yielding a better accuracy. To avoid overfitting and test data leaking into the classifier assessment, this is only performed on the training data in the classifier. After the n most bigrams are extracted the bigram score of a given article can be calculated with this function:

```
def calculate_bigram_score(bigrams,
    n_most_bigrams_fake, n_most_bigrams_true):
    score = 0
    for bigram in bigrams:
        fake_bigrams = len(n_most_bigrams_fake)
        for i in range(fake_bigrams):
            if(n_most_bigrams_fake[i] == bigram):
                score += fake_bigrams-i+1
                break
        true_bigrams = len(n_most_bigrams_fake)
        for j in range(true_bigrams):
            if(n_most_bigrams_true[j] == bigram):
                score -= true_bigrams-j+1
                break
    return score
```

where *n_most_bigrams_fake* and *n_most_bigrams_true* represent the aforementioned extracted bigrams. A bigram score above 0 indicates that the article uses bigrams which are more typical for fake news articles, while a score below 0 indicates that the article uses bigrams which are more typical for true news articles.

The final characteristic we analyzed was the parts of speech, which tells us what kind of words are used more frequently in fake news articles. We divided the parts of speech into five main categories: nouns, verbs, adjectives, adverbs and determiners. For those categories we calculated the relative frequency in which they appear in fake and real news articles respectively. We found that fake news articles contain around 2% more adverbs but 1.5% less nouns than real news articles. For verbs and adjectives, the frequency is almost identical and the frequency of determiners was too low(0.2% and 0.1% respectively) to make any meaningful assessment. Similar to the bigram score, we came up with the "parts of speech score", "pos score" in short. First, the mean noun and adverb frequency is extracted from the training data. Then the pos score is calculated by the following formula:

$$(\text{mean\_noun\_proportion} - \text{noun\_proportion}) + (\text{adverb\_proportion} - \text{mean\_adverb\_proportion})$$

where mean_noun_proportion and mean_adverb_proportion represent the extracted mean noun and adverb proportion of the training data and noun_proportion and adverb_proportion represent the noun and adverb proportion for the given article. Again, a pos score above 0 indicates that there are less nouns and more adverbs, which is typical for fake news articles and a pos score below 0 indicates that there are more nouns and less adverbs, which is typical for real news articles.

3.3 Classifier

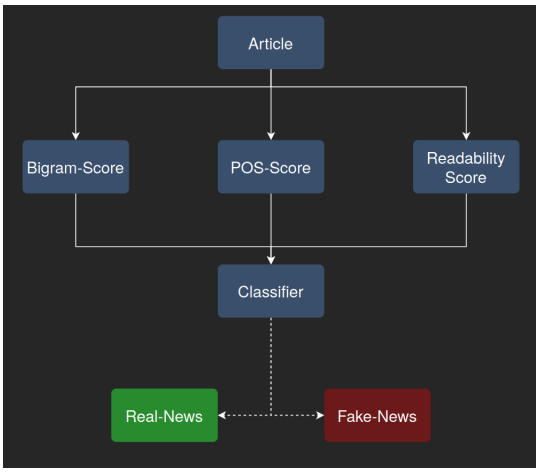

Figure 1: Fake News Classifier

Figure 1 shows our classifier and how it works. The implementation is encapsulated in a python class which can be used via its *fit* and *predict* methods. The class also takes a classifier as its parameter, which allows us to test the performance of different classifiers. The main purpose of the class is the data processing. As the figure shows, the inputs for the classifier's *fit* and *predict* methods are the complete texts of the articles. These are then processed to extract the features presented in section 3.2. Different natural language processing techniques are also employed. The Flesch Reading Ease Score needs to be calculated on the complete texts, as punctuation is important, while the bigrams and the parts of speech need the texts to be tokenized after punctuation and stop words have been removed. Also note that, as mentioned before, the *n_most_bigrams* as well as the *mean_adverb_proportion* and the

mean_noun_proportion are calculated only from the training data. Thus, as figure 1 shows, the calculation of the bigram score and the pos score can be conducted directly after the natural language processing during classification. After the features have been extracted, the *fit* and *predict* methods of the underlying classifiers can be called without any other preparation, as all features are now numeric values which any classifier can handle.
The amount of bigrams to include in the calculation can also be given as a parameter when using the Classifier to enable maximum flexibility for different use cases where a specific amount might be of interest.
Furthermore there is also the possibility to enable normalization of the feature values via Z-Normalization:

$$Z = \frac{X - \mu}{\sigma}$$

Depending, again, on different use cases this might be beneficial if one feature has a much larger value range than the others. Given that normalization might not always be wanted, it is kept optional.

4 RESULTS

In the following we will present the results of our work. It shows the outcomes of our classifier and the results this project yields including an answer to our research question. We also touch on the highlights and lowlights of our working phase. Finally, we discuss what our research implies regarding fairness, accountability, transparency and ethics in machine learning.

4.1 Classifier Results

The final classifier model was able to classify samples of fake and real news with an accuracy of **91.59%** (on 20% test data). While these results look very promising at first glance they were measured on only one dataset and might not be reproducible for more diverse datasets. Nevertheless our accuracy is more than satisfying for our first attempt at such.
As mentioned before the class for our classifier enables the option to choose whether the values should be normalized. In our testing, the accuracy differences between the averages of ten runs with normalization(Accuracy of 96,7%) and without(Accuracy of 97,3%) are negligible and running the classier on the entire dataset was thus performed without normalization.
Furthermore testing the classifier on a subset of the dataset with 2000 samples for fake and real news each, we got the following results for different classifiers: Testing K-Nearest-Neighbors, Support-Vector-Machine and a Neural-Network resulted in mean accuracies on five runs of 97,07%, 97,10% and 97,52% respectively. Given how close all these results were, we decided that using the Neural-Network would be the easiest, given the fact that when using the KNN-Classifier the k would have to be adjusted when changing the sample size.
On top of that the design of the model allows for easily changing the classifier. This means with more time and through cross-validation or examining the featurespaces, a more educated decision on what classifier to choose could be made.

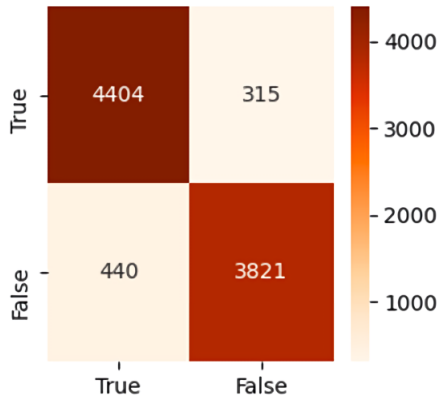

Figure 2: Confusion Matrix for a test subset of our dataset (8980 samples / 20%)

From this matrix we can also calculate other metrics of accuracy:

$$Precision = 90.92\%$$
$$Recall = 93.32\%$$
$$F1 - Score = 92.10\%$$

The confusion matrix as well as the calculated recall value show that our classifier is slightly biased toward predicting articles as *true news*. Therefore our Recall is higher and we end up with more *false positives* and less false negatives. In practice this means our classifier basically grants articles the 'benefit of the doubt'. We consider this behavior desirable when dealing with a delicate topic such as fake news, since an overly strict classifier could lead to censorship of articles which did not contain any false information but were merely written in a, to our model at least, suspicious way. It remains to be seen though why those 315 samples were false classified as *fake news* (same goes for the *false positives*). Later we will look a bit more deeply into possible reasons for these errors.

4.2 Research Results

At the beginning of this project we asked our self: "Can a text analyzing system reliably detect fake news articles?"
Now at the end of our research project we think it is safe to say that we can answer this question with "yes".
In more detail: We were able to derive three numerical values from each text which described:

- How common words that indicate fake news are in the text. (Bigram-Score)
- How high the frequency of adverbs relative to nouns is. (POS-Score)
- How readable the text is. (Flesch Readability-Score)

As shown by our data analysis and later the accuracy of our model these numerical metrics allow a good separation between fake and real news. Our analysis showed that fake news articles generally scored much higher on the Flesch Readability-Score (s. 3.2) which

makes quite a bit of sense since any producer of fake news would want to reach the broadest possible audience and spread their fake news to as many people as possible.
Another result of the analytical phase was the definition of those Bigrams which were most common in true or false articles. Some parts of the results like the most common fake Bigrams ”donald, trump” and ”hillary, clinton” showed that these probably tell us more about the usual topics of fake news than the truthfulness of a single article. This over-occurrence of politics in fake news articles (which can be attributed to the time span of our data) could bias our model towards classifying political articles in general as fake regardless their truthfulness. Despite this possibility we decided to keep the feature and in combination with the other two features it seems to be a good enough differentiator, at least for our dataset.
The relative frequency of nouns and adverbs and the differences between fake and real news in this metric are also unsurprising as articles which try to convince their reader of an untruth are usually more reliant on ’strong’ adverbs and at the same time don’t want to be overly complicated by using technical terms which tend to be nouns.
Following we want to take a closer look at some samples that were misclassified and see whether they unveil potential weaknesses of our model. Here are two randomly selected samples from the group of samples that were falsely classified as *fake news* (**false negatives**).

> "BEIJING (Reuters) - China said on Monday that it hopes all sides’ words and actions can help reduce tensions on the Korean peninsula, after Japanese Prime Minister Shinzo Abe said Japan would shoot down North Korean missiles if necessary. Foreign Ministry spokeswoman Hua Chunying made the comment at a regular news briefing."

> "The following statements were posted to the verified Twitter accounts of U.S. President Donald Trump, @realDonaldTrump and @POTUS. The opinions expressed are his own. Reuters has not edited the statements or confirmed their accuracy. @realDonaldTrump : - Jerry Falwell of Liberty University was fantastic on @foxandfriends. The Fake News should listen to what he had to say. Thanks Jerry! [0627 EDT] - Thank you, the very dishonest Fake News Media is out of control!"

While there is nothing extraordinary that can be said about the first example, the second example shows us several fallacies in our model. The continued mentioning of Donald Trump would make this article prone to being classified as false in the first place but another specialty of the article probably increased this likelihood. It contains a direct quote from a twitter post written by Donald Trump. Our model cannot differentiate between the words of the author and the words of other people he/she quotes in his/her article. It is not unlikely that this article was falsely classified as fake because of the rhetoric in the part of this article that consists of the quote from Donald Trump. We see this problem or similar ones in other articles that were falsely classified as well.
Even though the existence of this flaw might seem obvious, solving this problem is a much harder problem than the scope of this project would allow. To improve the accuracy of *fake news* detection one therefore had to use a much smarter and more sophisticated text analysis algorithm to differentiate between the words of the author, direct citations and even indirect citations.
Another thing that stood out about the false negatives is the high occurrence of very short news articles (like Reuters) which could probably attributed to simple lingual structure of these articles and the higher variance in feature values caused by less text.
When looking at all wrongfully classified samples though one things stands out immediately: None of the outliers defer from the norm of their class in just on feature dimension. Almost all of them have feature values that would categorize them as the opposite of their actual class. This shows us two things:

- Our classifier is fairly robust when it comes to singe-feature outliers
- Some articles just do not fit into the usual patterns at all and are almost impossible the be classified by our means

4.3 Highlights & Lowlights

During the process of our research project we ran into relatively few obstacles and were able to solve most problems in a straightforward manner.
We started our project with exploratory data analysis in order to find text-derived features with high discriminative power when it comes to differentiating between fake and real news. This process delivered very satisfying results as using the three numerical scores (s.3.2) as features resulted in a good accuracy and high variance in at least one of the feature dimensions. Analogous to the final models its precision and recall also ended up to exceed our expectation (s.4.1). In general we were able to positively answer our research question. We managed to create a classifier which fairly accurately detects articles containing fake news and we managed to do so without any fact checking data bases.
The only lowlight in our project is probably the fact that we were unable to find additional data sources of adequate quality to complement the data from Kaggle. This lack of variety in our training data obviously limits the real world applicability of our model.

4.4 FATE

In many cases when creating something that you want the general public to use there are several factors to look out for. In today’s day and age, people are starting to get more and more aware of the implications of using any kind of technological system, be it simply when worrying about how private data is handled, or seeing how different systems affect the public.
Bearing this in mind, it is all the more important for us to also think about the risks and responsibilities that come along with the machine learning model we created, given the fact that we set out to create something that can be used by anyone.

If people don't understand how the system works and what role it is supposed to play then we would be at least partially responsible on either them relying on misinformation or possibly even spreading information from articles that are not based in reality. Because no system is perfect, our model might well have some sort of bias to misclassify certain types of articles.
Incorrectly classifying an article could also not only be harmful because incorrect information gets spread, but also when a completely truthful article gets marked as fake. This could potentially hurt the reputation of a news outlet if people don't trust them because their articles get identified as fake.
Imagine a news outlet targeting a younger audience to get them interested in whats happening in the world. Because one metric we look at to identify fake news articles is the readability score these kinds of articles might be prone to being incorrectly identified.
That's just one of many possible scenarios where because of how the model is designed certain types of media might be discriminated against.
With why it's important to understand how our model works out of the way we can now briefly talk about how it actually works. The model only focuses on the text of the articles, this means the news outlet or even title are disregarded because these don't always help make assumptions about the article. With a completely text based model we also avoid having to e.g. compile a list of trusted outlets. From the text our model then extracts three features.

- **Bigram-Score**: This score simply looks at combinations of words that often appear in fake news and real news. With the knowledge of these combinations it then checks which of these combinations are also found in the article that is being tested. This means certain patterns are being looked at and compared. However it's important to understand that the meaning of these words is *not* checked. Neither is context for example.
- **Flesh-Readability-Score**: This was briefly touched on before. With this the model rates how difficult an article is to read. From looking at several thousand articles a pattern is visible, where fake news articles are, in general, easier to read.
- **POS-Score**: Having looked at the distribution of different types of words a pattern where fake news had more adverbs and less nouns used, was visible. With that knowledge in mind we look at the distribution of different word types and based on that build a score for the article.

Another problem is, that parties that want to create fake news articles can, with the understanding of how the system works, write articles which would be classified as real news by the model. This is the flip side of the coin, when being transparent as to how the system works. If it wasn't known what metrics the model employs to classify an article this wouldn't be possible. So one has to decide what matters most.
We think that knowing how the model works outweighs the resulting possibility of fake news outlets creating articles that would be classified as real news. As long as people know how to use the system and what it can and can't do, simply writing an article that gets identified as real news should hopefully not fool anyone. We assume they would aware of the weaknesses of model and would still consult more articles from different sources to either confirm or deny the statements made in the article in question. To finalize on the thoughts of fairness, accountability, transparency and ethics, we think it is imperative for us to make the ins and outs of the model as clear as possible. With full knowledge of what the model does, we hope that everyone can decide how to act in regard to the information that our model presents.
In the introductions we said, that we set out to create a system that could mitigate or at least reduce the amount of work that has to be put in, to identify a fake news article. When we came up with the research question and started working on our project. We simply tried to produce the highest possible accuracy, such that technically, if someone were to use the model, the amount of times the system was incorrect would be as small as possible.
But as it often stands with research even with relatively satisfying results for our testing, applying these methods in the real world is a whole different playing field and thus, doesn't guarantee similar accuracy.
So in saying that "A system which can reliably detect these news, could help minimize the necessary guesswork to be made by an individual." our question and whole research was focused on whether we could. On the other hand whether one should use the model is a decision that everyone has to make for themselves.

5 CONCLUSION

To create our model we first needed to find a dataset that we can use. Sourcing our own data through e.g. web scraping would have meant having to do a lot of work before we could even begin on our model and could have potentially not yielded any usable results. That's why we chose to go with already existing datasets.
Looking through the options we found the Kaggle dataset[2], which seemed to be a very good starting point with a solid amount of sample data for both real and fake news, as well as being very complete with not only metadata on the article but also the texts themselves.
Having settled on the Kaggle dataset to begin with, we then got to the stage where we could work on our research question. Finding out what we want to explore with that dataset, how we want to tackle classifying fake and real news.
Having settled on our research question we then started exploring the dataset for possible features to use for classifications as explained in section 3.2.
Having found three features that can be applied to all kinds of texts without the need for any metadata, we then started building our first basic machine learning model to see if our features were any good, before we spend a lot of time on features that yield a poor accuracy. In these first tests we got an accuracy of around 90%.
Knowing that we were on the right track we then decided to build a class which neatly wraps up all the methods and make sure, that there is no leakage between train and test data.
Based on our findings so far, we would recommend the following steps to take for future research.
One of the main problems we faced was the fact that we only found one dataset which was complete enough to be used. Especially because we rely on text only and a lot of datasets only really collect metadata about fake and real news articles. Finding more datasets which are potentially more diverse than just articles from America

would be interesting, especially to see whether the model would still hold up if the train set is more diverse. It would also be interesting if the observations made for English articles would translate into other languages, e.g. German.
Moving forward, the classifier could be tested on different data to assess whether our conclusions which went into the design of the classifier still hold true. If this is the case, we can assume that our classifier has a good general understanding on whether an article is true or fake. If the accuracy declines, further investigation into the features would be necessary in order to refine them.
Overall, our work provides a good baseline to build upon for classifying fake news articles, as well as helping with a problem that affects all of society.

APPENDIX

FRS	Bigram	POS
37.68	8	0.27430832456096343
49.55	58	4.536394838555868
54.39	-53	12.77430832456096
43.43	133	-14.580764139207156
48.57	25	-2.289794239541603
35.44	-62	4.151771906394721
48.13	99	6.497937016544078
37.61	233	3.9908364870724977
45.9	0	13.228103704098913
41.03	-53	7.487193478622584
49.86	40	-5.885111965294117
26.17	0	6.729707385593825
50.4	96	-2.503469453216816
47.93	95	-4.812271762019126
58.11	52	-5.152048264586331
41.09	-53	3.5409749912276256
50.09	48	-0.8537848340998844
42.82	-61	4.940974991227626
51.11	13	-4.529613244066489

Figure 3: False Negative feature values

FRS	Bigram	POS
37.68	0	-3.8923583421057115
36.66	-130	2.5124035626561927
58.82	-49	-4.684791879938016
8.71	-112	6.112097985880027
65.96	-71	-6.0182086822417675
10.44	-31	1.8229808909326417
48.57	-125	2.2424825289160664
44.82	-68	-1.8996047189173018
32.6	-1	-7.553974503721871
5.2	21	-3.930584641799893
17.71	-47	8.607641657894291
46.61	-170	-1.0454503481388935
29.56	-25	0.924025838685254
53.85	-125	-2.4646194143667826
5.8	-42	-3.157064224458649
24.89	-383	2.060022610275243
42.18	-83	5.274308324560962
36.56	-108	-3.1183135463217851
27.63	-189	6.58477446352314

Figure 4: False Positive feature values

Values for True Positive		
FRS	Bigram	POS
Min Values		
-67.59	-2716.0	-28.892358342105716
Max Values		
84.0	0.0	13.3695464197990054
Mean Values		
34.20387719298245	-458.6280701754386	-4.13459731955122
Values for True Negative		
FRS	Bigram	POS
Min Values		
-45.77	-383.0	-14.580764139207156
Max Values		
83.15	3201.0	18.131451181703817
Mean Values		
43.62249206349206	351.2888888888889	3.0264707808621134

Figure 5: Base values for TP and TN

REFERENCES

[1] Hadeer Ahmed, Issa Traore, and Sherif Saad. 2018. Detecting opinion spams and fake news using text classification. In *Journal of Security and Privacy, Volume 1, Issue 1.*

[2] Clément Bisallion. 2020. Fake and Real News Dataset. https://www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset

[3] Cambdridge Dictionary. 2020. Fake News. https://dictionary.cambridge.org/de/worterbuch/englisch/fake-news

[4] Rudolf Flesch. 1981. *How to Write Plain English.* Barnes & Noble.

[5] several27. 2018. Fake News Corpus. https://github.com/several27/FakeNewsCorpus