BATTLING FAKE NEWS: A STEP-WISE APPROACH TOWARDS STANCE DETECTION

CS-406 (UIP) PROJECT REPORT

Dhruv Agarwal
Department of Computer Science
Ashoka University

Prakhar Jain
Department of Computer Science
Ashoka University

December 1, 2019

ABSTRACT

The advent of social media has brought along with it a convenient way of sharing information with the masses. This has allowed people to share inauthentic information and click-baits in order to garner publicity in the form of views, like, and comments. The problem of fake news detection is complex, but can be broken down in numerous small steps. The first amongst these—called Stance Detection—is to find what different articles are saying about a given topic. More specifically, given a headline, does an article agree with it, disagree with it, discuss it, or is unrelated to it? In this paper, we present a step-wise supervised-learning approach towards stance detection. We break down the problem into two parts: detecting relatedness, and then detecting the class of relatedness. That is, only if we find that an article is *related* to a headline do we find the class of relatedness – *agreement*, *disagreement*, or *discussion*. In both the steps, we adopt a TF-IDF/BoW and neural network based approach. However, the steps differ in text representation, feature engineering, and the training process. We obtain better than state-of-the-art results in classifying *disagreement*, and close to state-of-the-art results in the other classifications.

***K*eywords** Fake news · Stance detection · Deep learning · NLP · TF-IDF · Bag-of-words

1 Introduction

The Pew Research Center suggests that **55% of adults** in the United States get their **news from social media**. The report does not present such a number for teenagers and children, even though they are more exposed to social media than adults, *and* are more gullible to believe fake news that large corporations throw onto them. In the age of Social Media, where everyone is a publisher, the task of **finding the truth** amongst the millions of fake pieces is **important but cumbersome**. One way of doing this is to fact-check every piece of news that one reads. However, this requires extensive research from primary sources, which defeats the purpose of having a secondary source to quickly relay only the important points to busy readers. Moreover, due to the time-consuming nature of fact-checking and the ephemeral nature of news, researching about the topic could take enough time for the news to become obsolete and irrelevant. Hence, it is important that we come up with **faster methods of detecting fake news**.

Automating the detection of fake news is a complex and cumbersome task. However, fortunately, it can be broken down into smaller sub-tasks. A helpful first step towards fake news detection is to find **what different new outlets are saying about a given piece of headline**. A given news headline can be compared to articles from a large number of media outlets. If a majority of these **articles *disagree*** with the news headline in question, the headline and the corresponding article is **probably fake news**. The role of stance detection in flagging fake news is shown in figure 1.

In this paper, we present a novel step-wise supervised-learning approach to stance detection. Instead of training a single model to classify amongst the four classes, we break down the process into two steps: detecting relatedness, and then detecting the class of relatedness. With this approach, we are able to classify *disagree* with 13.2% accuracy while the state-of-the art model only reached 9% accuracy. We explain our model in detail in section 5.

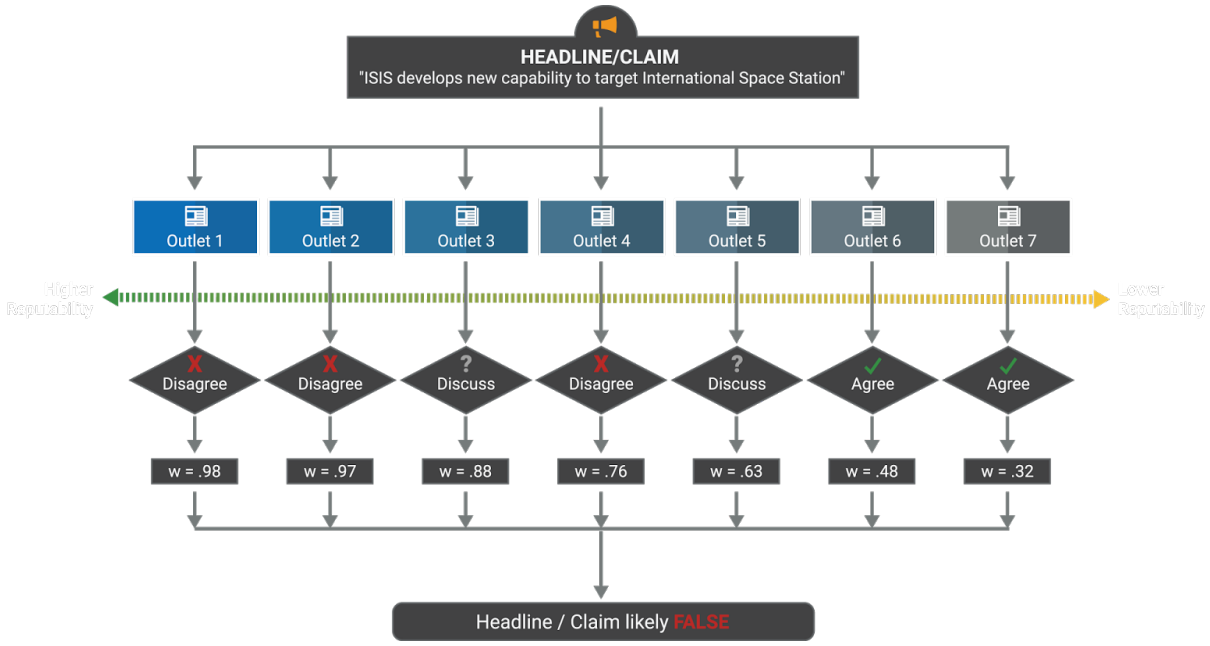

Figure 1: The role of Stance Detection in Fake News detection. Courtsey: Talos Intelligence

2 The Problem Definition

Input A headline and a body text - either from the same news article or from two different articles.

Output Classify the stance of the body text relative to the claim made in the headline into one of four categories:

- **Agrees**: The body text agrees with the headline.
- **Disagrees**: The body text disagrees with the headline.
- **Discusses**: The body text discuss the same topic as the headline, but does not take a position.
- **Unrelated**: The body text discusses a different topic than the headline.

Table 1 shows examples of headline-article combinations that lead to the 4 different stance classes.

3 Previous Work

The problem of automatic stance detection has been attempted in the past using a variety of machine learning methods. This problem attracted a lot of attention from industry and academia in 2017, as part of the first stage of the Fake News Challenge (FNC-1). The FNC-1 was a grassroots effort of over 71 teams around the world to explore how artificial intelligence technologies could be leveraged to combat fake news. The idea was "address the problem of fake news by organizing a competition to foster development of tools to help human fact checkers identify hoaxes and deliberate misinformation in news stories using ML, NLP and AI" [1].

The winning team, Talos Intelligence, used an 50/50 weighted average of a Gradient-Boosted Decision tree and a deep convolutional neural network. The deep CNN used a word embedding from Google News pre-trained vectors as input and uses dropouts for regularization. The preprocessing step used n-grams to derive features like TF-IDF, SVD and Word2Vec. These features were then fed into the Gradient-Boosted Decision Trees (GBDT) model to output one of the four classes. They performed various scoring techniques like difference in predicted class value and actual class value to weight their models accordingly [2].

The team that stood second, TU-Darmstadt, used a deep network approach to auto-detect features from the input feature vector. Their feature vector consisted of a bag-of-words (BoW), a latent semantic indexing (LSI), and Paraphrase

Headline	Body (Partial)	Stance
Hundreds of Palestinians flee floods in Gaza as Israel opens dams	Hundreds of Palestinians were evacuated from their homes Sunday morning after Israeli authorities opened a number of dams near the border, flooding the Gaza Valley in the wake of a recent severe winter storm. ... Gaza has experienced flooding in recent days amid a major storm that saw temperatures drop and frigid rain pour down. ... In Dec. 2013, Israeli authorities also opened the dams amid heavy flooding in the Gaza Strip. ... In 2010, the dams were opened as well, forcing 100 families from their homes. ...	Agree
Spider burrowed through tourist's stomach and up into his chest	Fear not arachnophobes, the story of Bunbury's "spiderman" might not be all it seemed. Perth scientists have cast doubt over claims that a spider burrowed into a man's body during his first trip to Bali. ... Dylan Thomas says he had a spider crawl underneath his skin. ... But it seems we may have all been caught in a web...of misinformation. Arachnologist Dr Volker Framenau said whatever the creature was, it was "almost impossible" for the culprit to have been a spider. ... They canŕ get through skin," he said.	Disagree
Seth Rogen to Play Apple's Steve Wozniak	Seth Rogen is being eyed to play Apple co-founder Steve Wozniak in Sony's Steve Jobs biopic. Danny Boyle is directing the untitled film, based on Walter Isaacson's book and adapted by Aaron Sorkin, which is one of the most anticipated biopics in recent years. ... Of course, this may all be for naught as Christian Bale, the actor who is to play Jobs, is still in the midst of closing his deal. ... Insiders say Boyle will is flying to Los Angeles to meet with actress to play one of the female leads, an assistant to Jobs. ...	Discuss
Police find mass graves with at least '15 bodies' near Mexico town where 43 students disappeared after police clash	Seth Rogen is being eyed to play Apple co-founder Steve Wozniak in Sony's Steve Jobs biopic. Danny Boyle is directing the untitled film, based on Walter Isaacson's book and adapted by Aaron Sorkin, which is one of the most anticipated biopics in recent years. ...	Unrelated

Table 1: Examples of the 4 stance classes

Detection (PPDB) representation of the headlines and articles. This vector was passed as input to a 5-7 layered deep neural networks, which outputted the predicted class label [3].

The team that stood third also used a Deep Neural Network but instead of providing the network with a bag of words, they used a term frequencies vector with vocabulary as the 5000 frequently occurring words in the corpus. The team also inputted to the DNN the cosine similarity between the TF vectors representing the body and headline. The input to the Deep Network was thus the TF vectors of size 5000 each for headlines and body and a single cosine similarity value, making the total input of size 10001 [4].

However, neither of the top 3 teams achieved **more than 9% accuracy** in predicting *disagree* correctly. Since predicting *disagree* is the **very basis of the efficacy** of stance detection in fake news flagging, we believe that more effort needs to be made to make **correct predictions in this class**, even at the cost of lesser overall accuracy. Hence, our approach focuses on correct prediction within *agree*, *disagree*, and *related* classes.

4 Dataset

We use the dataset provided by the Fake News Challenge organizers, which itself is sourced from the Emergent Dataset created by Craig Silverman. The data provided is `(headline, body, stance)` instances, where stance is one of `{unrelated, discuss, agree, disagree}`. Some of the samples in the dataset have been shown in table 1. We are also provided with another file with similarly structured data for testing our model. The distribution of stance classes in the training data is shown in table 2.

Stance	Samples	Percentage
agree	3678	7.36%
disagree	840	1.68%
discuss	8909	17.82%
unrelated	36545	73.13%

Table 2: Training Dataset Distribution

Since we adopt a step-wise approach, we transform the class labels in the training dataset in the two steps. We describe these transformations in detail in section 5. Moreover, it must be noted that the dataset is extremely unbalanced, and hence biased towards the *unrelated* class. Hence, we also perform data augmentation on the training set in the second step of our approach. We describe this augmentation in detail in section 5 as well.

5 Model

We break down the Stance Detection problem into two steps. This architecture is shown in figure 2. In step 1, we predict if the given news headline is *related* to the given article or not. If it is not related, we immediately output *unrelated* and we're done. But if it *is* related, we pass the headline and the article to step 2. In step 2, we predict the class of relatedness – *agree*, *disagree*, or *discuss*.

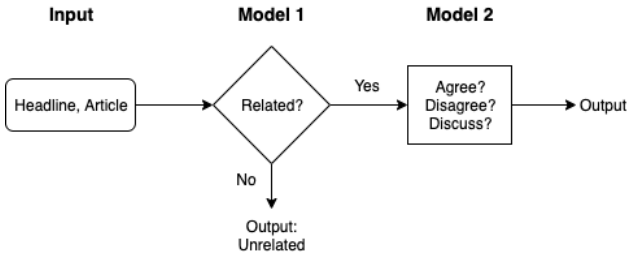

Figure 2: Step-wise System Architecture for Stance Detection

Why a Step-wise approach? None of the teams who finished in the top 3 ranks in the Fake News Challenge adopted a step-wise approach to this problem. We believe that this is one of the reasons which led to the poor performance of their models on predicting the class of relatedness, especially *agree* and *disagree*. We choose a step-wise approach because of primarily two reasons:

- We observe that the class *unrelated* is not logically comparable to the classes *agree*, *disagree*, or *discuss*. Relatedness/unrelatedness is based **simply on the similarity** of two text documents, whereas as agreement/disagreement/discussion is based more **deeply upon the content** of the two documents. These 3 classes can be thought of as sub-classes within *related*.
- Due to a large number of examples of *unrelated* (73%), a **single model will only learn similarity metrics** between documents to classify them between the 4 classes. It will **not learn to find deeper insights** by looking at the content of the documents. Hence, it will perform well on predicting *unrelated*, but it will not perform well on predicting the other 3 classes. This observation is validated by the results of the winners of FNC-1. Therefore, it is better to train a **specialized model** that can learn to look at deeper differences in documents for the agree/disagree/discuss classification.

We now describe the two models in detail.

5.1 Model 1: Predicting Related-Unrelated

5.1.1 Data Preparation

Since this model makes a 2-class classification, we need to transform our dataset accordingly. For this, we make use of the observation that *agree*, *disagree*, and *discuss* are sub-classes of *related*. Hence, for each training example that belongs to one of these 3 classes, we transform the stance into *related*. More formally, we apply the the function f

on the stance column of the dataset, where $f : \{\text{agree}, \text{disagree}, \text{discuss}, \text{unrelated}\} \rightarrow \{\text{related}, \text{unrelated}\}$, and f is defined below. The effect of this transformation on the training dataset is shown in table 3.

$$f(x) = \begin{cases} \text{related} & x \in \{\text{agree}, \text{disagree}, \text{discuss}\} \\ \text{unrelated} & x == \text{unrelated} \end{cases}$$

Headline	Body	Old Stance	New Stance	One-hot Encoded
h_1	b_1	agree	related	$[0, 1]$
h_2	b_2	disagree	related	$[0, 1]$
h_3	b_3	discuss	related	$[0, 1]$
h_4	b_4	unrelated	unrelated	$[1, 0]$

Table 3: Effect of the transformation f on Model 1

Finally, we one-hot encode the stance labels to avoid imposing any unnatural ordering on the data.

5.1.2 Feature Extraction

Document Representation: TF-IDF First, we need a way of representing the documents in a form that can be input into our neural network. For this model, we use a TF-IDF representation for each document. For this, we first create a word-document matrix, T, such that each row represents a word, and each column represents a document [1]. For computational convenience, we limit the number of words, i.e. the size of the vocabulary, to 5000 most frequently occurring words in our corpus after removing stopwords. We choose to remove stopwords because they do not help in gauging relatedness between two documents.

Then, T is filled as $T_{ij} = TF_{ij} \times IDF_i$, where:

$$TF_{ij} = \frac{\text{\# of times word } i \text{ appears in document } j}{\text{Total \# of words in document } j}$$

$$IDF_i = \log \frac{\text{Total \# of documents}}{\text{\# of documents with word } i \text{ in them}}$$

Now, for each training sample in our datatset, we use T to compute the TF-IDF representation of the headline and the article.

Computing Similarity Metrics For this model, we do manual feature extraction. We choose this approach because relatedness/unrelatedness is a form of similarity, and we already know various similarity metrics. Moreover, this also helps prevent overfitting that would be caused by passing in the raw TF-IDF vectors to the nerual network.

At this stage, we have two vectors of length 5000 – one each for the headline (h) and the article body (b). We compute Euclidean Distance (ED), Cosine Similarity (CS), and Dice Similarity (DS) on these two vectors. These similarities are calculated as follows:

$$ED(h, b) = \sqrt{\sum_{i=1}^{5000} (h_i - b_i)^2}$$

$$CS(h, b) = \frac{\sum_{i=1}^{5000} h_i \cdot b_i}{\sqrt{\sum_{i=1}^{5000} h_i^2} \sqrt{\sum_{i=1}^{5000} b_i^2}}$$

$$DS((h, b) = 2 \times \frac{\sum_{i=1}^{5000} \min(h_i, b_i)}{\sum_{i=1}^{5000} (h_i + b_i)}$$

We use these three similarity metrics as input into the neural network.

[1] Note that for this matrix, we club the headlines and the articles together into a common corpus. Hence, the number of columns in T is equal to the number of unique articles plus the number of unique headlines.

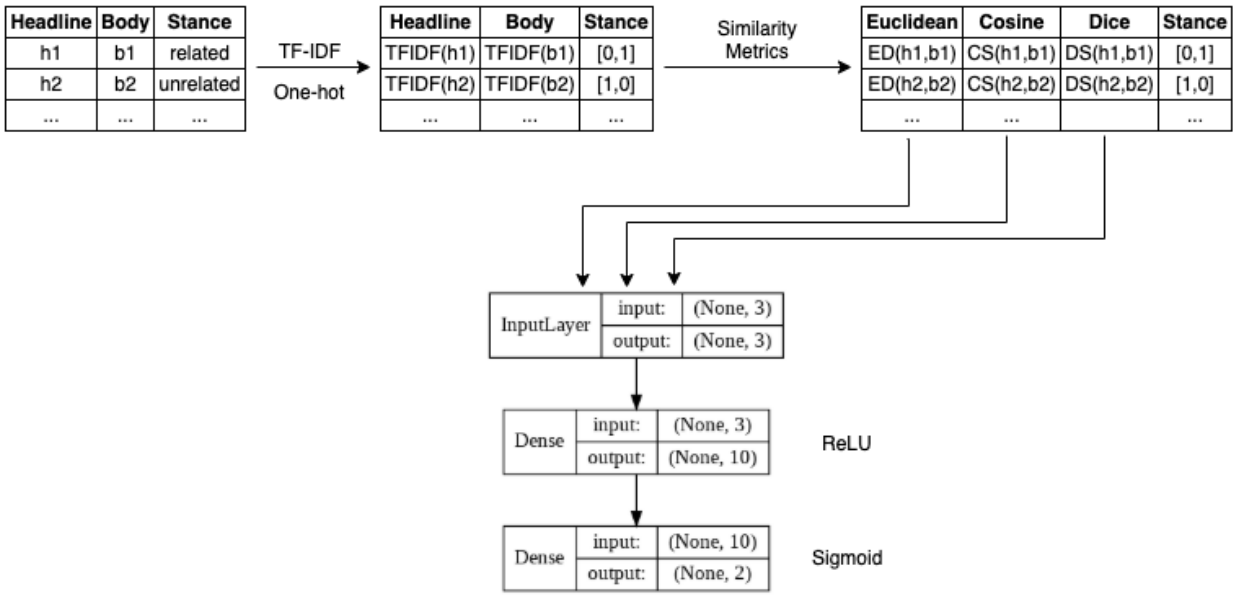

Figure 3: The entire pipeline for training Model 1 for related/unrelated

5.1.3 Model Architecture

Since we do manual feature extraction for this model, a shallow network suffices. The input into the neural network is the 3 similarity scores: Euclidean Distance, Cosine Similarity, Dice Similarity. These inputs are passed through 1 hidden layer containing 10 neurons, each with a ReLU activation function.

We use the binary-crossentropy loss function and Adagrad optimizer to train the network. Further, to account for the imbalance in the dataset (73% *unrelated* and 27% related), we also weigh the loss function accordingly: 0.68 times for *unrelated* samples and 1.86 times for *related* samples.

The entire training process is shown in figure 3.

5.2 Model 2: Predicting Agree-Disagree-Discuss

5.2.1 Data Preparation

Since this model predicts between the three sub-classes of *related*, we remove all the training examples belonging to the class *unrelated*. This results in the following distribution of data: 840 examples of *disagree*, 3678 examples of *agree*, 8909 examples of *discuss*. As in model 1, we again one-hot encode the stance labels to remove integer ordering bias.

5.2.2 Feature Extraction

Document Representation: TF (Term Frequency) First, we need a way of representing the documents in a form that can be input into our neural network. For this model, we use a TF representation for each document. Like in step 1, we limit the size of the vocabulary to to 5000 most frequently occurring words in our corpus. However, in this case, we **keep the stopwords**. This is because stopwords like "not", "isn't", etc. provide a lot of evidence about agreement or disagreement. The process of computing the term-frequency table is similar to that in step 1. The word-document matrix, T, is filled simply as $T_{ij} = TF_{ij}$, where TF_{ij} is defined in the same way as in step 1:

$$TF_{ij} = \frac{\text{\# of times word } i \text{ appears in document } j}{\text{Total \# of words in document } j}$$

For this model, we use TF as opposed to BoW or TF-IDF for two reasons:

- TF gives better performance than a simple Bag-of-Words approach because TF normalizes the frequency of occurrence of a word by the length of the document. In our case, we have a common corpus of headlines and articles, and the headlines and are much shorter than the articles. Hence, using BoW would mean giving

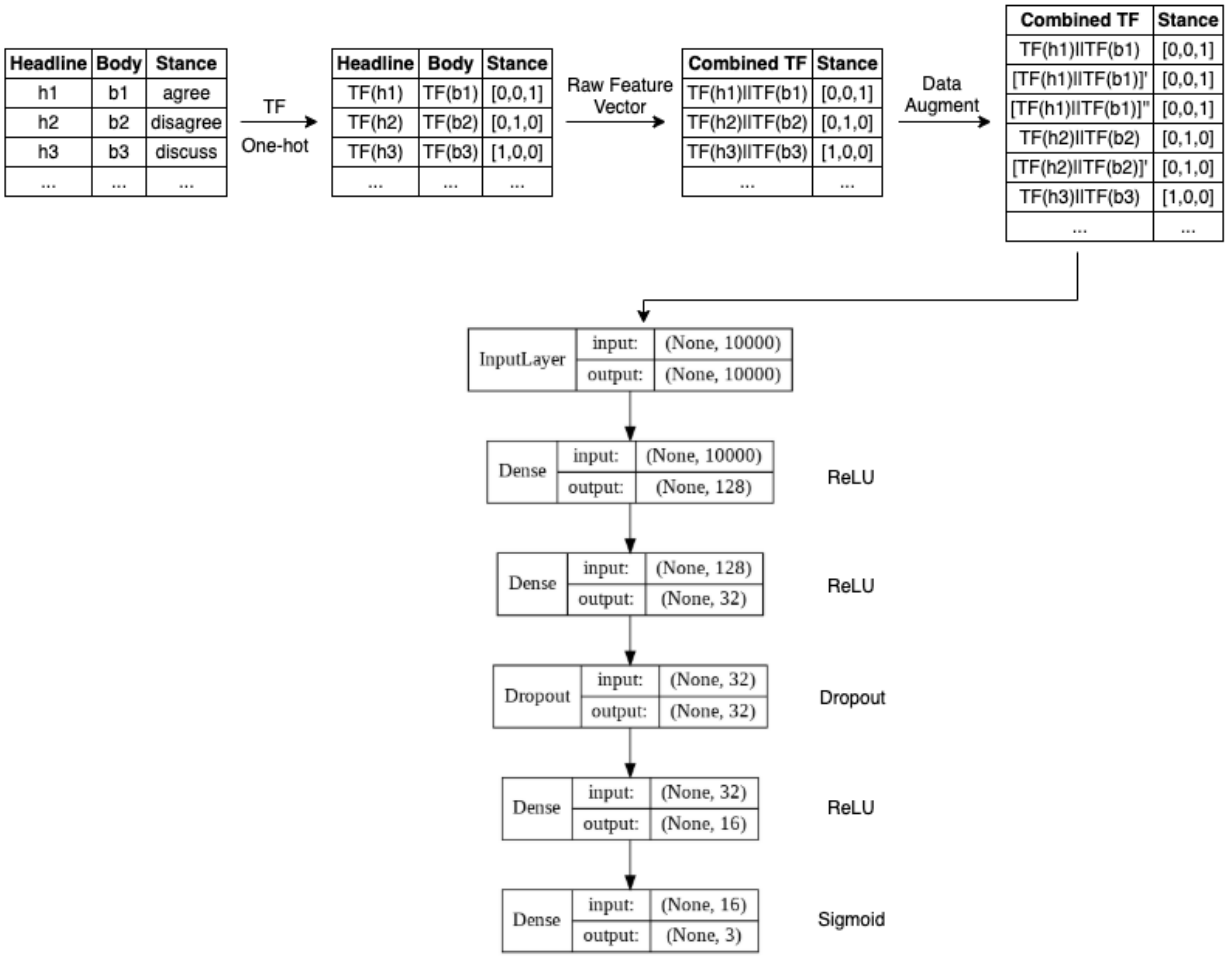

Figure 4: The entire pipeline for training Model 2 for agree/disagree/discuss

higher priority to words in the articles than the words in the headlines. To avoid this inappropriate weighting of articles, we use a TF representation.

- TF gives (slightly) better performance than TF-IDF because commonly occurring words like "not", "for", "against", etc. that give a lot of information about agreement/disagreement are not weighed down by the IDF value. Essentially, keeping stopwords would be useless if we still weighted using IDF values.

Now, for each training sample in our datatset, we use T to compute the TF representation of the headline and the article.

Constructing Raw Feature Vector For this model, we do not do manual feature extraction. Instead, we give the raw TF vectors to a deep neural network and expect it to do feature extract for us. We choose automatic feature extraction because for this classification, we are looking for deep connections between words and sentences. The neural network can find these features in a high-dimensional space, which we cannot do manually.

Hence, we simply concatenate the two 5000 length vectors (one each for the headline and the article) into a single vector of length 10000, which we input into the deep neural network.

5.2.3 Data Augmentation

In this step, unlike in step 1, we also augment our dataset with more examples of *disagree* and *agree*. There are two reasons for this augmentation – one obvious immediately, and the other more subtle.

- The dataset is extremely unbalanced. As noted in 5.2.1, only 6% of our data is *disagreed* and only 27% of our data is *agreed*. Hence, we add more samples from these classes to avoid biasing our model towards always predicting *discussed*.

- *Discuss* subsumes *agree* and *disagree*. In normal conversation and writing, **agreement and disagreement is a part of discussion**. Hence, it is very difficult to teach the model how to differentiate between *agree/disagree* and *discuss*. In some sense, our data itself confuses the model – there is no clear decision boundary between *agree/disagree* and *discuss*! Hence, we add random noise to our existing examples so as to create a better decision boundary.

To augment the data, we go through each example of class *agree* and *disagree*. For each such sample, we randomly pick 1,000 features out of the 10,000 length feature vector that we want to add noise to. We sample a random number from a standard normal distribution ($\mu = 0$, $\sigma = 1$), and add it to the existing value in the chosen feature. Since the resulting value may be negative, we simply take the absolute value of the result of this addition. More formally, for all 1000 chosen indices i, we do $x_i \leftarrow |x_i + \text{rand}(\mu = 0, \sigma = 1)|$. Note that we cannot choose the random number to add depending upon the average or maximum value in that feature, because this feature vector is very sparse and hence the average and maximum values may be 0. We use this use modified feature vector of length 10,000 without the changing the original label.

5.2.4 Model Architecture

In this model, since we do not do any manual feature extraction, we use a deep neural network with dense layers. The input into the neural network is the feature vector of length 10,000. This input is passed through 3 hidden layers, each with a ReLU activation function. Due to the size of the input feature vector, this model is prone to heave over-fitting. To regularize and counter over-fitting, we also use a dropout layer in our network.

We use the binary-crossentropy loss function and Adam optimizer to train the network. Further, to account for the imbalance in the dataset as described above, and to incentivize the model to answer *agree* or *disagree* instead of *discuss*, we artificially weigh the loss function: 10 times for *disagree* samples and 2 times for *agree* samples. These weights have been chosen by trial-and-error.

The entire training process is shown in figure 4.

6 Failed Approaches for Model 2

As noted earlier, we focused more on trying to improve *agree/disagree* predictions. In this section we describe the various approaches we tried before selecting upon the approaches described above in section 5. These approaches helped us identify the relevant features that we should focus on to solve the problem.

6.1 Random Forest

Since the problem seemed to be a decision problem, we believed that a tree based approach could produce good results. We provided the model a bag-of-words representation of headlines and articles. We expected that the model would do an instance level learn to identify the group of words that lead to *agreement/disagreement*. The model achieved 65% accuracy on test data but performed poorly on distinguishing between *disagree* and *discuss* stances.

6.2 Doc2Vec with Deep Neural Network

Since we needed to find closeness between documents on the basis of the context in which the words were used, we also tried a Doc2Vec model to generate representation of documents. We represented the article body with a vector of length 65 and the headline with a vector of 15 (since headlines are much shorter than articles). We then fed these features to a deep neural network expecting it to find the relation between the two vectors by deriving contextual information. The model gave us only 57% testing accuracy. We believe that this was because Doc2Vec model only takes into account closeness between docucments as a whole, but does not represent the document in terms of the words that it contains. This word-level information that it ignores if crucial for *agree/disagree/discuss* distinction.

6.3 Boosting using AdaBoost

Since *discuss* is a generalized word and can encompass *agree* and *disagree*, our model was biased towards it even after we augmented data for the other classes. We used AdaBoost hoping that our model would learn the 3 classes indiscriminately. This model gave us only about 45% accuracy on the testing dataset.

6.4 Stemming

A news article and a news heading can be in different tenses and/or use different verbs. Quite frequently, a news headline is in present tense while the news is in past tense. We felt that stemming our corpus could help solve this issue by converting words into their root forms. This approach did well, but not better than the without-stemming approach. We got a testing accuracy of 73% which was lesser than the accuracy we achieved without stemming. We suspect that stemming lead us to lose context which was crucial to detect stance.

6.5 Bag of Words

The Bag of Words approach works by taking the most common words in the entire corpus and representing all the documents using these words. As explained previously, this model gave us a 70% testing accuracy. We realized that this model was weighing the bodies more than the news headline because bodies are longer and hence have higher frequency of words. This realization led us into the direction of a normalizing frequency by document length, i.e., a TF vector representation.

7 Results

7.1 Model 1: Classifying Related and Unrelated

As can be seen from the confusion matrix shown in Table 5, our model given slightly better predictions for the *unrelated* class. One reason for this is that the training set had more examples for the *unrelated* class. Neverthelesss, our model performs well on the *related* class as well. This means that the features we extracted are relevant and capture the problem sufficiently. The model performs well on unseen real-world data in Table 4 as well. Even though the training set had no technology related news, the model is able to classify technology news headlines and bodies as well. We thus conclude that ‘ “Eucledian Distance”, “Cosine Similarity” and “Dice Similarity” sufficiently capture the differences between *Related* and *Unrelated* news headlines and news bodies. This can further be seen by the fact that even though we provided just 3 floating point inputs, our model did not overfit on the training data.

7.2 Model 2: Classifying Related news items into Agree-Disagree-Discuss

Since Discuss is a more generalized form of Agree and Disagree, unless the headlines strongly agree or disagree with the body, the model predicts the relation between them as discuss. We thus see better results for the Discuss class. The training data did not have any sports related news article, but the network predicted the results accurately. Interestingly, changing even a single word like “worth” in Table 6 at any place would make the the combination be predicted as “Discuss”. The accuracy achieved by state of art models for disagree were 9% while our accuracy is 13.2% on the testing set in Table 7. Since the challenge of detecting fake news is essentially dependent on identifying bodies and headlines that Disagree with each other, we believe that our model works better than the present state of the art models. Our model seems to overfit, which can be attributed to the fact that we gave it almost the entire corpus in the training process, but, providing it any less data might mean leaving out information relevant to a particular news article.

8 Conclusion

The problem of fake news needs can be broken down into smaller sub-problems. The first step in detecting fake news is finding the stance of multiple media outlets with regards to a given news headline. However, this task of fact-checking is cumbersome and time-consuming. Hence, there is a rising need to automate stance detection. We employed a two step solution to solve this problem. We first filter for *relatedness* or *unrelatedness*. If we find relatedness, we pass it through a agree-disagree-discuss filter. We found out that detecting disagreement is a hard task given the small syntactical differences in a sentence that can change its property. This process requires deep understanding of the context and domain.

References

[1] “Fake news challenge.” `http://www.fakenewschallenge.org/`. Accessed: 2019-11-30.

[2] “Talos targets disinformation with fake news challenge victory.” `https://blog.talosintelligence.com/2017/06/talos-fake-news-challenge.html`. Accessed: 2019-11-30.

Headline	Body	Real Stance	Predicted Stance
The lowest-ever Apple Watch 3 price is back... but there's a catch	The best-ever Apple Watch 3 deal is back in stock - although you're not going to get it easily. The white, 38mm version of one of our best wearables is down to $129 - a simply amazing price for what you're getting - but you'll need to seek it out in a local store.That's right: while we're hearing a few reports that you can still snag it online if you're lucky, the Apple Watch 3 is only going to be this low price if you're willing to walk to a local store.We'd also add the caveat that the Walmart prices are fluctating wildly throughout the day on a number of deals, so you might find this Apple Watch 3 deal fluctuating throughout the day.. ...	Related	Related
Social media solidarity for wounded Palestinian journalist	Thousands of social media users have taken part in a campaign to support Palestinian freelance journalist Muath Amarneh, who lost his eye after being wounded while covering a protest in the occupied West Bank last week.Journalists from across the Arab world launched the campaign of solidarity after Amarneh was hit in the eye, apparently with a rubber bullet fired by Israeli forces who had arrived to quell the demonstrations last Friday in Surif.	Related	Related
The lowest-ever Apple Watch 3 price is back... but there's a catch	Thousands of social media users have taken part in a campaign to support Palestinian freelance journalist Muath Amarneh, who lost his eye after being wounded while covering a protest in the occupied West Bank last week.Journalists from across the Arab world launched the campaign of solidarity after Amarneh was hit in the eye, apparently with a rubber bullet fired by Israeli forces who had arrived to quell the demonstrations last Friday in Surif.	Unrelated	Unrelated
Social media solidarity for wounded Palestinian journalist	The best-ever Apple Watch 3 deal is back in stock - although you're not going to get it easily. The white, 38mm version of one of our best wearables is down to $129 - a simply amazing price for what you're getting - but you'll need to seek it out in a local store.That's right: while we're hearing a few reports that you can still snag it online if you're lucky, the Apple Watch 3 is only going to be this low price if you're willing to walk to a local store.We'd also add the caveat that the Walmart prices are fluctating wildly throughout the day on a number of deals, so you might find this Apple Watch 3 deal fluctuating throughout the day.	Unrelated	Unrelated

Table 4: Real-World Predictions from Model 1

	True Related	True Unrelated
Predicted Related	**4903**	2163
Predicted Unrelated	890	**17459**
Accuracy	84.62%	88.97%

Table 5: Confusion Matrix: Test Data Predictions from Model 1

Headline	**Body**	**Real Stance**	**Predicted Stance**
Truly innovative products leave their mark on the world instead of the planet.	Just as much innovation goes into the materials your Apple products are made of — and how they're made — as into what they do. You can see that in the new MacBook Air and Mac mini. Their enclosures are made from 100% recycled aluminium, without compromising strength or finish. In so many ways, the most advanced products are the ones that make the least environmental impact.	Agree	Agree
Social media solidarity for wounded Palestinian journalist	Thousands of social media users have taken part in a campaign to support Palestinian freelance journalist Muath Amarneh, who lost his eye after being wounded while covering a protest in the occupied West Bank last week. Journalists from across the Arab world launched the campaign of solidarity after Amarneh was hit in the eye, apparently with a rubber bullet fired by Israeli forces who had arrived to quell the demonstrations last Friday in Surif.	Discuss	Discuss
Messi is a left legged human being and is extremely tall and muscular. He is the best footballer on the planet	Messi is a short footballer. He is such a mediocre player that he may make his team lose single-handedly. One should not ideally buy him for his worth. He should be valued far less and then he might be worth it. Since Barca got him for free, he is technically worth his worth.	Disagree	Disagree

Table 6: Real-World Predictions from Model 2

	True Disgree	**True Agree**	**True Discuss**
Predicted Disagree	**92**	103	928
Predicted Agree	346	**1085**	177
Predicted Discuss	259	715	**3359**
Accuracy	13.19%	57.01%	75.25 %

Table 7: Confusion Matrix: Test Data Predictions from Model 2

[3] "Team athene on the fake news challenge." `https://medium.com/@andre134679/team-athene-on-the-fake-news-challenge-28a5cf5e017b`. Accessed: 2019-11-30.

[4] B. Riedel, I. Augenstein, G. P. Spithourakis, and S. Riedel, "A simple but tough-to-beat baseline for the fake news challenge stance detection task," *CoRR*, vol. abs/1707.03264, 2017.