SMU Data Science Review

Volume 1 | Number 3 Article 10

2018

Fake News Detection: A Deep Learning Approach

Aswini Thota
Southern Methodist University, athota@smu.edu

Priyanka Tilak
Southern Methodist University, ptilak@smu.edu

Simrat Ahluwalia
Southern Methodist University, simeratjeeta@smu.edu

Nibrat Lohia
Southern Methodist University, nlohia@smu.edu

Fake News Detection:
A Deep Learning Approach

Aswini Thota[1], Priyanka Tilak[1], Simeratjeet Ahluwalia[1], Nibhrat Lohia[1]

[1] 6425 Boaz Lane, Dallas, TX 75205
{AThota, PTilak, simeratjeeta, NLohia}@SMU.edu

Abstract Fake news is defined as a made-up story with an intention to deceive or to mislead. In this paper we present the solution to the task of fake news detection by using Deep Learning architectures. Gartner research [1] predicts that "By 2022, most people in mature economies will consume more false information than true information". The exponential increase in production and distribution of inaccurate news presents an immediate need for automatically tagging and detecting such twisted news articles. However, automated detection of fake news is a hard task to accomplish as it requires the model to understand nuances in natural language. Moreover, majority of the existing fake news detection models treat the problem at hand as a binary classification task, which limits model's ability to understand how related or unrelated the reported news is when compared to the real news. To address these gaps, we present neural network architecture to accurately predict the stance between a given pair of headline and article body. Our model outperforms existing model architectures by 2.5% and we are able to achieve an accuracy of 94.21% on test data.

1 Introduction

"Fake News" is a term used to represent fabricated news or propaganda comprising misinformation communicated through traditional media channels like print, and television as well as non-traditional media channels like social media. The general motive to spread such news is to mislead the readers, damage reputation of any entity, or to gain from sensationalism. It is seen as one of the greatest threats to democracy, free debate, and the Western order [3].

Fake news is increasingly being shared via social media platforms like Twitter and Facebook [2]. These platforms offer a setting for the general population to share their opinions and views in a raw and un-edited fashion. Some news articles hosted or shared on the social media platforms have more views compared to direct views from the media outlets' platform. Research that studied the velocity of fake news concluded that tweets containing false information reach people on Twitter six times faster than truthful tweets [3]. The adverse effects of inaccurate news range from making people believe that Hillary Clinton had an alien baby, trying to convince readers that President Trump is trying to abolish first amendment to mob killings in India due to a false rumor propagated in WhatsApp.

Technologies such as Artificial Intelligence (AI) and Natural Language Processing (NLP) tools offer great promise for researchers to build systems which could automatically detect fake news. However, detecting fake news is a challenging task to accomplish as it requires models to summarize the news and compare it to the actual news in order to classify it as fake. Moreover, the task of comparing proposed news with the original news itself is a daunting task as its highly subjective and opinionated.

A different way to detect fake news is through stance detection which will be the focus of our study. Stance Detection is the process of automatically detecting the relationship between two pieces of text. In this study, we explore ways to predict the stance, given a news article and news headline pair. Depending on how similar the news article content and headlines are, the stances between them can be defined as 'agree', 'disagree', 'discuss' or 'unrelated'. We experimented with several traditional machine learning models to set a baseline and then compare results to the state-of-the-art deep networks to classify the stance between article body and headline.

Through experimental procedures, we propose a model which can detect fake news by accurately predicting stance between the headline and the news article. We also studied how different hyperparameters affect the model performance and summarized the details for future work. Our model performs reasonably well when classifying between all the stances with some variations in accuracy for disagreed stances.

Further, in Section 2, we have discussed problems with defining and identifying fake news, describe Fake News Challenge data set which we used to perform the experiment, and we discuss the previous work performed on similar problem. In section 3, we provide a primer on various techniques used in our experiments such as natural language and deep learning. In section 4 we explain the experimental design followed by our solution to solve the fake news detection problem in Section 5. In section 6, we present our results for different models and hyper parameter tuning for the models and we further conclude our findings in Section 7.

2 Fake News

Fake news can be come in many forms, including: unintentional errors committed by news aggregators, outright false stories, or the stories which are developed to mislead and influence reader's opinion. While fake news may have multiple forms, the effect that it can have on people, government and organizations may generally be negative since it differs from the facts.

2.1 The Problem of Defining Fake news

The very task of defining fake news is a challenge in itself and is open to interpretations. Table 1 has the prevalent definitions of fake news:

Table 1. Definitions for fake news

Definition statement
A made-up story with an intention to deceive [14]
News articles that are intentionally and verifiably false, and could mislead readers [2]
Fake news is a type of yellow journalism or propaganda that consists of deliberate misinformation or hoaxes spread via traditional print and broadcast news media or online social media [15].

Misinformation is a common theme when fake news is mentioned. Misinformation can itself be classified as shown in Table. 2.

Table 2. Misinformation Matrix

	Satire or Parody	False connection	Misleading content	False context	Imposter content	Manipul-ated content	Fabricated content
Poor Journalism		✓	✓	✓			
To parody	✓				✓		✓
To Provoke or to 'punk'					✓	✓	✓
Passion				✓			
Partisanship			✓	✓			
Profit		✓			✓		✓
Political Influence			✓	✓		✓	✓
Propaganda			✓	✓	✓	✓	✓

Considering these difficulties involved in detection of fake news, a good first step is to detect the stance between the body of text and the entity it's describing. The task of stance detection can be described as the process of automatically predicting if the news article or social media content is agreeing, disagreeing or unrelated to the entity it's describing. The below example in Table 3, explains stance between news headlines and news article.

Table 3. Example of News Article and Headlines and their corresponding stance

Article body	Headline	Stance
Awwwww! this is such a heartwarming story! when Youtuber Josh Paler Lin gave $100 dollars to a homeless man, he hoped he could videotape the man wasting the money away on alcohol. but all josh got was a shock, because Thomas, the homeless man, went into a liquor store, but came out with enough food to support everyone around him! [related: 12 heartwarming viral stories!] with the hundred dollars, thomas chose to help as many other homeless people he could find in an anaheim park while also moving Josh in the process. Josh immediately felt guilty for assuming worst of Thomas, and on top of giving him an extra $100 on the spot, has already raised over $44,000… and this video only went live on monday! ch-ch-check out the inspiring clip (below)!!!	Prankster gives homeless man $100, secretly follows him and learns he buys food for others	Agree
	Homeless man who received a meal and hotel stay after spending $100 donation on food for friends insists it was not a hoax – despite new witness insisting there was no way he could have been secretly filmed	Discuss
	Was heart-tugging viral video of generous homeless man all a hoax?	Disagree
	Resetting ios 8 preferences may wipe icloud files	Unrelated

2.2 The Problem of Detecting Fake News

The core task of detecting fake news involves identifying the language (set of words or sentences) which is used to deceive the readers. The idea of classifying fake news by learning word-level meaning is a very challenging task under the skin. For instance, consider Table 4.

Table 4. Examples of fake news [13]

Type of Fake news	Example
100% False	*#RIP Paul McCartney*
Slanted and Biased	News from Outlet A: *Climate change will produce more storms like Hurricane Katrina.* News from Outlet B: *climate change can lead to major hurricanes→ there haven't been major hurricanes→ Climate change isn't real*

Misusing the Data	*"Have a Beer, It's Good for Your Brain,"* reported Inc. But you should wait a minute before you grab a pint (or two). The study was done on mice — not people. And the amount of beer was the equivalent of 28 kegs in humans.
Imprecise and Sloppy	*"1 in 5 CEOs are Psychopaths, Study Finds."* But the headline is wrong. The research was based on a survey of professionals in the supply chain industry, not CEOs.

The above examples capture the complex nature of detection and classification of fake news. To rightly classify the above types of fake news, our language model needs to understand the subtleties involved in conveying messages through text.

Detecting fake news is hard for many reasons. First, manual task of identifying fake news is very subjective. Assessing the veracity of a news story is a complex and cumbersome task, even for trained experts. News is not only spread through traditional media outlets anymore but also through various social media channels. Automated solution requires understanding the natural language processing which is difficult and complex. These complexities make it a daunting task to classify text as fake news.

The combination of above issues of definition and detection makes the task of stance detection to solve the automatic fake news classification challenging.

2.3 Dataset Description: Fake News Challenge (FNC-1) Data

Fake News Challenge [7] opened to the public on December 1, 2016 as a competition. The data used for this competition were derived from the Emergent Dataset created by Craig Silverman. Emergent Research [8] is a research and consulting firm focused on the most dynamic sector of the global economy. The specific dataset of Emergent which is used for FNC-I challenge is a digital journalism project for rumor debunking.

Dataset includes body of the news article, the headline of the news article, and the label for relatedness (stance) of an article and headline. The data set is split into train, validation and test splits based on the methodologies outlined in Section 4. The data is released in 2 sections which give information of 1684 news articles and 49,973 unique pairs of news article and headlines along with its stances. Stance has four categories: Unrelated, Disagree, Agree and Discuss.

As an initial stage towards building an automated solution for identifying fake news, FNC-I targeted to solve the stance detection problem. The aim of solving stance detection is to determine the relatedness of news article and headline of the news article. Distribution across 4 stances is presented in table 5.

Table 5. Stance labels in FNC-1 training dataset

Stance category	Percentage	Description
Agree	7.36%	Headline agrees with the claim made in the news article

Disagree	1.68%	Headline disagrees with the claim made in the news article
Discuss	17.82%	Headline discusses same topic as news article but is not exactly agree
Unrelated	73.13%	Headline does not discuss same topic as news article

2.4 Review of Related Work on Stance Detection

In this section we discuss related work that was conducted in fake news detection and stance classification using FNC -1 data.

In the following Table 6 we discuss the models which achieved best accuracy on FNC-1 data. We limited our literature research to the published manuscripts which used FNC-1 data for their experiments and we use ‘accuracy’ as the evaluation metric for comparison.

Table 6. Top two models on FNC-1 dataset

Team Name	Overall accuracy	Model Description
Riedel et al.	88 %	A model with lexical and similarity features passed through a multi-layer perceptron (MLP) with one hidden layer
Davis et al.	93 %	A bag-of-words followed by a three-layer multi-layer perceptron (BoW MLP)

The top 2 teams used a variation of neural network architectures in their model.

3 Primer on Natural Language Processing and Deep Learning

In this section, we will discuss the relevant technologies that form the basis of the solution. Knowledge of these algorithms will help understand the paper in a better way.

3.1 Introduction to Natural Language Processing

Our news article data and headlines are in text format. Building automated machine learning models on text data involves cleaning and converting textual data to machine readable format. Natural-language processing (NLP) is an area of computer science and artificial intelligence concerned with the interactions between computers and human (natural) languages, and how to program computers to fruitfully process large amounts of natural language data [19]. In this project, we are leveraging textual data from FNC-1.

Natural language processing (NLP) models have made significant progress in automatically detecting sentiment and in mining opinion from text. A wide variety of benchmark datasets and techniques have emerged due to significant research conducted in this space [11]. Most of the pre-neural network era NLP techniques focused on developing extensive domain specific features. These techniques often involved manual feature engineering and they require linguists and semantic experts to parse and curate the text. However, the NLP landscape has evolved at great pace. Collobert et al. (2011) [12] introduced Natural Language Processing nearly from scratch which introduces unified neural network architecture and algorithms that can be applied to various NLP tasks. This paper [12] describes how we can learn word and sentence level representations instead of exploiting man-made input features carefully optimized for each task. This breakthrough research paved way for researchers to represent words and sentences as vectors, which can understand the context in which they are being used.

3.1.1 Word Vector Representation

The method of representing words as vectors is commonly referred to as word vector representations or word embeddings. In this paper, we experiment with the below mentioned word vector representations:

Bag of Words: In a Bag of Words (BoW) model, sentences are represented as multiset of its words. BoW model disregards the order and hence it also disregards the context in which the word occurred.

Tf-Idf: Tf-idf stands for term frequency-inverse document frequency. Tf-idf weight gives us an indication on how important a given word is for a sentence or document in relation to the entire corpus.

GloVe: Glove representation are learnt by first constructing a co-occurrence matrix of all the words in the corpus and the dimensions are then reduced by matrix factorization methods. Researchers have trained and created GloVe models on multiple corpora and made them available to the public. The details can be found at [20].

Word2Vec: Word2Vec is a predictive model as the model is designed to predict the word given a context window.

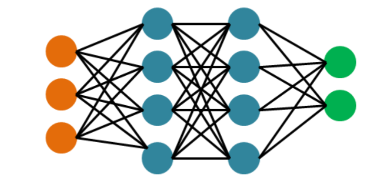
3.2 Introduction to Neural Network Architectures:

Given the significant breakthrough in neural network research, we used various versions of deep neural nets detailed in further sections. In this section we discuss different versions of neural network architectures used in common research. A high-level summary and working details of neural network architectures are listed in Table 7.

Table 7. Summary of Neural Network architectures

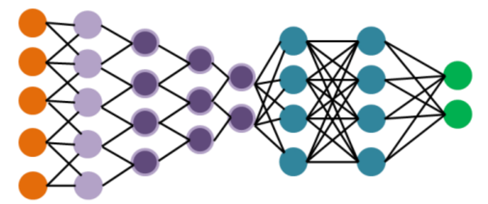
Model Name	Model Description

Dense Neural Network (DNN)



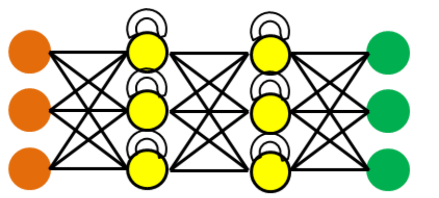
The fully connected dense neural network allows us to pass the input as sequence of words. The layered architecture allows us to experiment with the right depth that is needed for our task. The network consists of an input layer, an output layer and can consist of series of hidden layers.

Convolution Neural Network (CNN)



Convolutional Neural Networks (CNN) are very similar to ordinary Neural Networks: they are made up of neurons that have learnable weights and biases. A CNN consists of an input and an output layer, as well as multiple hidden layers. The hidden layers of a CNN typically consist of convolutional layers, pooling layers, fully connected layers and normalization layers

Recurrent Neural Network (RNN)



Recurrent neural networks (RNN) are popular with sequential data as each unit can have memory about the state of previous unit. This is particularly useful in natural language processing because it helps gain deeper understanding of language. RNN's have an input layer, output layer and can have a number of hidden recurrent units which have memory gates.

4 EXPERIMENTAL DESIGN AND SOLUTION

In this section, we provide details into the data preprocessing, word vector representations and our sampling design.

4.1 Data Preprocessing

Text data requires special preprocessing to implement machine learning or deep learning algorithms on them. There are various techniques widely used to convert text data into a form that is ready for modeling. The data preprocessing steps that we outline below are applied to both the headlines and the news articles. We also provide insights into different word vectors representations we used as part of our analysis.

4.1.1 Stop Word Removal

We start with removing stop words from the text data available. Stops Words (most common words in a language which do not provide much context) can be processed and filtered from the text as they are more common and hold less useful information. Stop words acts more like a connecting part of the sentences, for example, conjunctions like "and", "or" and "but", prepositions like "of", "in", "from", "to", etc. and the articles "a", "an", and "the". Such stop words which are of less importance may take up valuable processing time, and hence removing stop words as a part of data preprocessing is a key first step in natural language processing. We used Natural Language Toolkit – (NLTK) library to remove stop word.
Figure 2 illustrates an example of stop word removal.

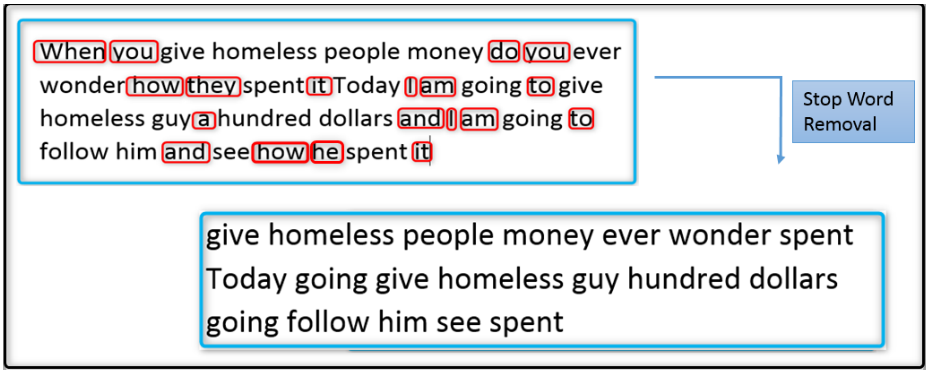


Fig. 2. Example for Stop Word removal

4.1.2 Punctuation Removal

Punctuation in natural language provides the grammatical context to the sentence. Punctuations such as a comma, might not add much value in understanding the meaning of the sentence. Figure 3 shows an example of Punctuation removal process.

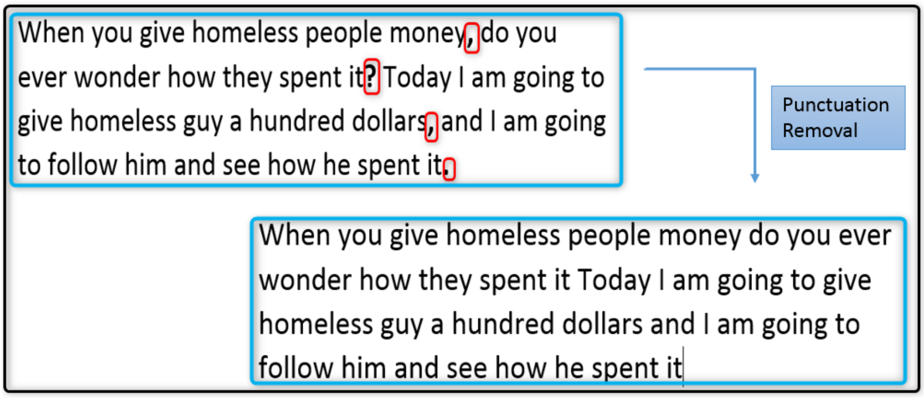


Fig. 3. Example for punctuation removal

4.1.3 Stemming

Stemming is a technique to remove prefixes and suffixes from a word, ending up with the stem. Using stemming we can reduce inflectional forms and sometimes derivationally related forms of a word to a common base form. Figure 4 shows the example of stemming technique.

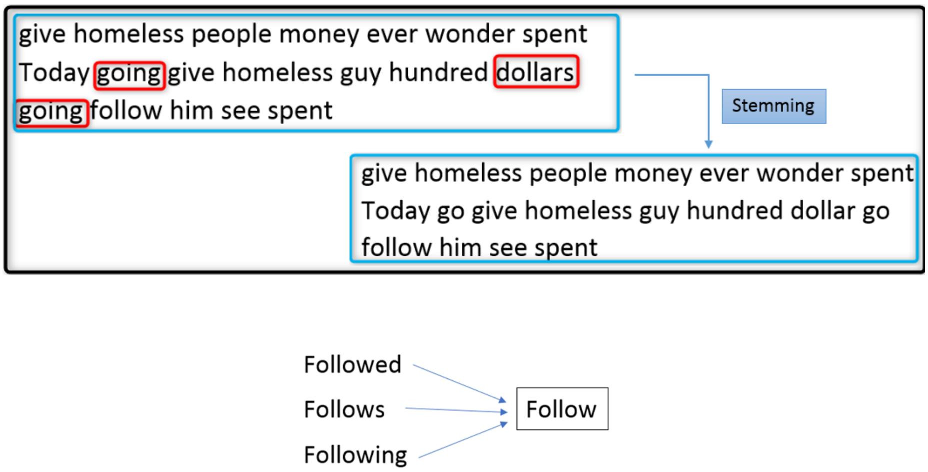


Fig. 4. Example Stemming

4.2 Word Vector Representation

Preparing the text from the body and headline of the news article for modeling is quite challenging. To perform text analytics, we need to convert raw text into numerical features. We experimented with two techniques to transform the raw text and feature extraction: Bag of Words and TF-IDF.

4.2.1 Bag of Word

The Bag of Words (BoW) technique processes each news article as a document and calculates the frequency count of each word in that document, which is further used to create numerical representation of the data, also called as vector features of fixed length.

Bag of Words converts raw text to word count vector with CountVectorizer function for feature extraction. CountVectorizer splits the text form content, builds the vocabulary and encodes the text into a vector. This encoded vector will have a count for occurrences of each word that appears more like a frequency count as a key value pair.

This methodology has drawbacks in terms of information loss. The relative position of the words is not considered, and the information about the context is lost. This loss can be expensive sometimes, compared to the gain in computing simplicity with the ease

4.2.2 TF-IDF vectorizer

We have also used the technique "Term Frequency-Inverse Document Frequency" (TF-IDF) for feature extraction. Term Frequency and Inverse Document Frequency are two components of TF-IDF. Term Frequency identifies local importance of a word by its occurrence in a document. Inverse Document Frequency identifies the signature words, which are not appeared more often across the documents

Word with a high TF-IDF is a signature word which is important for the document in consideration, has high frequency in the document but is not a common word across other texts.

4.3 Sampling Techniques

We split our data into Train, Validation and Test data sets. Because our dependent variable (Stance) has four different classes and the data is unbalanced, we used stratified proportional allocation and random shuffling to perform our data splits. We allocated 67% of our FNC data to the Train set and the remaining 33% to the Test set. The training data is further divided into validation sets (80/20 split). All of our experiments are conducted on training and validation sets in a 3-fold cross-validation setup.

5 Methods and Process

In this section, we discuss several neural network architectures that we experimented with and we present the network architecture that gave us best results.

5.1 Methods

We trained our model on three different variations of neural networks. In this sub-section, we provide a high-level summary of the network architectures. The details of these models are discussed in the later section.

5.1.1 Tf-Idf Vectors with Dense Neural Network

This model takes the Tf-Idf vectors of the headline-article pair, their cosine similarity (standard metric to measure the similarity between 2 non-zero vectors) as input and predicts the output stance. We then passed the Tf-Idf vectors to a dense neural network. The dense neural network represents the words in a hidden dimension which can then be used by the other layers in a seamless fashion. The final dense layer predicts the output probabilities of the stances. The model architecture and the model parameters are outlined in Figure 5 and details are provided in the next section.

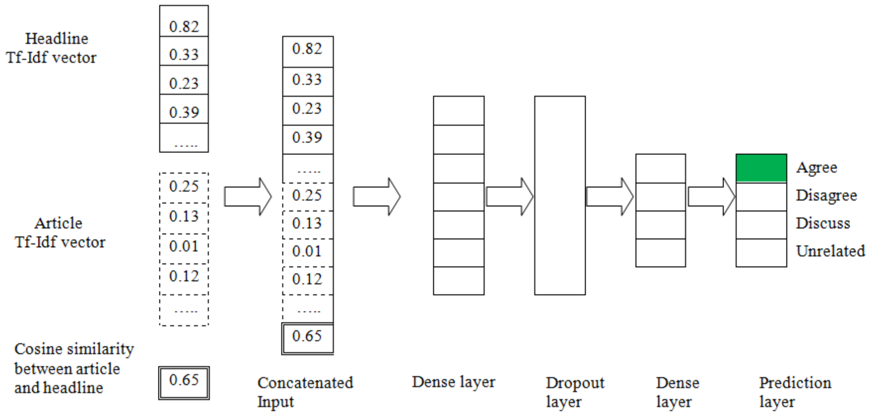


Fig. 5. Tf-Idf vector with dense neural network architecture

5.1.2 Bag of Words Vector with Dense Neural Network

The BoW with dense neural network architecture uses a simplified vector space embeddings to represent text. The BoW vectors are computed for each headline-article pair. The BoW vectors are concatenated cosine similarity measure are passed as input features to our dense neural network. The model architecture is presented in Figure 6 with details following it and results are discussed in the next section.

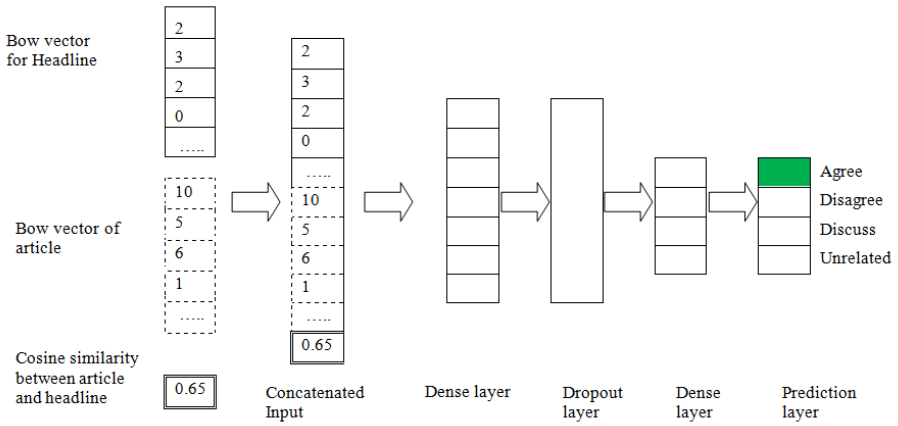


Fig. 6. Bag of Words vector with dense neural network architecture

Table. 8. Bag of Words model details

Hyperparameters	Value
Final layer activation function	Softmax
Batch size	64
Dropout rate	0.2
Epochs	100
L2 penalty	0.0001

5.1.3 Pre-trained word embedding's with Neural Networks

Pre-trained word vectors are created based on the distributional hypothesis, which states that the meaning of a word can be determined by its company [16]. The fact that natural language is temporal in nature and that neural networks are capable of handling temporal data, offers a great flexibility to represent words using pre-trained word embeddings. In this neural network architecture, we used Google's Word2Vec for representing the words in 300-dimensional vector space embeddings. The word embeddings then fed into a dense layer. The network architecture is summarized below in Figure 7 and Table 9.

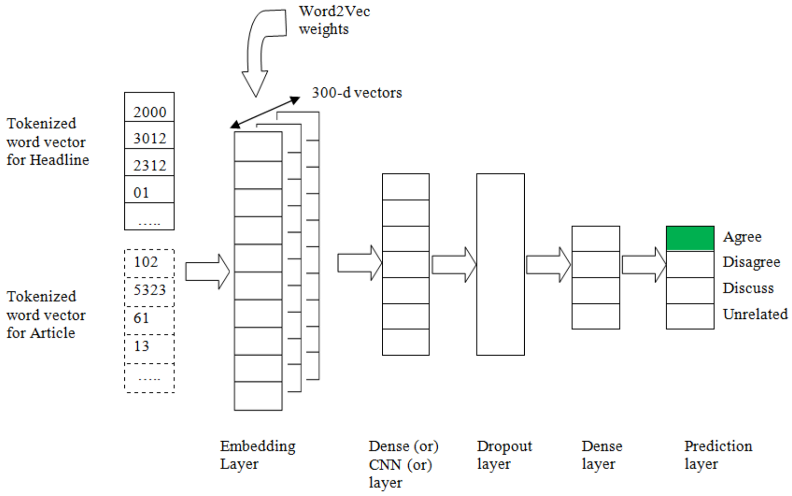


Fig.7. Pre-trained word embedding's with Neural Network Architecture

Table. 9. Pre-trained word embedding's model details

Parameter name	Value
Final layer activation function	Softmax
Batch size	64
Dropout rate	0.2
Epochs	50
L2 penalty	0.0001

5.2 Proposed Solution:

We tried with different word vector representation and neural network architectures as mentioned in the Methods sub-section (5.1). Our best performing model takes Tf-Idf vector representation of words combined with preprocessed engineered features as concatenated inputs and uses dense neural network architecture to predict the target stance.

The input features for our model consists of Tf-Idf word vector representations of article-headline pair, Cosine similarity between article-headline pairs represented using Tf-Idf, and cosine similarity between article-headline pair represented using Google's word vectors (Word2Vec). We computed Tf-Idf scores on unigrams and bigrams. To avoid bias due to unbalance in dataset, we did not consider words which appeared in more than 50% of all training documents and excluded the words which appeared in less than 50 documents.

Our model attempts to capture the relative importance of a word present in article-headline pairs locally (how important is the word for that specific headline-article pair) and globally (how common or uncommon that specific word is in relation to all the words in the corpus). In order to capture the similarity between the headline-article pair, we calculated the Cosine similarity between Headline- Article Tf-Idf pairs.

Given the size of our vocabulary and the unbalanced nature of our data, it is very easy for a neural network model to over-fit, meaning unable to predict accurately on the unseen test set. We deployed L2, dropout and early stopping as regularization techniques to overcome overfitting and improve generalization.

The neural network architectures offer a wide variety of hyperparameters, the Table 10 presents the hyperparameters we focused on.

Table 10. Hyperparameter Tuning performance

Hyperparameters	Experiment range	Choice
Final layer activation function	{ReLU, Tanh, Softmax}	Softmax
Batch size	32 – 256	64
Dropout rate	0 – 1	0.1
Epochs	50 – 200	50
L2 penalty	{0.1,0.01,0.001,0.0001}	0.0001

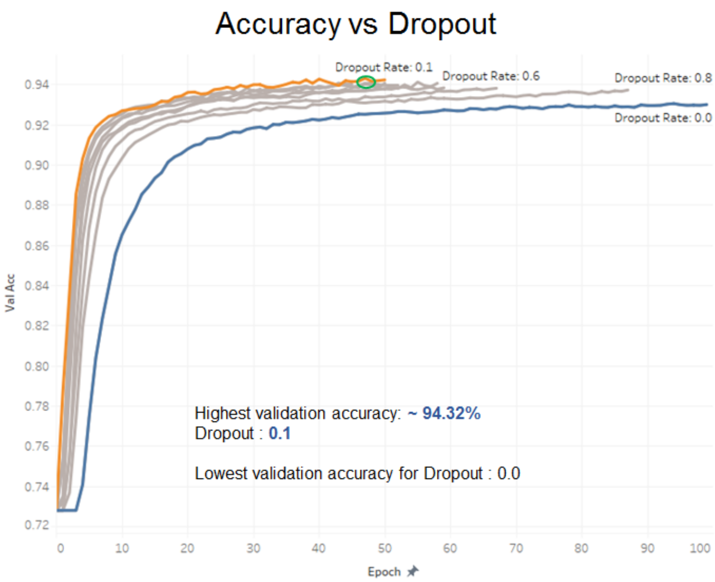


Fig. 8. Accuracy for different epochs and dropout

The Figure 8 illustrates the accuracy variation on the validation set for different epochs and dropout rates. The best validation results are achieved with a dropout rate of 0.l and with 50 epochs.

6 Results

After a thorough hyperparameter tuning on our best performing model, we evaluated the model on test data. Since our intention is to accurately measure the closeness of the predicted stance to the original, we choose 'classification accuracy' as our evaluation metric. The predicted accuracy for the models described in Section 5.1 is presented in Table 11.

Table 11. Accuracy for different Model variation

Variations	Accuracy
Tf-Idf on unigrams and bigrams with cosine similarity fed into dense neural network	**94.31%**
BoW without unigrams and bigrams with cosine similarity fed into dense neural network	89.23%
Pre-trained embeddings (Word2Vec) fed into dense neural network	75.67%

As Table 9 shows, Tf-Idf word vector representations when propagated through a dense neural network gives an accuracy of 94.21%.

Table 12 compares our model accuracy with the other models presented in the literature. Our model performs slightly better than the second best performing model.

Table 12. Comparison between best performing model and our model

Model Description	Accuracy
Tf-Idf on unigrams and bigrams with cosine similarity fed into dense neural network (our model)	**94.31%**
BoW with multilayer perceptron [6]	92.46%
BoW with cosine similarity fed into dense neural network [5]	88.46%

We recognize that the FNC-1 dataset is an unbalanced dataset. It is highly important for our model to perform reasonably well on minority stances. Table 13 summarizes the accuracy results on test data for different stance classifications.

Table 13. Prediction Accuracy for different Stances

Stance	Prediction Accuracy on Test data

Agree	73.29%
Disagree	44.38%
Discuss	88.08%
Unrelated	99.37%
Overall	**94.31 %**

The confusion matrix in Figure 9 shows the misclassification rates for various Stance types. As seen, the misclassification rate of 'disagree' class as 'agree' is high and raises concern. The cost of error can be taken into consideration for critical evaluation of such an error and is beyond the scope of this research.

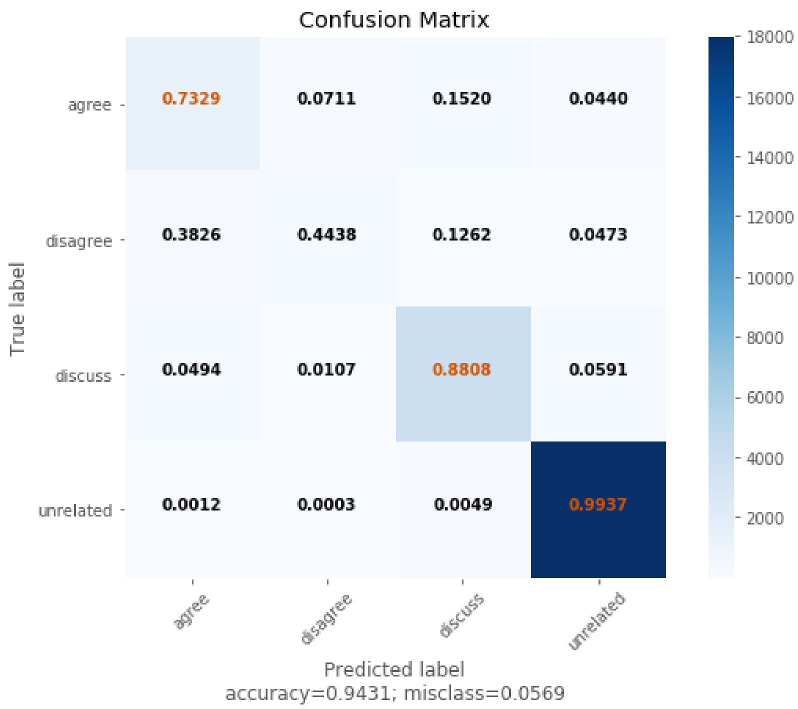


Fig. 9. Confusion Matrix for Final Model

7 Ethics

Data collected for FNC-1 [23] is taken from real life as per the definitions provided by the respective competition's organizations Fake news challenge [7]. No human subject testing was conducted as part of the research in this paper.

FNC-1 data does not include the PII of the author of the content. The identity information of the authors does not influence the outcome of the models' behavior. Thus, the models are objectively defined to be independent of any personal information of the authors or social media account holders.

The generalized models proposed are not going to be in favor of any political, social or economical organization. These models will solely be based on the relationship between the "body of text", "target entity/headline" and the labeled "stance" provided in the training datasets. The generalized models created for the solution may not understand the context in case of parodies or satirical news articles. It may classify parodies or satire as fake news.

As discussed in the previous section, the cost of misclassification is a concern and should be taken into consideration before making interpretations. Ignorance of the fact that the models are not perfectly accurate and can sometimes lead to misinterpretation of information, can alter credibility. Although attempts are made to avoid overfitting, generalization can be challenging sometimes.

The research is based on a general sample of the population of all news published around the world. Although enough care is taken to accommodate the variation of news writing styles, it cannot be ignored that the performance may differ on articles not represented accurately in the training data.

8 Conclusion

Using a finely tuned Tf-IDF – Dense neural network (DNN) model, we are able to outperform existing model architectures by 2.5% and we are able to achieve an accuracy of 94.21% on test data. Our model performs reasonably well when the stances between headline and news article are 'unrelated', 'agree' and 'discuss', but the prediction accuracy for "disagree" stance is low (44%).

The model represented using BoW- DNN is our second best performing model. It's very surprising to see that the words when represented using pre-trained word embedding's such as Word2Vec consistently yielded low accuracy scores when compared to simple Tf-Idf and BoW representations. There can be several reasons for this phenomenon including size of the news article. It is possible that the Word2Vec based latent space representations are not able to capture the word semantic level importance if the news article length is very large.

Our strategy to compute the Tf-Idf vectors based on unigrams and bigrams turned out to be very effective. Hand-crafted input such as cosine similarity between the news article and headline also proved to be a valuable feature to our model input.

We also experimented with different hyperparameters. We observed that, by deploying regularization techniques such as Dropout, L2 regularization, cross-validation and early stopping, we were able to achieve a very smooth and consistent learning process.

Finally, we want to extend this work by performing similar analysis on a completely different dataset such as Twitter and Facebook. By classifying fake news from social media platforms, we hope to get one step closer towards building an automated fake news detection platform. This study provides a baseline for the future tests and broadens scope of the solutions dealing with fake news detection. The social media data will ensure that the variations in the language are taken care of.

We would like to further dig deep and evaluate the effects of such news propagation on the readers and come up with simple techniques for faster prediction. The research can borrow qualitative models built on similar tasks by other disciplines and re-evaluate feature engineering and preprocessing techniques used.

References

1. Titcomb, J., Carson, J.: www.telegraph.co.uk. Fake news: What exactly is it – and how can you spot it?
2. Allcott, H., Gentzkow, M.: Social media and fake news in the 2016 election. Technical report, National Bureau of Economic Research (2017)
3. Langin, K.: http://www.sciencemag.org. Fake news spreads faster than true news on Twitter—thanks to people, not bots (2018)
4. Wardle, C.: Fake News. It's complicated. First Draft News (2017). https://firstdraftnews.com/fake-news-complicated/
5. Throne, J., Chen, M., Myrianthous, G., Pu, J., Wang, X., Vlachos, A.: 2017. Fake News Detection using Stacked Ensemble of Classifiers. In ACL.
6. Davis, R., Proctor, C.: 2017. Fake News, Real Consequences: Recruiting Neural Networks for the Fight against Fake News. https://web.stanford.edu/class/cs224n/reports/2761239.pdf
7. http://www.fakenewschallenge.org/
8. http://www.emergentresearch.com/
9. SemEval-2016: Semantic Evaluation Exercises International Workshop on Semantic Evaluation (SemEval-2016). http://alt.qcri.org/semeval2016/
10. Sobhani, P., Inkpen, D., Zhu, X.: A Dataset for Multi-Target Stance Detection. http://www.aclweb.org/anthology/E17-2088.
11. Pang, B., Lee, L.: 2008. Opinion mining and sentiment analysis. http://www.cs.cornell.edu/home/llee/omsa/omsa.pdf
12. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, M., Kuksa, P.: 2011. Natural Language Processing (Almost) from Scratch. Journal of Machine Learning Research 12 (2011) 2493-2537.
13. Johnson, J.: 2016. The Five Types of Fake News. https://www.huffingtonpost.com/dr-john-johnson/the-five-types-of-fake-ne_b_13609562.html
14. As fake news spreads lies, more readers shrug at the truth. In https://www.nytimes.com/ 2016/12/06/us/ fake-news-partisan-republican-democrat.html.
15. https://en.wikipedia.org/wiki/Fake_news
16. https://www.tandfonline.com/doi/pdf/10.1080/00437956.1954.11659520
17. Zarrella, G., Marsh, A.: MITRE at SemEval-2016 Task 6: Transfer Learning for Stance Detection. https://arxiv.org/pdf/1606.03784.pdf
18. Wei, W., Zhang, X., Liu, X., Chen, W., Wang, T.: pkudblab at SemEval-2016 Task 6 : A Specific Convolutional Neural Network System for Effective Stance Detection. http://www.aclweb.org/anthology/S16-1062
19. https://en.wikipedia.org/wiki/Natural-language_processing
20. https://nlp.stanford.edu/projects/glove/
21. Haykin, S.: NEURAL NETWORKS A Comprehensive Foundation Second Edition.
22. Collobert, R., Weston, J.: A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. http://www.thespermwhale.com/jaseweston/papers/unified_nlp.pdf

23. William Ferreira and Andreas Vlachos, “Emergent: a novel data-set for stance classification” http://aclweb.org/anthology/N/N16/N16-1138.pdf
24. https://machinelearningmastery.com/prepare-text-data-machine-learning-scikit-learn/