Text Mining - Fake/True News Articles Analysis

Rebecca Jones, Iván Aguilar, Juan Picciotti

April 2022

Introduction

The term ‘Fake news’ was popularised by Donald Trump in 2016, and is defined as journalism that contains incorrect or misleading information.

The rise in technology and social media has hugely increased the amount and diversity of news sources that people have access to, with around 62% of adults in the US consuming news from social media in 2016, relative to 49% in 2014 (Shu et al, 2017). In addition, the younger generation tends to rely on social media to educate themselves on political news and world events (Rubin, 2017).

While access to greater amounts of information is clearly beneficial, this increases the potential for the propagation of misinformation if the sources are not legitimate, which can have highly negative real world consequences. For example, misinformation regarding Covid-19 was a direct cause of lower vaccination rates for certain segments of the population.
To counteract this, identifying fake news is now a great cause of focus for governments and institutions (Yale Law School, 2017).

Detection of fake news is a highly complex task due to the fact that it is created with the intention of deceiving the reader. The result of this is that people are almost no better than random at detecting fake news articles (Zhou 2020, Wang 2017). In addition, the advancement of AI techniques means that ‘bots’ can create huge quantities of highly realistic content for a low cost to the actor spreading the fake news (Stahl, 2018), meaning that it is almost impossible for a human fact checker to solve this detection problem.

Hence, identifying fake news through computational means is a highly relevant and important topic. In this paper we first discuss and seek to understand and identify differences between fake and true news articles, and then we build on the techniques used in other literature to produce a fake news detection classification model.

Literature Review

Many papers have explored the topic of fake news classification using a text-based approach. The majority use the frequency of terms (tokens) and/or the frequency multiplied by the inverse

document frequency as a feature input, as well as some stylometric characteristics such as the length of document, length of words or amount of punctuation (Reddy et al., 2020).
In addition, some use word embeddings, as measures of sentiment analysis, as features for the classification model. For example, Yang et al. (2018) use word embeddings to try and incorporate latent (hidden) features into their model.

Lastly, some authors expand their features past text analysis in order to improve their predictions. For example, Shresthsa (2018) uses network metadata to analyse the reliability of sources, such as the URL, author, and number of social media likes. Jin et al (2016) go further and also use images as feature inputs, obtaining improved results compared to sole text analysis.

Once the features have been identified, many studies apply a binary classification method to identify fake news or true news. There are many different techniques that can be employed, for example Ahmed (2020) uses binary classification methods of Naive Bayes, Passive Aggressive (remains passive for correct predictions but penalises incorrect predictions heavily) and Support Vector Machines, and finds that the passive aggressive classifier performs with 93% accuracy. Shrethsa (2018) uses a Random Forest as a classifier, which has the benefits of using a combination of numerical and categorical feature inputs.

We contribute to this literature by first performing some analysis prior to our classification exercise to allow us to better understand and observe differences between the true and the fake news, including a polarisation metric and topic modelling.

For our classification model, we use the all the words of the corpus, only excluding basic punctuation, to create word embeddings as a layer for a Neural Network, similar to the implementation by Yang et al. (2018) and as originally described by Mikolov et al. (2013)

Datasets

The dataset we use for our analysis and to train the classification model is a collection of US political news articles, sourced from Kaggle (2017). The ‘true’ news data was originally sourced from trusted news website Reuters, and the ‘fake’ news data was sourced from Politifact, a website that aims to fact check news and identify misinformation. The dataset contains 21,192 and 22,851 true and fake news articles and their titles respectively, with publishing dates ranging from January 2016 to December 2017.

To test our classification model, we obtained separate test data from a variety of both left-leaning and right-leaning news sources, with the publishing dates matching those of the training dataset. The sources include New York Times, Breitbart, CNN, Business Insider,

Atlantic, Fox News, Talking Points Memo, Buzzfeed News, National Review, New York Post, Guardian, NPR, Reuters, Vox, Washington Post, with an approximate total of 144K articles.

We also aimed to test the classification model with current news outside of the date range of the trained model. To do so, we scraped front page news from the 'CNN lite' news website, which contains an average of 90 articles on a given day.

Methodology Summary

In order to understand the datasets in more detail, we first perform exploratory analysis and display some basic statistics. We then employ a 'polarisation' metric to analyse the differences in language between the true and fake news sets. The metric was first employed by Gentzhow and Shapiro (2010) when comparing the polarisation of language used by the Republican and Democratic parties in the US. We then perform Latent Dirichlet Allocation (LDA) separately on the two data sets, which is an unsupervised learning technique that learns 'topics' within text. This allows us to examine whether there are differences in content between the true and fake news articles.

Lastly, we construct an embedding neural network architecture, to which we apply our supervised training data, to train a model to learn which class belongs to. Afterwards, we perform predictions on our test (unsupervised) data as an empirical test for our model.

Exploratory Analysis

We first perform some basic analysis to explore our data in greater depth. The number of words per article, after the removal of stop words, is similarly distributed for both true and fake articles, with the majority between 50 and 300 (Fig. 1). However, we observe differences when analysing other metrics - for example the vocabulary in the true news is more diverse than that of the fake news (Fig. 1). In addition, the proportion of capital letters used in the title of the fake news articles is distributed around 30% (with a spike at 100%) - much greater than the true articles' proportions, distributed around 10% (Fig. 1).

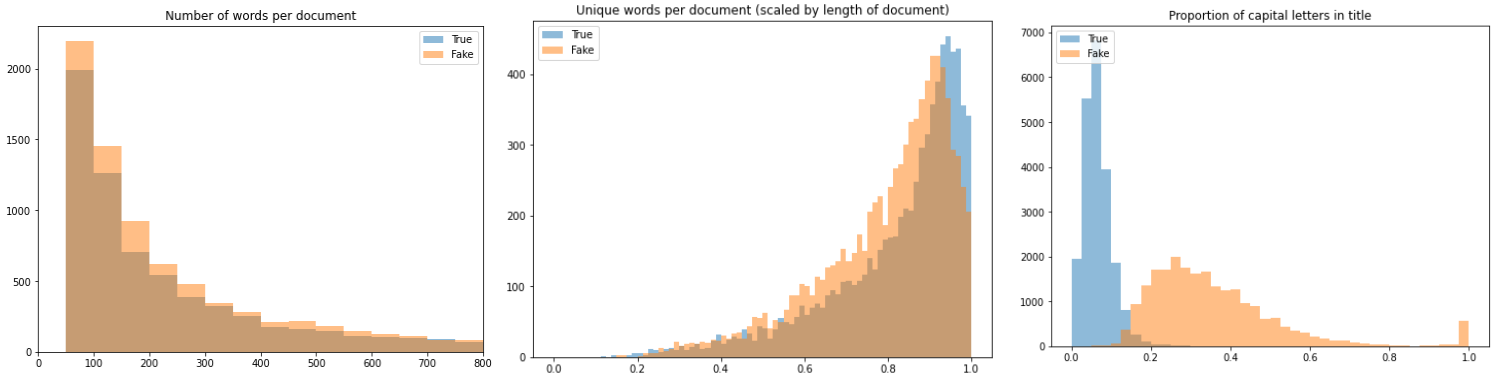


Fig 1 - Exploratory analysis: number of words per document (left), number of unique words per document, scaled by document length (centre), proportion of capital letters in document title (right)

For each of the true and fake datasets, we tokenize and remove punctuation, common stopwords and additional words that do not provide meaning, as well as words that appear in at least 100 documents but less than 70% of the total documents. As seen in the word cloud (Fig. 2), there is not an immediately obvious difference between the most frequent terms - we note that terms relating to president Donald Trump and Barack Obama are commonalities in both.

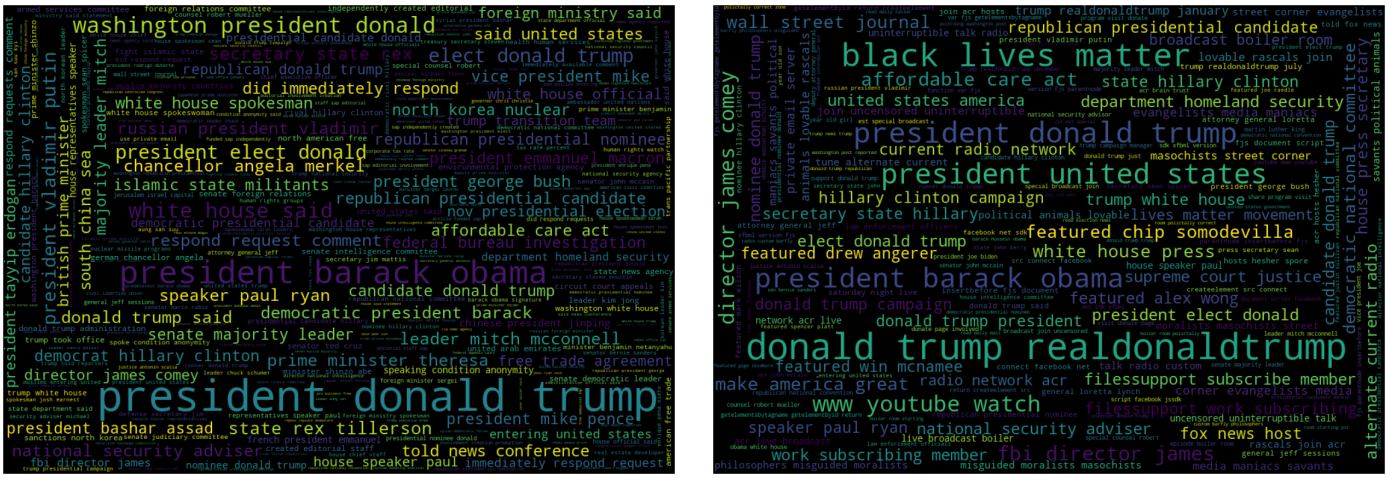


Fig. 2 - Word cloud of true news articles (left) and fake news articles (right)

Multiplying the term frequencies by the inverse document frequencies (to reduce the importance of words that are spread across many documents and increase potentially meaningful but less frequent terms), we again observe from Table 1 that the tokens with the top 5 scores do not

differ greatly. Therefore for the remainder of our exploratory analysis, we remove the tokens ‘trump’, ‘obama’, ‘president’, ‘united’ and ‘states’.

	True			Fake		
#	term	tf	tfidf	term	tf	tfidf
1	president donald trump	5872	1524.59	donald trump realdonaldtrump	1692	635.524
2	president barack obama	2959	783.446	president donald trump	1044	631.401
3	washington president donald	1137	396.250	president united states	998	586.147
4	prime minister theresa	584	360.892	black lives matter	1304	569.698
5	white house said	967	315.55	president barack obama	949	534.070

Table 1 - Words with top 5 term frequency / inverse document frequency (tfidf) score for both true and fake news articles

Polarity

We next aim to observe how “polarised” the true news articles are compared to the fake news articles to further understand differences between the two datasets. To do so, we implement the Chi Squared metric. This measure was first used by Gentzhow and Shapiro (2010) to compare the polarity of language used by the Republican and Democratic parties in the US. The Chi-squared formula is given by:

$$X_v^2 = \frac{(x_{v,F}x_{\sim v,T} - x_{v,T}x_{\sim v,F})^2}{(x_{v,F}+x_{v,T})(x_{v,F}+x_{\sim v,T})(x_{\sim v,F}+x_{v,T})(x_{\sim v,F}+x_{v,T})}$$

Where $x_{v,F}$ is the frequency of term v in the fake news articles and $x_{\sim v,F}$ is the frequency of all words aside from v (and the same notation is present for the true articles).

This metric captures the uniqueness of each word to either dataset. Analysing bi-grams and tri-grams with the highest Chi squared metric (Fig. 3), we observe that the most important true article terms are ‘north korea’, ‘human rights’ and ‘prime minister’’. In 2017, there was a deterioration in relations with North Korea [add ref] and during the same time period there was a

general election in the UK to elect the next prime minister. Therefore we could interpret this theme as news on global affairs, a theme which is present in the remainder of the top scoring terms.

For the fake news, we observe that ‘Hillary Clinton’, ‘mainstream media’ and ‘didn want’ are the highest scoring terms. The theme here is less clear, but the first term likely relates to the 2019 elections given that she was the favourite candidate, and the second might relate to the attempted smearing of ‘true’ news. Looking at the rest of the fake terms, we note that there are racial themes (‘black people’, ‘african american’) and also unwanted noise (‘featured screenshot’, ‘download episode’), which could indicate more clickbait and spam text present in fake news articles.

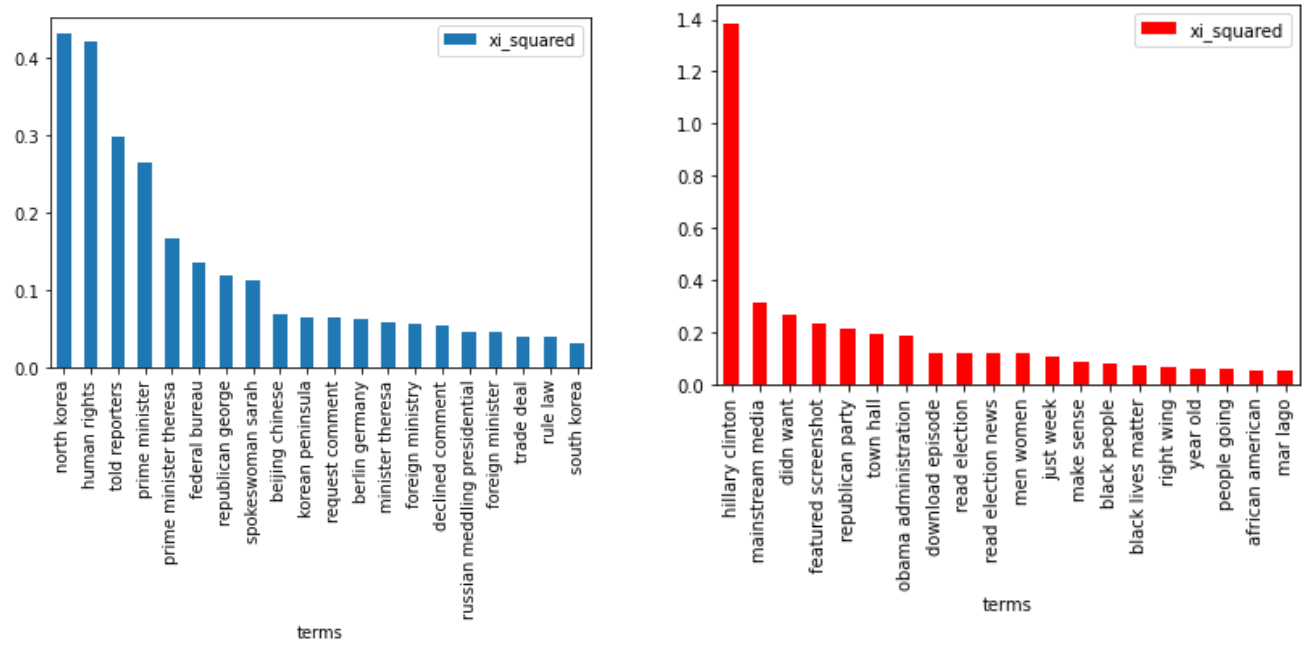


Fig. 3 - Top 20 words with the highest Chi-squared score for the true articles (left) and the fake (right)

Multiplying the relative term frequencies (i.e. normalised by the document length) of each token by its chi-squared value and summing these values for each document, we can observe a metric of polarisation and evaluate how this changes over time.

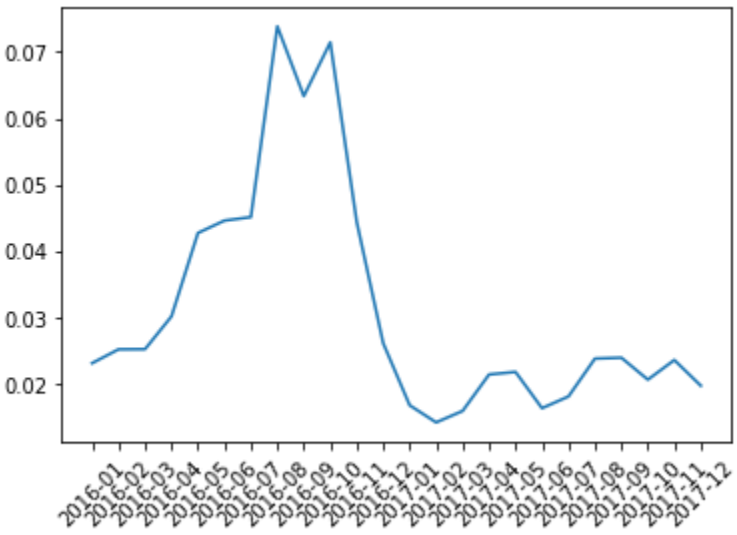


Fig 4 - Average Chi-squared score per document over time

We note from Fig. 4 that during the campaign period of the US general election (the months preceding November 2016), the metric of polarisation increases. This tells us that the difference in terms used increased in the run-up to the election.
To build further on the difference in themes between the two datasets, in the next section we apply a more sophisticated model - Latent Dirichlet Allocation.

Topic Modelling

Topic modelling is a method for unsupervised classification of documents, which allows for deriving insights from text corpora. In this section we describe how we used one of the models that can be employed for this task - Latent Dirichlet Allocation - in our corpus and what results we obtained.

Theory
Latent Dirichlet Allocation, first introduced by Blei et al. (2003) in their famous paper published in the Journal of machine Learning research, in 2003, to text analysis, is based on the idea that documents are represented as a random mixture over latent topics.
Each topic is characterised as a distribution over words. LDA assumes that words of each document arise from a variety of topics where each topic is a Multinomial over a fixed vocabulary.
The topics are conveyed by all documents in the collection, as they are randomly drawn from a Dirichlet distribution; hence topic proportions vary across documents stochastically (Blei, 2007)

In mixed membership models, each document (indexed by m) is assumed to generate as follows:

- First a distribution over topics, θ_m, is drawn from a global prior distribution. In our case, uniformly distributed.
- Then, for each word in the document (indexed by n), we draw a topic for that word from a Multinomial distribution based on its distribution over topics ($z_{n,m}$ ~ Mult(θ_m)). Conditional on the topic selected, the observed word $w_{m,n}$ is drawn from a distribution over the vocabulary $w_{m,n}$ ~ Mult($\beta_{zm,n}$), where $\beta_{k,v}$ is the probability of drawing the v-th word in the vocabulary for topic k. LDA, assumes a Dirichlet prior for the topic proportion such that β_m~Dirichlet(α).

Although LDA is a powerful tool to analyse text, it makes some restrictive assumptions. Firstly, it assumes that topics within a document are independent of one another; these come from assuming a Dirichlet distribution on the topics for a document. Secondly, the distribution of words within a topic is stationary, meaning that words are exchangeable within each document. Thirdly, topics are modelled entirely based on the text of the document.

Parts of Speech

In order to build a Topic Model, as for other applications of text analysis, after the pre-processing of the Documents, these are split into tokens (words and other terms) and a portion of them is removed due to carrying little value in terms of defining the underlying topics. Typically, predefined dictionaries of stop-words (tokens with small impact on topics) are used, in addition to manually generated data-specific collections of tokens.

Our strategy builds on this general approach by pre-specifying each token's role in the sentences they are part of. This classification into 'Parts of Speech (POS)' - syntactical tags: nouns, verbs, adjectives, etc. - allows for the removal of entire classes of tokens which offer little information for defining topics.

In our case, given the number of newspaper articles in our corpora (21,000+ true news, and 22,000+ fake news) and their lengths, the numbers of tokens are over 9 million (in true news), and 10.8 million (for fake news).

In order to make the analysis computationally cheaper, as well as removing stopwords, punctuation, numbers and symbols we also remove verbs, ad-positional phrases, determiners, pronouns, auxiliaries, adverbs, coordinating and subordinating conjunctions.

Specifying topics can be efficiently done by focusing on the people, places and events - public figures, countries, elections, pandemics, scandals, etc. - that are mentioned in the documents. This information is typically represented in the articles' nouns.

Despite adjectives usually being included for sentiment analysis and, arguably, could be disregarded for topic modelling, in our case they offer added value when considering n-grams (groups of n tokens that are found together). In our context, for instance, 'White' and 'House' alone have very different meanings from 'White House'.

Hence, for our analysis, we removed all POS except Nouns and Adjectives. As a result, almost 7 million tokens are removed from the fake news (leaving only 36.2% of the original quantity), and 6.5 million from the true news (leaving 39.6%).

For this purpose, we employed Python's NLTK tagger function, which has many different and precise categories for Parts of Speech, including many types of nouns, verb tenses, etc. In order to summarise the data, we grouped the POS into fewer sets (see Fig. 5).

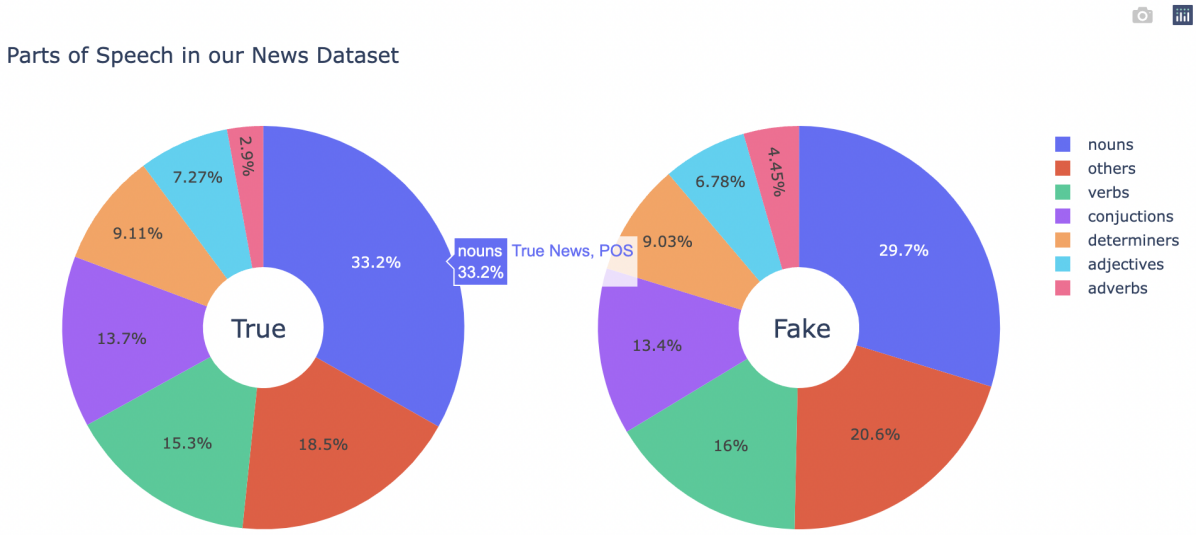


Fig. 5: Proportion of word types per dataset. Note: this is an interactive chart that can be fully viewed in the associated HTML file.

A dictionary based approach could be considered, where all nouns and adjectives are saved into a list, and then any words not present in the list filtered out.

However, sometimes the exact same word plays different roles in a sentence: for instance, "He gave him the *look*" and "You *look* nice". *Look* acts as a noun in the first case, and as a verb in the second. Hence, we take the approach of filtering out tokens considering their roles in the sentences instead of the words themselves.

Top 10 Nouns, Verbs, Adjectives and Adverbs for Fake and True News

Fake - Nouns	Freq
Trump	72709
people	24777
Clinton	17957
Obama	17727
Donald	17172
President	16469
Hillary	13029
time	12129
America	10507
media	10407

True - Nouns	Freq
Trump	53830
U.S.	36964
Reuters	28360
government	17591
President	17409
House	15369
United	15048
people	14243
States	12261
state	12072

Fake - Verbs	Freq
is	108574
was	67244
be	48239
have	45640
are	45483
has	41999
said	31021
been	22913
were	21475
had	20260

True - Verbs	Freq
said	99019
is	55097
was	47910
has	46220
have	36384
be	34256
are	26050
had	25643
been	19599
were	18892

Table 2 - Top 10 Nouns and Verbs in each corpus

Next, we perform some additional general preprocessing, which includes lowercasing, stemming and other basic tasks.

LDA Model

We use the gensim library to construct our models, one for each corpus. The three main arguments gensim's LDA model requires are the following:

- The preprocessed corpus
- The associated dictionary
- The number of topics

We create bigrams and trigrams with gensim's native function, modifying the arguments 'min_count' and 'threshold'. The higher their values, the harder it is for words to be combined into bi/trigrams. We additionally create the associated dictionary employing the library's tools.

As for defining the number of topics we want to work with, this task can be done by computing different diagnostic values (e.g. Held-Out Likelihoods, Semantic Coherence, Residuals methods) for a varying number of topics, and then choosing an "optimal" *k*. In our case, we simply intuitively analyse results for values of *k* between 8 and 12, and find that 8 produces the most clear and interpretable topics.

Additional important arguments for gensim's LDA model are:

- alpha: A-priori belief on document-topic distribution.
- eta: A-priori belief on topic-word distribution.

Since we have no prior insights on how topics or words within topics could be distributed, we did not input these.

Topic Modelling Results

We manually label the resulting topics as can be seen in Table x. Note that we also create an in-detail interactive dashboard that allows for more in-detail of the topics. See the attached HTML file to access the dashboard, and Appendix A.2 for instructions on how it can be used and interpreted.

Fake	True
Police shootings	UK elections
US elections between Clinton and Trump	US elections
Political affairs in the US	Puerto Rico State
Syrian war	Refugees
Trump	Industry - Regulation in the US
FBI investigation on emails from Clinton	Russian interference in US Elections
Media depiction of politics in the US	Iran - Kurdish
Racism in the US	China, North Korea and Foreign Affairs

Table 3 - Manually labelled topics produced from LDA analysis

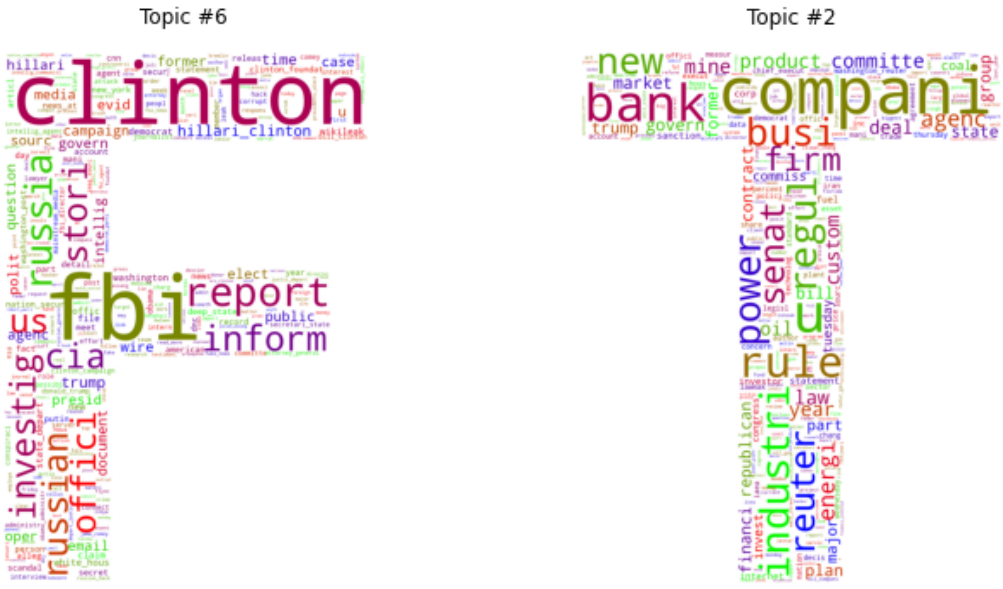


Fig. 6 - Word-clouds depicting terms usage in different topics in Fake and True news. Note: Please see notebook for all 16 topics (8 from each corpora).

As an example, we plot word clouds (Fig. 6) to observe the most frequent words for the ‘FBI investigation on emails from Clinton’ topic for the fake news and ‘Industry - Regulation in the US’ topic for the true news.

We see some overlap of the topics for the fake news and the true news articles, for example the ‘US elections’ topic for true and the ‘US elections between Trump and Clinton’ are clearly very similar. In addition, we can consider that the topics ‘Refugees’ for the true news and ‘Syrian war’ for fake news are overlapping, as well as ‘Industry - Regulation in the US’ for true news and ‘Trump’ for the fake news.

However, we see that some themes are more unique to each of the datasets. For instance the ‘Iran-Kurdish’, and ‘China, Korea and Foreign affairs’ themes for the true articles, and ‘Racism in the US’ for the fake articles. These results consolidate some of the initial findings in the earlier more basic polarisation analysis.

Furthermore, the dashboard also allows us to see that there is a sharp clustering in the case of true news, where most of the topics within the true news dataset are evenly distributed and separated. As for fake news, they are less separated and few topics dominate the distribution with “Police Shootings” and “Trump” taking 40% of the share.

It should be noted that these conclusions are stochastic in nature, therefore could vary slightly with new runs.

Classification

With the growing use of neural networks in the machine learning space to solve prediction tasks, we decided to use them in this exercise to design a network which could learn and predict whether a news article is fake news or not. We implement a specific type of neural network layer called an ‘embedding layer’, which is very commonly used in natural language processing tasks.

A ‘word embedding’ is a class of approaches for representing words and documents using dense vectors. It has benefits over the ‘bag-of-words’ encoding models, where large sparse vectors are used to represent each word, or to score each word within a vector. Instead, embeddings represent words in dense vectors, where the vector represents the projection of the word into a continuous vector space. This method also allows semantic similarity of words to be captured, something which the bag-of-words methods do not.

In our implementation, we train an embedding neural network with our labelled dataset and attempt to predict the correct class for the article (fake news or not). We first focus our attention on those examples that are predicted incorrectly by the model, and afterwards on those articles marked as fake news, to try to intuitively and formally assess what the factors are that define fake news on this dataset.

After our initial training and testing exercise, we further test our trained model on two new scenarios. Firstly, we test it on legitimate *current* news articles, to observe how the model performs on articles outside of the date range it was trained on. Secondly, we test it on a large set of articles from a variety sources that are in or close to the trained date range, and measure how our model ranks each source. We discuss the details in the following sections.

Dataset

To train the neural network, we continue to use the same Kaggle dataset, which contains an almost even spread of fake and true articles, using 30% of this data to validate our training.

For the current news test, we scrape data from the front page CNN website using the beautiful soup package in python, producing around 100 articles at the day of execution. Whilst the dataset is small, we aim to assess the model's ability to identify fake news on a time period that is not the one it was trained on.

For the second test, we use a large dataset of unlabeled articles from different news sources from the same time period as the training dataset. The dataset contains around 144k articles from various sources: New York Times, Breitbart, CNN, Business Insider, Atlantic, Fox News, Talking Points Memo, Buzzfeed News, National Review, New York Post, Guardian, NPR, Reuters, Vox, Washington Post. This second unsupervised exercise is to assess how the model performs at classification for each source and whether any biases are present.

The preprocessing for the classification part is minimal, as by nature the embedding approach used for prediction is not as affected as much as a bag of words approach would. Therefore, we tokenize the articles and only remove punctuation, leaving in stopwords and numbers, unlike our pre-processing for the earlier analysis.

Network architecture

Our neural networks uses an embedding layer followed by two linear layers. This simple configuration should be sufficient as for a binary classification (fake or true). The architecture is fully described in Fig. 7, including a global average pooling layer that reduces our initial embedding layer input and a dropout layer, which prevents our model from overfitting by excluding 10% of the learned parameters on each iteration.

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 embedding_1 (Embedding)      (None, 500, 32)           320000

 global_average_pooling1d_1   (None, 32)               0
 (GlobalAveragePooling1D)

 dense_2 (Dense)              (None, 128)               4224

 dropout_1 (Dropout)          (None, 128)               0

 dense_3 (Dense)              (None, 1)                 129

=================================================================
Total params: 324,353
Trainable params: 324,353
Non-trainable params: 0
_________________________________________________________________
```

Fig. 7 - Description of the Neural Network embedding layer steps

Parameters

The parameters we use for the embedding network are the following:

```
embedding_dim = 32       # Embedding hidden units size
vocab_size = 10000       # Vocabulary size to use for the full embedding
padding_type = "post"    # Padding type for embedding
trunc_type = "post"      # Truncate type for embedding
max_length = 2500        # Max length of embedding sequence
```

The embedding dimension parameter is the number of units the embedding layer will contain. Usually this parameter is the most obscure in neural networks configuration, but in general terms it we could say the more units we provide the more we can resemble our training data (with the risk of overfitting).

The vocabulary size is the size we would like to consider for our embedding “container”. Our overall vocabulary from the training set is much larger, but setting this parameter to 10,000 will pick only the 10,000 most frequently used terms to create the matrix on each embedding sequence.

The padding and truncation type parameters express how to fill the gaps when the text in a document is shorter or larger than the max length of the sequence. As indicated by the value passed we will pad or truncate the embedding sequence in the posterior positions of the sequence.

Finally, the max_length (aka embedding size) parameter refers to the maximum length of each embedding sequence, i.e. for each term in the embedding vocabulary, the model will map 2500 positions of term representations using real numbers. This parameter was chosen based on the average number of words in all the articles so that as much meaning as possible is extracted without exceeding computational power.

Results

When applying to our test dataset, we obtain a prediction accuracy of over 99% (Fig. 8).

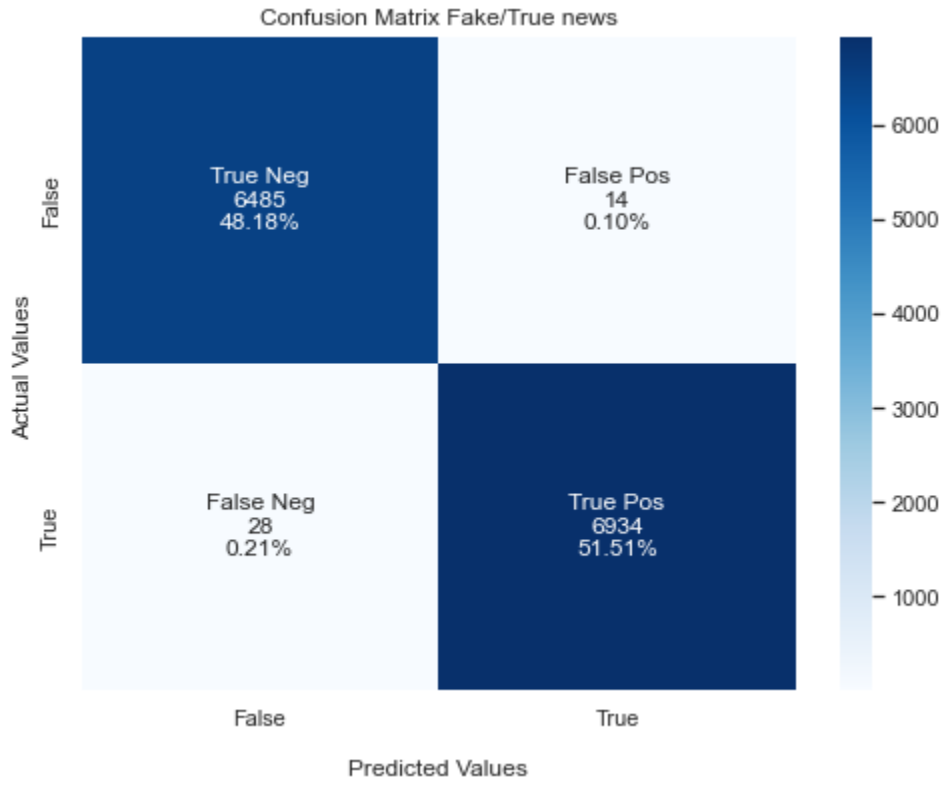


Fig. 8 - Confusion matrix for neural network fake news classifier results

Apart from the high accuracy, we can also observe that there is a greater chance of the model predicting the article as true when it is not (false negatives), as compared to predicting it as fake when it is true (false positives). In our notebook we analyse in more detail some of those cases.

The network's learning progresses the results are also positive - there are few signs of overfitting and the relatively simple embedding network performs well from very early epochs as we can see in Fig 9. Hence overall, the embedding network is accurately able to classify a binary class with little computational power, using the labels of the training dataset.

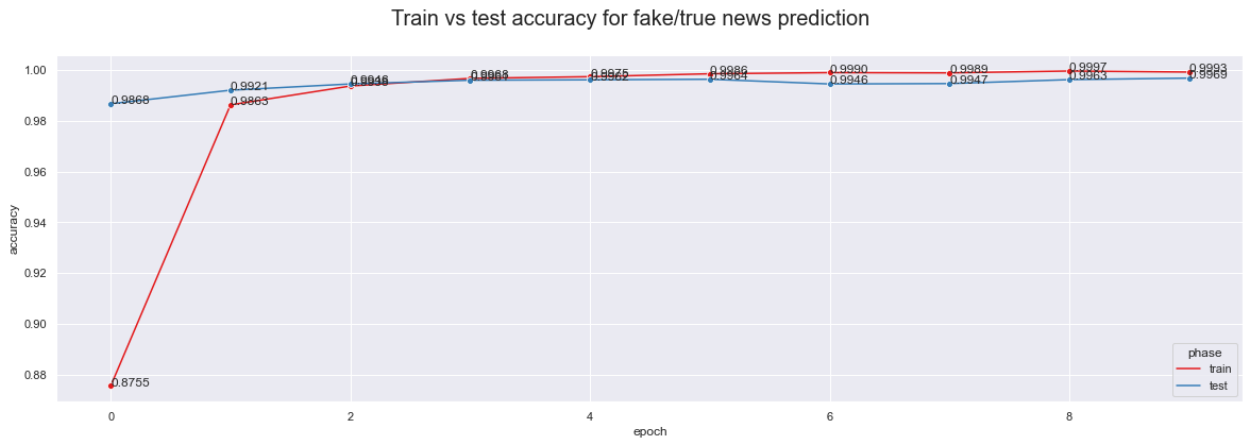


Fig. 9 - Model accuracy for neural network fake news classifier

To test the validity we expose our model to new prediction scenarios which are not necessarily subject to the original training. However, it is important to note that these new datasets are not labelled and so the accuracy of results cannot be fully measured, so for the purpose of this exercise we will use only intuitive tools to assess the results.

For our first testing scenario (using current CNN news), the model predicts 30% of the articles as fake news. This is a very high percentage for a fairly trusted news website that is not known for spreading fake news.
Therefore, our intuitive conclusion is that the time frame on which the model has been trained is extremely significant and a model trained with news articles from 6 years ago is not accurate to determine what is fake news in a more current time frame.

For our second testing scenario (using a variety of news sources over a similar timeframe to the training set), we obtain the percentage of articles that are labelled as fake news by the model. The results are summarised in Fig. 10 below.

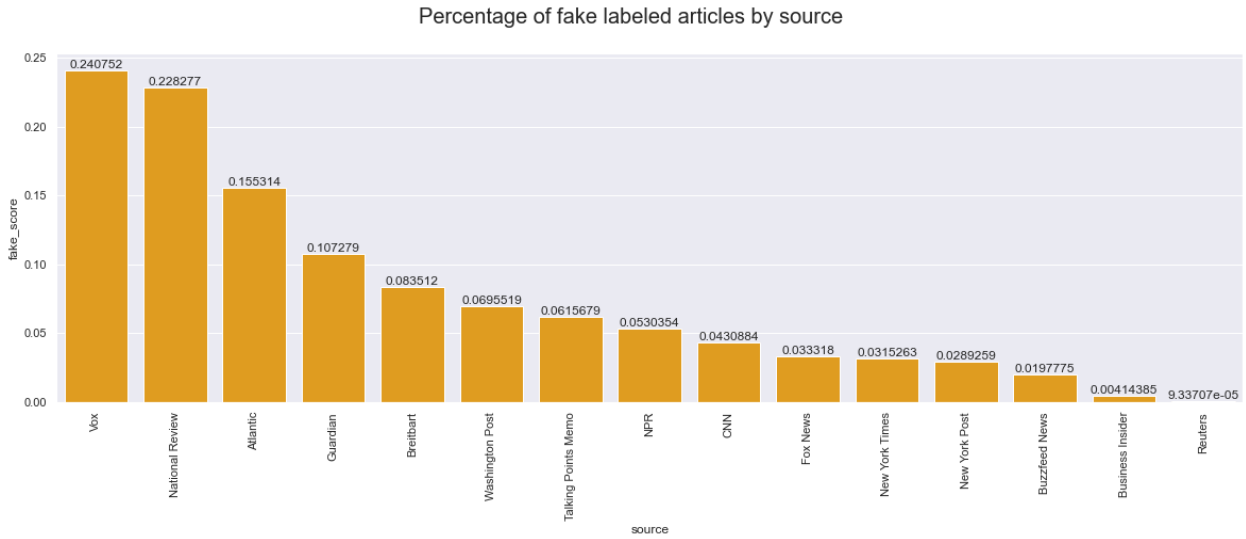


Fig. 10 - Percentage of fake news articles by source

The model predicts the highest percentage of fake news articles within Vox and National Review, at 24% and 22% respectively, whilst the lowest percentages are Reuters and Business Insider respectively with percentages under 0.3%.

Given the data is not labelled, it is difficult to estimate the accuracy of the predictions, however we run an informal validation based on data from Adfonts Media (2022) (see Appendix A.1 for visual aid) which is a metric that ranks sources by veracity, and the results appear to naively match these scores.
Furthermore, the results are consistent with the source of our train dataset, as the true news articles are mostly sourced from Reuters, so it is intuitive that the model has learned to qualify such sources as non-fake news.
It is worth noting that for less politically inclined news sources (e.g. Business Insider, which is not included in the political veracity ranking), the model identifies a very low rate of fake news, which could be due to a difference in language and vocabulary rather than the news source containing low rates of fake news.

Conclusions

In the exploratory analysis of the dataset, even after applying inverse document frequency scores, we find that there are some themes that overlap heavily in the fake and true news articles, making it difficult to observe differences between the datasets. By removing these and applying a polarisation measure to each term, we identify key terms that could help distinguish true news from fake. In addition, we see that polarisation in our particular dataset increased around the time period of the election.

When implementing Topic Modelling, we find the results consolidate the initial findings from the more basic polarisation analysis. Some topics overlap both the fake and the true datasets

(specifically around the US elections), whereas others remain specific to the true news (e.g. foreign affairs) and specific to the fake news (e.g. racism).
We also observe that topics within the true news are evenly distributed and separated, whereas for the fake news, they are less separated and few topics dominate the distribution.

For the classification task, we confirm the effectiveness of word embeddings on neural networks, obtaining high accuracy scores with a relatively simple architecture and without extensive computational time invested.

One key conclusion in the classification task of fake and true news is the importance of considering the time frame of the training data, otherwise as news themes and topics change, the predictive accuracy can greatly decrease. In addition, the content of the articles that the model is trained on needs to be diversified, or it risks overfitting to particular writing styles and specific themes.

Further iterations of this work could be to increase the classification difficulty by adding multiple classes (i.e. partially fake, partially true) to the veracity of the articles or using pre-trained models like GloVe or BeRT to aid in more complex classification tasks.
In addition, an ideal model should be constantly re-trained and adjusted to be able to keep up with the change in meaning on news articles topics.
Finally, further features could be included into the training model to increase predictive power, such as text from the article title (for which we found differences in capital letter usage), network metadata (i.e. information about the news source itself) or data on images included in the articles.

Bibliography

Adfonts Media, 2022. *Interactive Media Bias Chart.* [Online]. [Accessed 29 March 2022]. Available from: https://adfontesmedia.com/interactive-media-bias-chart/

Ahmed, S., et al., 2020. Development of fake news model using machine learning through natural language processing. Available from: https://arxiv.org/pdf/2201.07489.pdf

D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," the Journal of machine Learning research, vol. 3, pp. 993–1022, 2003. Available from: https://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf?ref=https://githubhelp.com

D. M. Blei and J. D. Lafferty, "A correlated topic model of science," The annals of applied statistics, vol. 1, no. 1, pp. 17–35, 2007. Available from: https://projecteuclid.org/journals/annals-of-applied-statistics/volume-1/issue-1/A-correlated-topic-model-of-Science/10.1214/07-AOAS114.pdf

Jin, Z., et al, 2017. "Novel Visual and Statistical Image Features for Microblogs News Verification," in IEEE Transactions on Multimedia, vol. 19, no. 3, pp. 598-608. Available from: https://ieeexplore.ieee.org/document/7589045

Kaggle. 2017. *Fake and Real News Dataset*. [Online]. [Accessed 23 March 2022]. Available from: https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset?select=True.csv

Mikolov, T., et al., 2013. Efficient estimation of word representations in vector space. Available from: https://arxiv.org/pdf/1301.3781.pdf

Reddy H. et al. (2020). Text-mining-based Fake News Detection Using Ensemble Methods. International Journal of Automation and Computing, vol. 17, no. 2, pp. 210-221. Available from: https://www.mi-research.net/en/article/doi/10.1007/s11633-019-1216-5

Rubin, V.L., 2017. Deception detection and rumor debunking for social media. In *The SAGE handbook of social media research methods* (p. 342). Sage. Available from: https://core.ac.uk/download/pdf/61692768.pdf

Shu, K. et al, 2017. Fake news detection on social media: A data mining perspective. *ACM SIGKDD explorations newsletter*, *19*(1), pp.22-36. Available from: https://arxiv.org/pdf/1708.01967.pdf

Shrestha, M., 2018. Detecting Fake News with Sentiment Analysis and Network Metadata. *Earlham college, Richmond*. Available from: https://portfolios.cs.earlham.edu/wp-content/uploads/2018/12/Fake_News_Capstone.pdf

Stahl, K., 2018. Fake news detection in social media. *California State University Stanislaus*, *6*, pp.4-15. Available from:

https://www.csustan.edu/sites/default/files/groups/University%20Honors%20Program/Journals/02_stahl.pdf

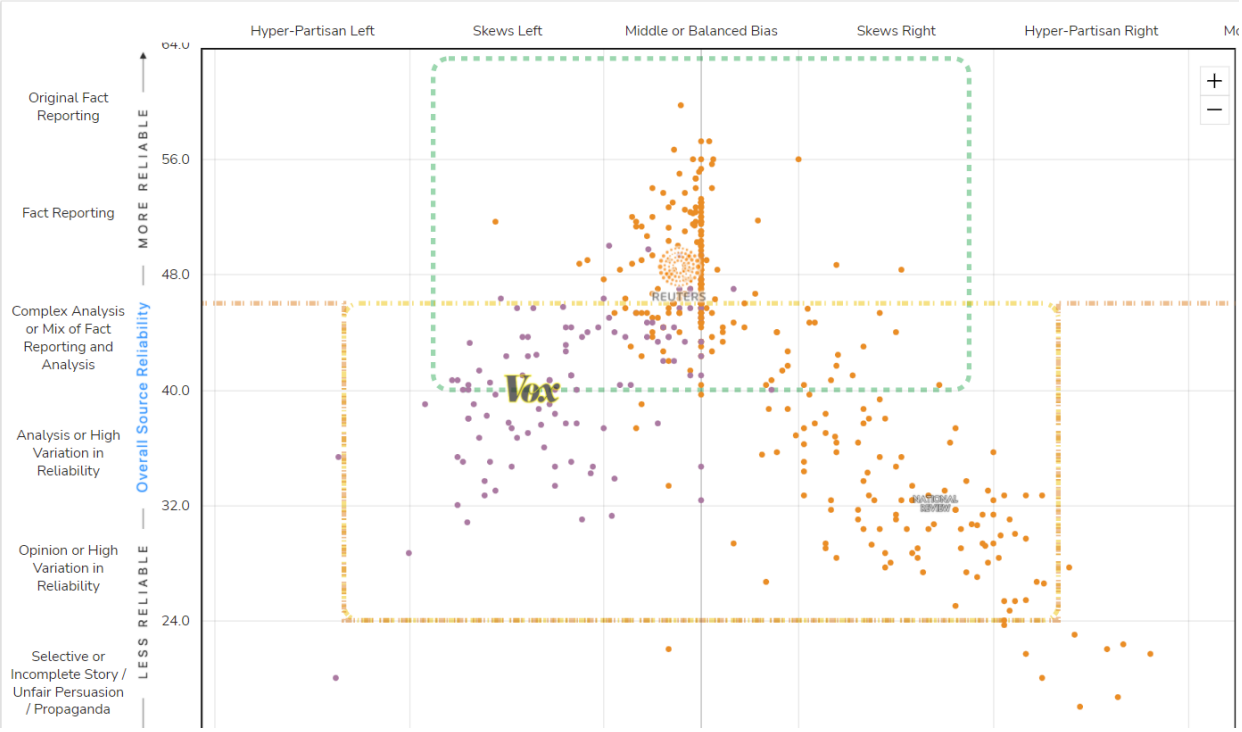
Wang W. (2017). "Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection. 422-426. Available from: https://www.researchgate.net/publication/318740546_Liar_Liar_Pants_on_Fire_A_New_Benchmark_Dataset_for_Fake_News_Detection

Yang, Y. et al., 2018. TI-CNN: Convolutional neural networks for fake news detection. Available from: https://arxiv.org/pdf/1806.00749.pdf

Yale Law School. 2017. *Fighting Fake News Workshop Report.* [Online]. [Accessed 30 March 2022]. Available from: https://law.yale.edu/sites/default/files/area/center/isp/documents/fighting_fake_news_-_workshop_report.pdf

Zhou, X. et al., 2020. A survey of fake news: Fundamental theories, detection methods, and opportunities. *ACM Computing Surveys (CSUR)*, *53*(5), pp.1-40. Available from: https://arxiv.org/pdf/1812.00315.pdf

Appendix



A.1. Biased and veracity estimation for Vox, National Insider and Reuters performed by Ad Fontes Media https://adfontesmedia.com/interactive-media-bias-chart/,

A.2: Interactive dashboard instructions:

This dashboard is based on the pyLDAvis library, and can be interpreted as:

Left side panel:

- It offers a representation of how topics are distributed in the 2-dimensional space (based on PCA). The larger the bubble, the more frequent the topic in the documents is.

  Typically, topic models with low numbers of topics have large bubbles than tend to not overlap. Topic models with high numbers of topics, on the other hand, have many overlapping small size bubbles.

- Intertopic Distance is an approximation of semantic relationship between the topics. Those that share many terms will be closer and tend to overlap.

Horizontal Bar Graph:

- The blue bars in the graph show the frequency distribution of the words in all of the documents.
- The red portion of the bars describe the frequency of each word, given a topic.

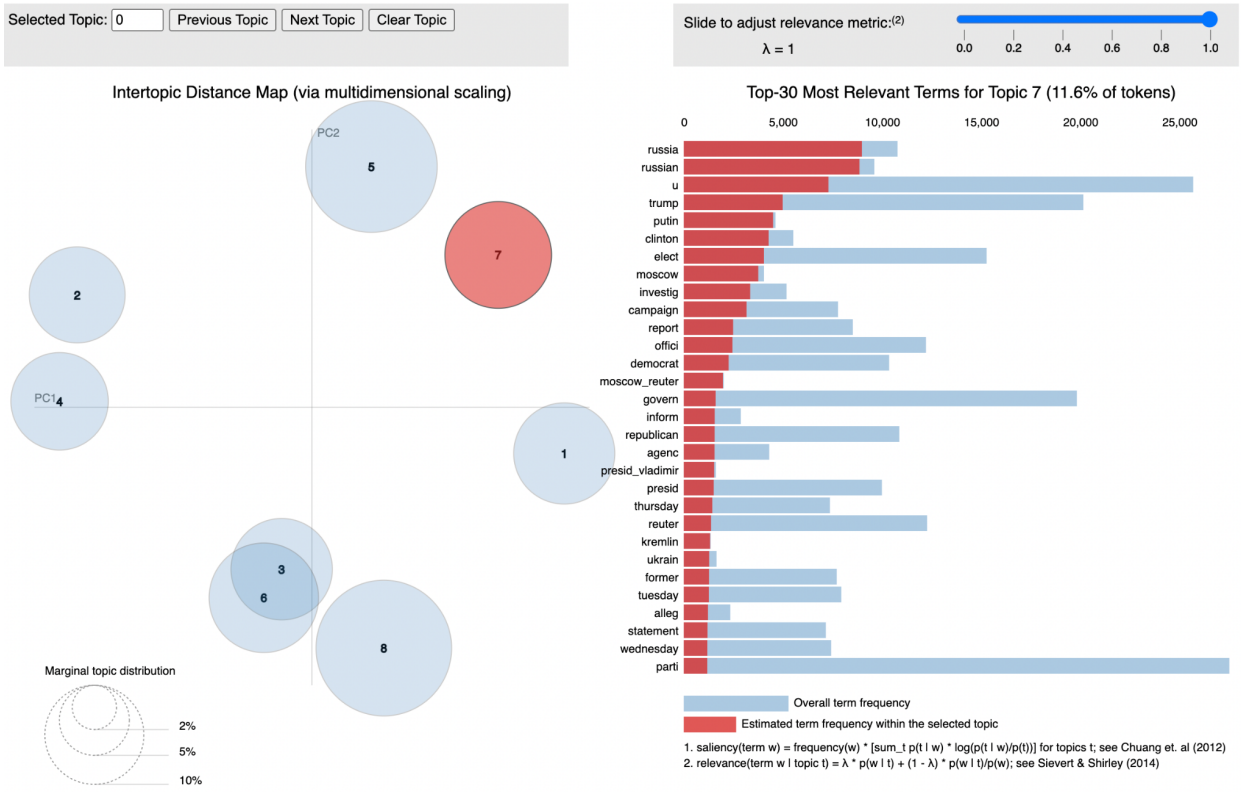
Interactivity:

- You can select a topic by clicking on a topic bubble. This will show the top 30 words associated with it, and their proportion in terms of total frequency.

- Hovering over the specific words, the topics containing those words are highlighted. The higher the proportion of that word in a topic, the larger the size of the bubble.

- You can play with the lambda parameter to re-rank words in topics based on their frequency. Decreasing the lambda parameter, increases the weight of the ratio of the frequency of word given the topic / Overall frequency of the word in the documents.

We attach two html files (one for each topic model) with the fully functional dashboards for in detail analysis of the topics.



Interactive visualisations of how the topics (tokens) are distributed in the corpora (topics). Please see html files to interactively explore the results of our LDA models