FAD: A Fake News Detection System

Abhinav Kohar
University of Illinois,
Urbana-Champaign
aa18@illinois.edu

Sidhartha Satapathy
University of Illinois,
Urbana-Champaign
ss46@illinois.edu

Samarth Kulshreshtha
University of Illinois,
Urbana-Champaign
samarth5@illinois.edu

ABSTRACT

With the increasing use of internet and social media by people across the globe, online news reading has become extremely popular due to the quick and easy availability of news articles from millions of sources around the world. An important question that demands an answer is how credible are these articles? The said credibility is of paramount importance because these articles have a potential to influence the opinions of masses which in turn can have an impact on policies over a global scale. It is not surprising that there have been cases reported where groups of people are deliberately releasing fake news articles on the web to influence popular opinion. This justifies the need to have an automated mechanism to identify fake articles. However, assessing the truthfulness of a news story is a complex and difficult task even for humans due to the regular use of sarcasm and facts that look seemingly true. We believe that Machine Learning and Natural Language Processing techniques can be leveraged to their maximum in order to solve this intricate and involved problem. In this project, we propose a few novel approaches to identify the stance of a news article's body relative to its headline. This, in essence, is the first step towards detecting fake news and can be eventually used in an AI-assisted fake news detection pipeline. In particular, we plan on classifying the body of a new article relative to a headline into the following four categories - agreeing, disagreeing, discussing or unrelated. Describe model and results and move current abstract under intro

CCS CONCEPTS

• **Computing methodologies** → *Natural language processing*; *Machine learning approaches*; *Machine learning algorithms*; Learning paradigms;

KEYWORDS

Fake news, stance detection

1 INTRODUCTION

Identifying fake news has become an important problem of our times, with the recent accusations of fake news helping Donald Trump win the presidential elections [5]. Fake news is usually used to sway the mood of masses with certain hidden agenda, like an increase in sales using fake advertising, for winning elections or some other ulterior gain. However, the effects of fake news can be damaging and are far reaching.

FNC, a non-profit organization released the fake news dataset to help develop new machine learning methodologies to tackle the above challenge. Identifying fake news is done in a human augmented way and consists of two parts. Firstly, stance detection in news articles and, secondly, this classified data is fed to human experts or graders to finally recognize if the article is fake or not [20]. Stance detection has been defined as identifying the relevance between two pieces of text which are based on the same topic: this can be for, against or simply observing the claim. [7]

One thing to note is that news articles which are against each other or disagree do not necessarily imply they are fake. This input though serves an an invaluable flag in the human augmented pipeline.

The rest of the paper is organized as follows: we present the problem statement, define the data set and evaluation metrics. Thereafter, we discuss the relevant related work. We then present the baseline model used for evaluation, and discuss our implementation .bidirectional LSTM with different describe feature sets- word embdeddings , glove etc. Towards the end we elaborate on evaluation of our model and compare it with the given baseline, and provide a conclusion and scope for future work.

2 FORMAL PROBLEM STATEMENT

2.1 Input

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

2.2 Output

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

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

3 DATA

3.1 Format

The dataset is in the following format:

- **Training set** - [HEADLINE, BODY TEXT, LABEL]
  Pairs of headline and body text with the appropriate class label for each.
- **Testing set** - [HEADLINE, BODY TEXT]
  Pairs of headline and body text without class labels used to evaluate systems.
- **Data distribution** - Table 1 provides a summary of proportion of the various classes present in the data.

Table 1: Data Distribution

rows	unrelated	discuss	agree	disagree
49972	0.73131	0.17828	0.0736012	0.0168094

3.2 Summary

The dataset contains 1648 distinct headlines, 1683 distinct articles, and 49972 distinct headline and article pairs. Headline lengths range from around 10 to 200 words whereas article lengths range from around 25 to 5000 words. Another important factor is that the dataset is heavily biased towards unrelated headline and article pairs. This justifies the scoring criteria which has a reduced weight for correctly identifying related/unrelated tuples and an increased weight for correctly predicting discuss/agree/disagree. More details about data characteristics are provided in the Appendix.

3.3 Example

For the headline, "*Bali spider burrows under Aussie's chest*", example body texts for the various classes are given below:

- **discuss:** "..this story that has been distorted through misdiagnosis or misunderstanding, but in the absence of any other evidence we remain extremely skeptical.."
- **agree:** "..that's where it actually borrowed underneath my skin.."
- **disagree:** "..the doctors in Bali pulled something out of Thomas's body, but it was far more likely to be a tick or a mite. Just not a spider.."
- **unrelated:** "..Keen golfers can have their very own putting paradise if they snap up this luxury island previously owned by Tiger Woods..:

3.4 Baseline

The Fake News Challenge-1 provides a baseline implementation based upon GradientBoosting classifier and hand-coded features. These hand-coded features include word/ngram overlap features, and indicator features for polarity and refutation. Such a classifier with 10-fold cross validation achieves an accuracy of 79.53% as per the scoring metric described in Figure 1.

4 MODEL EVALUATION

4.1 Criteria

Evaluation of the models is based on a weighted, two-level scoring system:

- **Level 1:** Classify headline and body text as related or unrelated 25% score weighting
- **Level 2:** Classify related pairs as agrees, disagrees, or discusses 75% score weighting

Rationale: The related/unrelated classification task is expected to be much easier and is less relevant for detecting fake news, so it is given less weight in the evaluation metric. The Stance Detection task (classify as agrees, disagrees or discuss) is both more difficult and more relevant to fake news detection, so is to be given much more weight in the evaluation metric.

4.2 Scoring schematic

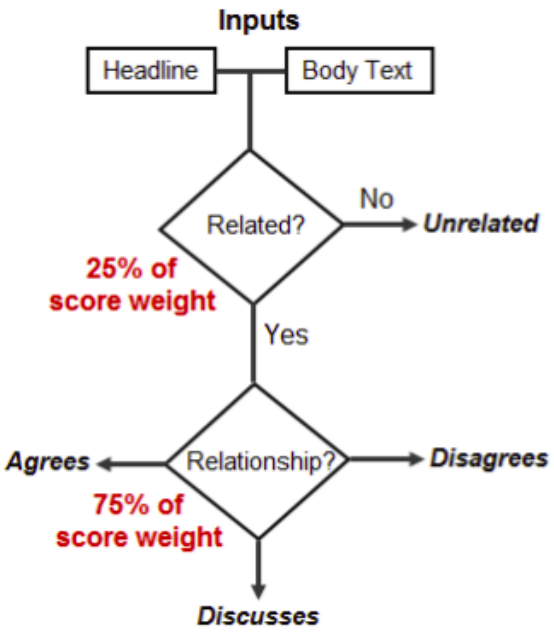

Figure 1: Scoring metric [20]

Figure 1 describes the scoring criteria. Concretely, if a [HEADLINE, BODY TEXT] pair in the test set has the target label unrelated, the evaluation score is incremented by 0.25 if it labels the pair as unrelated.

If the [HEADLINE, BODY TEXT] test pair is related, the evaluation score is incremented by 0.25 if it labels the pair as any of the three classes: agrees, disagrees, or discusses.

The evaluation score is incremented by an additional 0.75 for each related pair if it gets the relationship right by labeling the pair with the single correct class: agrees, disagrees, or discusses.

5 RELATED WORK

In this section we discuss related works. A lot of literature is available for the task of stance detection. We will also discuss some articles which address this specific dataset.

[24] presents a way to do stance detection on twitter data. They try to classify a given tweet and given entities into three classes: tweet being in favor of the given entity, tweet being against the given entity, or unable to detect the stance of tweet towards the given entity, that is, it is classified as neutral. The entity might or might not be present in the tweet. The task also discusses what happens when an opinion is presented in the tweet along with a detailed discussion on intricacies of stance detection. Also, relationships between stance detection and sentiments are explored, sentiment being an important factor in Natural Language Processing tasks. All the tweets which could not be classified in favor or against the entity are put in the neutral class. The given dataset is also expected to contain small neutral classes. Also, all the tweets in neutral class are not completely neutral, which means that they might be just

unrelated to the target entity, and are neither in favour nor directly against the entity. The task of stance detection is somewhat related to the task of sentiment analysis. In sentiment analysis we classify text as positive, negative or neutral. But, there is no explicit entity being targeted in the text. This is one major difference between the two tasks. Also, in stance detection the target entity may or may not be present in the tweet and on top of that it may not even be the subject of opinion in the given tweet. A simple example is :

Entity: Bernie Sanders
Tweet: Ann Coulter oops I mean #JoyReid has stooped to Trumps level of bashing against the communist and his wife.

As can be seen from the above example, the entity of opinion is Donald Trump but the target is Bernie Sanders who has not been named explicitly in the tweet.

The task was solved using two frameworks, one of which is supervised and the other is a weakly supervised framework. Macro average F1 score serves as an evaluation metric. They also evaluate a lot of different frameworks for stance detection with baselines being Majority Class, SVM with ngrams and SVM with unigrams. Best accuracy achieved was 71.66%.

[10] presents a conditional long-short term memory (LSTM) encoding for twitter stance detection for the data provided by [24]. They augment it with bidirectional encoding to get better results. This stance detection has been proved very useful by Mendoza et al. [14] for fact checking. They argue that tweets which are related to each other can be used to affirm facts 90% of the time and at the same time recognizing fake news or misinformed tweets effectively. Hence, stance detection proves to be useful in both human augmented fact checking and automatic fact checkers. They train their model in two ways. First, making the model learn stances towards entities not present in the tweets and second, learning a model with respect to the entity without labeled training data. A neural network architecture is proposed based on conditional encoding given by Rocktaschel et al. [26]. This has been a source of inspiration on how we can use stance detection for detecting fake news using the FNC data [20]. A macro score of 0.58 is achieved by this model on the given data set.

Multiple encoding approaches are proposed to combine stance entity with tweets. One of them is independent encoding which solely relies on non-linear projections for adding entities in stance detection. Additionally, bidirectional conditional encoding has been proposed which will consider the context of both the words on the left and the right side of the word being considered for entity-dependent data representations. An unsupervised method has been used to generate more labelled data specific to the entities available in the training data. We realized that this step was really important for the performance of an LSTM. Multiple models have been run on the three versions of conditional encoding namely:

- entities conditioned on tweets
- tweets conditioned on entities
- bidirectional encoding model

The models tested include:

- SVM with n-gram feature combination
- majority baseline
- a bag of words and vectors approach
- independent encoding of tweets and entities
- dependent encoding of tweet data and entity

It is observed that for the task of stance detection on unknown data, learning a model with entity dependent encoding works better than any other model available. But the most notable result which inspired us to use features like polarity and refutation apart from the conditioned encoding is the performance of bag of words vector representation of the headline and article concatenated with these global features.

Comparing the above to the task of weakly supervised stance detection, one must understand the differences in the task setup. In this task training data has already been automatically annotated with the entities. For example, tweets such as *"Oh wow... come on Venezuela! We are rooting for you !!"*, have been annotated with Bernie Sanders given his recent interviews regarding the same. All the baselines for the task fail to achieve an F1 score anywhere close to the state of the art stance detection corpus. The average score is 0.29, which shows difficulty of the task at hand. Things are worsened if we try to incorporate feature extraction into this system, the F1 score falls rapidly. Only when a bi-conditional LSTM is used results closer to the best available results are realized, implying a mean F1 score of 0.59. This work derives from the task of textual entailment Rocktaschel et al. 2016 [26]. We will develop further on this method in later parts of this paper. Another important thing to realize is that both in tweets and fake news, the task of stance detection is much harder than text corpora from debates, news, books and other reliable sources, in addition to the use of irregular and language intended to deceive, the context is mostly missing in data set.

Kim [12] presents convolutional neural networks for the task of sentence classification. This work is important for many reasons: first, the author uses static word vectors and pre-trained word vectors, both individually and in combination, in conjunction with convolutional neural networks. This with a little nudging of hyper parameters beats the best models for sentence classification in sentiment analysis and question answering. Deep learning methodologies in natural language processing are mostly used to find word embeddings via the use of neural network models [27]. In essence it is a projection from a sparse 1-V dimensional space, where V is the number of words in vocabulary, to a lower dimensional space via hidden layers. The good thing why this works is because the semantic meanings are preserved after the hidden layer encoding. So the words which were similar or close in original space maintain that relationship. Convolutional neural networks similar to computer vision are used for these extracted features, showing promise to solve natural language processing tasks such as natural language based querying, machine translation and other such tasks.

The convolutional neural network architecture is simple, see Figure 2 . There are 2*m matrices of size n*k, where m is the number of sentences, n is the number of words and k is the word vector (padding was used to make each vector of same length). Each matrix has a counterpart, the first one corresponds to static word vectors without any context and the second one entails context. The second to last layer applies multiple convolutional filters for extraction of different features. Multiple width filters and feature maps are used

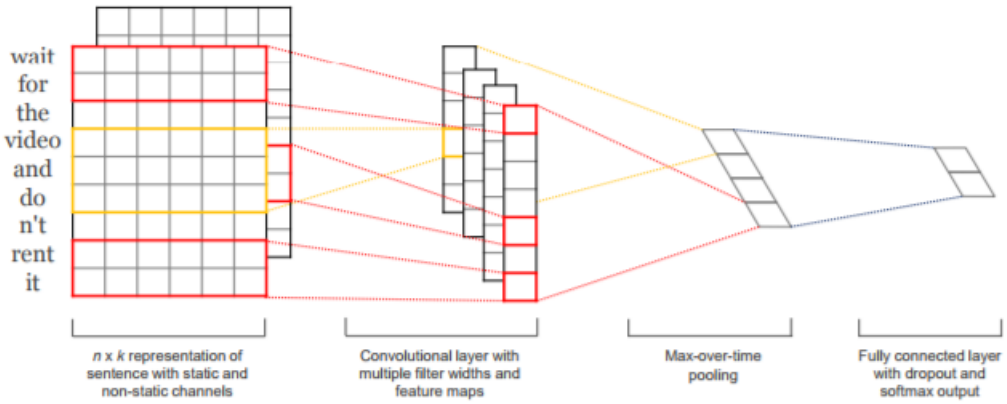

Figure 2: Model architecture with two channels for an example sentence [12]

to get a variety of features. To down sample the input representation, max-pooling , a sample based discretization process is used. This reduces dimensionality and prevents overfitting by providing an abstracted representations of features. As usual in convolutional neural network architectures there is a fully connected layer in the last with softmax output and dropout, where randomly selected neurons are selected neurons are ignored during training [16]. This model has been trained and tested on multiple datasets:

- database of movie reviews and classifying them as positive or negative. (Pang et al.) [3]
- stanford sentiment treebank (SST-1), this data has more granular labels allowing classification as very positive, positive, neutral, negative and very negative. (Manning et al.) [23]
- stanford sentiment treebank - 2 (SST - 2), same as previous but just has binary labels and no neutrality.
- objectivity dataset - which helps determines whether the data is objective or subjective in nature. (Pang et al.) [2]
- customer review database - containing opinions of customers on various products and features. (Hu et al.) [13]
- MPQA dataset - helps in detection of opinion stance. (Wilson et al.) [11]

These datasets are important as sentence classification on these can provide us with a rich set of features (classified sentences) from variety of corpora and provides scope for future work in stance detection in fake news for previously unseen data, since the given FNC-1 training data is not exhaustive in it's topic coverage.

ReLU has been used for parameter tuning. word2vec has been used to provide initial word vector in lieu of lack of large training data ((Collobert et al., 2011) [22]. Model variations used are:

- CNN-rand: words with random initialization, updates occur with epochs
- CNN-static: words from word2vec, only model parameters tuned, initialized word remains static
- CNN-non-static
- CNN-multichannel: two set of vectors from word2vec used for initialization, there are two channels but only one of them is used for back propagation of information, keeping the other set of words static.

Randomly initialized word of vectors model does not perform well at all, while the hypothesis stating - *using already trained word of vectors will achieve better results* is proved wrong. Also, multichannel performs better than other models since it uses a culmination of features. All the previous benchmarks are beaten with a best accuracy of 93.0%. Also non-static channel is able to club features which are closer to each other in semantic space much better than the static channel. For example: for the word 'bad', static channel finds out the synonyms as *good,terrible,horrible, lousy etc.* whereas good is out of context. Backpropogation in non static channel updates the same set of similar words to *terrible, horrible,lousy and stupid*, which seems to be much closer to bad in original semantic space. This step has been tested and used by multiple researcheres, where they use pre-trained set of vectors to boost their model performance, with them even a single layer of CNN or multi-layer perceptron achieves tremendous results.

A slightly similar problem of stance detection of a post written for a two sided debate (for or against) in an online forum is addressed by Kazi and Vincent [8]. Their idea is to explore learning based stance classification systems and assess their performance under different settings such as by varying the amount and quality of training data, changing the model complexity, varying the feature set quality and introducing certain extra linguistic constraints. The difficulty of this problem is in some sense similar to stance detection of article and headline pairs because of the use of insults, sarcasm and concerns over the credibility of other posts. They compared the performance of generative models like Naive Bayes with discriminative models like Support Vector Machine and observed that both of them perform equally well. They even experimented with fine-grained models which involved tweaking the model to take into account the stance of each of the sentences present in the post and concluded that a fine-grained model performs better

than a coarse-grained model. Another interesting takeaway from their work is that an increase in the amount of training data (with the help of an unsupervised model) provides a performance boost even though this training data might be noisily labelled. They compared the performance of various models by using different kinds of feature sets such as n-gram and the ones proposed by Anand et al. [19] and observed that the latter did not provide a significant improvement.

Another similar problem in hand is recognizing textual entailment (RTE) which is essentially the problem of determining whether from a given pair of hypothesis and premise if the hypothesis can be inferred from the premise. There are 3 possible outcomes - entailment (inferred to be true), contradiction (inferred to be false) and neutral (truth unknown). Most deep learning approaches to solve RTE can be classified under two groups - sentence-encoding based (as the name suggests sentence-encoding is the crux of the approach) and matching-encoding based (these approaches directly model relation between two sentences and generate sentence representations). Amond sentence-encoding based approaches, there are LSTMs based models, GRUs-based models, TBCNN based models and SPINN based models. Liu et al. [28] propose a unified deep learning framework for RTE based on bidirectional LSTMs. Their main idea involves a two stage process. First, use average pooling over word level biLSTM to generate a first stage stance detection (this provides an intuition about the context of the sentence) and then use an attention mechanism to replace average pooling on same sentence for better representation. The reason they do this is because average pooling assigns each word an equal importance and the attention mechanism help re-weight words according to their importance (Nones, Verbs and Adjectives are assigned more importance). Their work compares the performance of biLSTMs with single directional LSTM based , GRUs-based, TBCNN based and SPINN based models. biLSTM uses both previous and future context by processing the sequence in two directions which addresses the problems of models like single directional LSTMs and GRUs-based since they don't utlize contextual information from future tokens and CNN based models which don't make full use of info contained in the word order. They extend their model by using Inner Attention on both sides which helps generate more accurate and focused sentences for classification. They even introduce a very simple and effective input strategy to remove same words from hypothesis and premise which improves the accuracy.

There have been several other works which indicate that attention improves the performance of natural language processing task by selectively focusing on words present in the input. Xiong et al. [4] define a dynamic coattention network for question answering. The model first fuses co-dependent representation of the question and document to focus on their important aspects. A coattention encoder attends to both the question and document encoding simultaneously and fuses their attention contexts. This is achieved by first computing the affinity matrix then an affinity score corresponding to all pairs of document words and question words. The matrix is then normalized row-wise to generate attention weights for each word present in the question and normalized column-wise to produce each attention weights for each word present in the document. These weights are then used to compute attention contexts of the document, and also the attention context of the question in consideration with respect to each word of the document. A co-dependent representation of the question and document attention contexts is what defines the coattention context which is then fused with the temporal information via a biLSTM. The reason we mention this technique is because we can employ this coattention context to better train our model. Two more attention based approaches have been evaluated by Luoang et al. [15] to accomplish the task of neural machine translation. These differ in terms of whether the attention is placed on all source positions or only a few source positions. They describe a global approach (always attend to all source words) and a local one (only attend to a subset of source words at an instant). The global approach has the drawback of attending to all words on the source side for each target word which is expensive and potentially impractical. They also show that the local attention approach outperforms non attentional architectures with a significant gain and also beats the global attention technique.

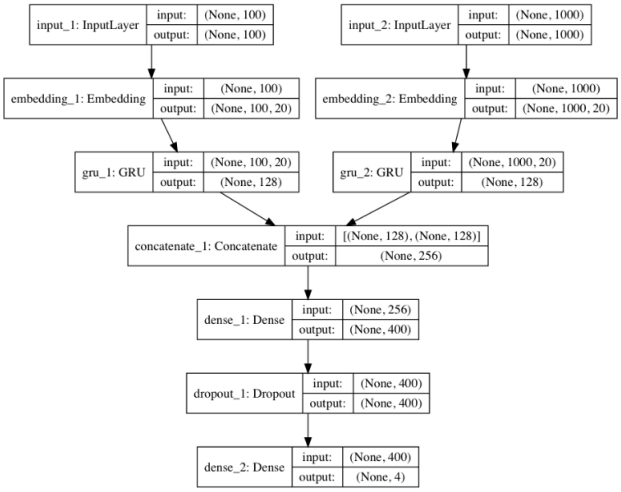

Figure 3: Dual GRU [6]

There has been some other work using neural networks to solve the problem in question. Richard et al. [6] experimented with four Neural Network based models, two of which are based on feed forward architecture and two on recurrent architecture. These are:

- Concatenated multi-layer perceptron (Concat MLP) for truncating and tokenizing the headline and article, these tokens are then passed through sentence embedding matrices. The output of these matrices through several dense hidden layers with dropout and then finally separate softmax layers to make the two separate kinds of classification which the problem demands - a binary classification to determine if the headline and article pairs are related or not and then the stance of the article with respect to the headline.
- The Dual GRU based approach that processes the headlines and articles separately which is then fed to a sentence embedding layer. Each embedding is then passed on to a recurrent neural network based on gated recursive units. Figure 3 describes the GRU based approach.

- A bag-of-words multi-layer perceptron which tokenizes headline and articles using a fixed sized vocabulary of highly frequent tokens. These tokens are then reduced to a bag-of-words vector to reflect the token distribution which is then passed through several hidden layers with dropout. Parallel softmax layers were then used as in the case of Concat NLP to perform the final classifications.
- A bi-directional concatenated stacked LSTM (Bi-dir LSTM) model based on the idea that the beginning of the headline and the article are linked and that concatenating them would allow the recurrent neural network to learn better. Similar to Concat MLP, they used parallel softmax layers for the two kinds of classification (related/unrelated and discuss/agree/disagree).

As per them, the best model was a bag-of-words followed by a three-layer multi-layer perceptron (BoW MLP).

The MultiGenre Natural Language Inference corpus (MultiNLI) [1] used to evaluate the performance of recognizing textual entailment also known as natural language inference is an extremely relevant dataset for the problem in question. It consists of pairs of sentences consisting of a premise and a hypothesis. This dataset contains two different categories of tuples - standard in-domain or the matched set in which the test and train data are from the same source and cross-domain or unmatched set in which the train and test data differ a lot. The reason to include the unamtched set is to test a model's ability to learn representation of sentence meaning that capture broadly useful features. This dataset consists of pairs of sentences from a wide range of genres of spoken and written English. The dataset is balanced across all three labels. In each pair, the premise has been derived from one of ten sources of text which have been compiled from ten different genres.

Nikita et al [18] evaluate the performance of various natural language processing models based on sentence encoders on the MultiNLI corpus. The models they compare are all based on BiLSTM. In terms of depth of neural network, they asses that Chen et al's model [21] which uses a three layer bidirectional RNN performs significantly better compared to other models which employ a single layer RNN. They also observe that the use of shortcut connection between recurrent layers (done to ease gradient flow) shows an improved performance. Vu et al. [9] use pre-trained GloVe word embeddings augmented with other features. Their model creates sentence embeddings for part-of-speech (POS), character level info, dependency relation between a word and its parent and then concatenate it with the embeddings for each word, this shows small but non trivial improvement in the matched setting. They also demostrate that the use of max pooling over average pooling when colleting the hidden states of BiLSTM for use as sentence representation enhances the overall performance. However, max pooling isn't that effective when used along with intra-sentence attention. Nikita et al. also notice that every model uses uses elementwise product and difference features, comparing the two sentence encodings as part of the input to the classifier MLP that predicts the final relation label, this technique is observed to provide a performance boost. Upon measuring the cosine similarity of sentence embeddings of a random sample from the matched set for all models and computing the number of times the first nearest neighbor belongs to the same genre as the chosen sentence, all models demonstrate good performance. This indicates that the learned representations are not genre-agnostic even though they perform well on unseen genres.

[17] Niall et al. discuss various approaches for assessment of veracity of texts broadly based on two categories - linguistic cue approaches and network analysis approaches. They even mention how hybrid approaches combining the two are promising. Techniques employing linguistic cues are based on the intuition that language domain helps to identify deceptive content. Extracting and analyzing text to associate language patterns with deception and then use these patterns to build models to detect deceptive content. The intuition behind using linguistic cues is that most liars tend to have a pattern. No matter how hard they control there always tends to be some language leakage within certain verbal aspects such as pronouns, conjunctions and negative word usage and the linguistic approaches help spot these leakage instances. These approaches involve leveraging the representation of data using techniques such as bag of words which analyze and aggregate the frequencies of words to reveal cues of deception. Often times data representation is not enough and we need to analysze deeper language structures to identify deception instances. Semantic analysis which requires incorporating profile compatibility features are also known to provide better results since someone who doesn't know about the content is highly likely to include contradictions or omit important facts. Prominent use of rhetorical relations is also indicative of deception. On the other hand, network analysis approaches require harnessing of message metadata or structured knowledge queries to retrieve aggregate deception measures. Both these approaches make use of machine learning techniques to build classifiers to accomplish the task in hand.

6 MODEL ARCHITECTURES

Based upon initial experiments, we use different neural net architectures that give good results. These models are described below.

6.1 Baseline

FNC-1 provides us with a baseline model which uses a gradient-boosting classifier over global features such as n-gram co-occurrence between the headline and article, hand-tuned feature sets such as appearance of a provided set of highly-polarized words (e.g. "fraud" and "hoax"). The baseline model has an accuracy of 79%. Our primary objective was to beat the baseline with a new architecture.

6.2 FNC-1 Winner

The winners of the FNC-1 challenge implemented several different models, and found out that their results were the best when they combined several of these models in an ensemble. Their final model was an ensemble based on a 50/50 weighted average between gradient-boosted decision trees and a deep convolutional neural network.

6.3 TF-IDF Representation with Global features into an MLP

We first develop a TF-IDF representation of the concatenated headline and body. This representation is appended with the global

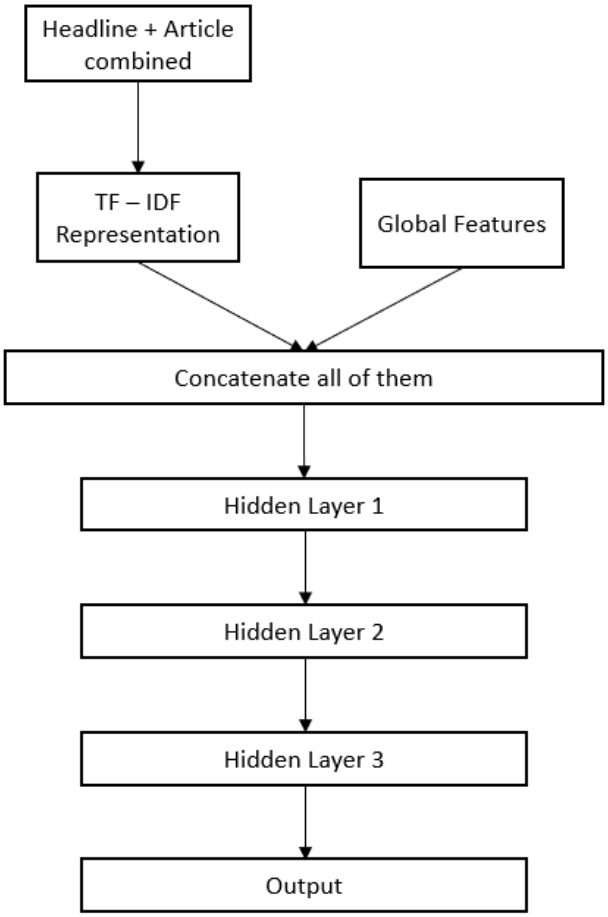

Figure 4: Architecture of the TF-IDF Representation with Global features into MLP

features which is a combination of word overlap, refuting features, polarity features and hand features. This combined representation is used as an input to our MLP system which we concluded to be ideal at three hidden layers after a lot of experimentation. In addition, our loss function gives a higher weightage to the agree, disagree and discuss classes as well. (82.87% accuracy by giving more weightage to the related classes)

6.4 Using Sentence Encoding for headline and body to predict

For this task, we first develop a sentence embedding for both the headline and the body. The sentence embedding is a fixed size representation of the sentence which is later used as an input to our MLP Layer. The sentence embedding is generated using a pre-trained bi-directional LSTM encoder which takes as input a sentence and outputs a sentence embedding. We now call u as the sentence embedding for the headline and v as the sentence embedding for the body. After this we use a concatenation of u, v, abs(u-v), u*v as our input to the MLP and generate our results accordingly. We get a very low accuracy for this architecture which is described in the table.

One of our conclusions for this system not working well enough is that the system is unable to learn appropriate sentence embeddings using the pre-trained GloVe word embedding which is a general purpose word embedding.

We also use some of the other encoders for sentences that are based on LSTM's and Gated Recurrent Units (GRU) but none of them work well for our task.

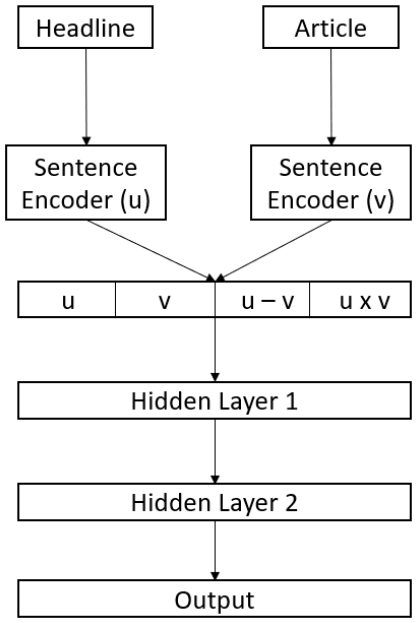

Figure 5: Architecture for the system using a pre-trained sentence encoder for headline and body to predict

6.5 Concatenating Global features, TF-IDF Representation and Sentence embeddings

One of our final experiments was merging Global features, TF-IDF Representation of the concatenated body and headline along with sentence embeddings of the headline and the body. So first we use the TF-IDF representation of the concatenated headline and body in a similar fashion to part 7.1, after this step we use the pre-trained infersent model to build the sentence embeddings for each sentence. Finally we append these two with the global features and input this into an MLP Layer. However, we do improve on the result in part 7.1, so we plan to present part 7.1 in detail in this paper.

6.6 Weighting Rare Classes

In this approach, we run several standard classifiers by giving more weightage to examples that are present in very small amounts like the agree and disagree labels. We insert multiple instances of the same agree and disagree labels while training our models. We notice that some of the classifiers show an increase in their accuracy just by adding more training examples for the agree and disagree labels.

Table 2: Accuracy values by using only global features for different models with weighted sampling

Model	Accuracy
Baseline	79.1%
Decision Tree	79.15%
Random Forest	76.04%
MLP	76.23%
K Neighbors	74.52%
AdaBoost	78.18%
Gaussian NB	68.91%
SVC with linear kernel	79.64%
Gradient Boosting	79.94%

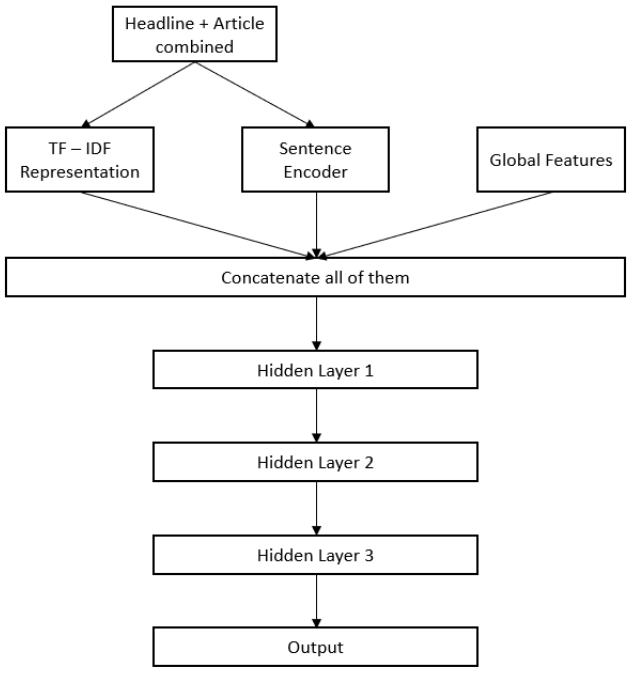

Figure 6: Architecture for concatenated Global features, TF-IDF Representation and Sentence embeddings as an input to the MLP Layer

Table 3: Accuracy values for neural network models as compared to the baseline

Model	Accuracy
Baseline	79%
FNC-1 Winner	82.02%
Sentence Encoder	67.45%
Concat Global, TF-IDF and Sentence Embedding	80.1%
TF-IDF with Global Features	82.8%

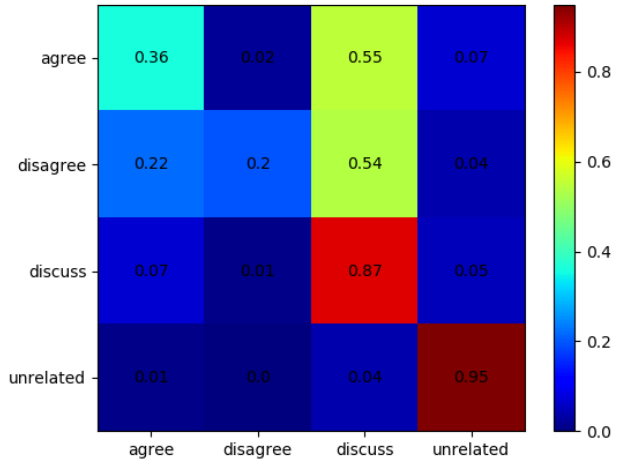

Figure 7: Confusion matrix for the test set for the model: TF-IDF Representation with Global features into MLP

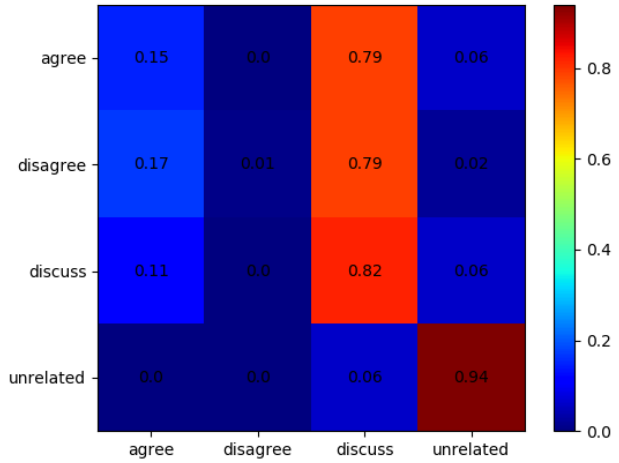

Figure 8: Confusion matrix for the test set with regularization (dropout = 0.5) for the model: TF-IDF Representation with Global features into MLP

7 EXPERIMENTS

During the course of the project, we ran several experiments on our dataset. By Tuning our hyper parameters we were able to improve most of our models but the TF-IDF Representation with the global features as an input to the MLP worked the best. Now, I would like to describe the various experiments we conducted to tune our model. In all the experiments, we do cross-validation. First we decided we were going to represent the combined headline and article into its TF-IDF Representation. We choose the vocabulary size of of 5000 as

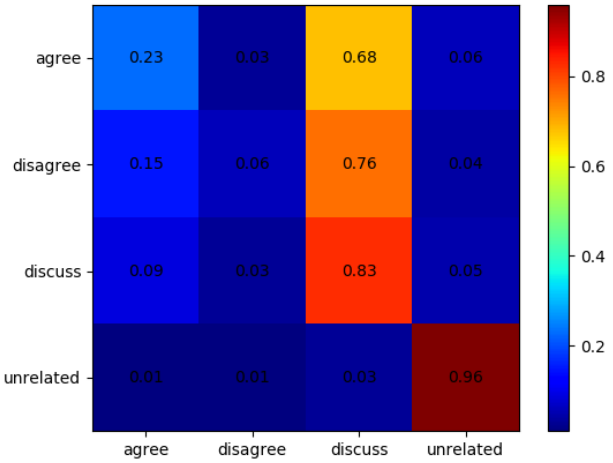

Figure 9: Confusion matrix for the test set for the best model trained by giving higher weight to rare samples (Gradient Boosting)

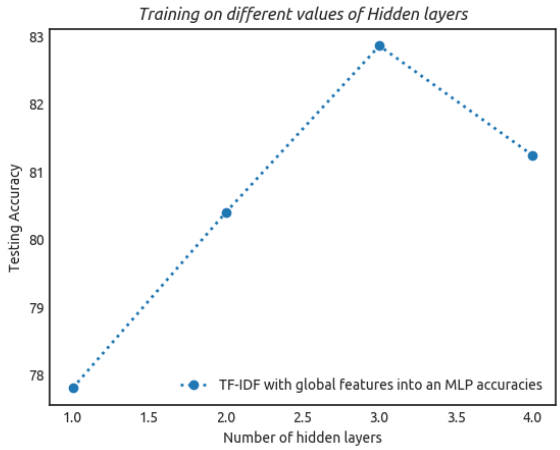

Figure 10: Training on different values of hidden layers without dropout (Number of nodes for different layers: 600, 600, 300, 300)

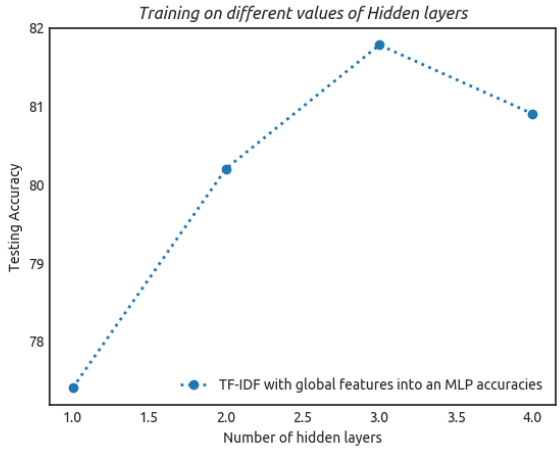

Figure 11: Training on different values of hidden layers with dropout = 0.5 (Number of nodes for different layers: 600, 600, 300, 300)

it produces the best results as shown in the graph in Figure 12. We then decided what should be the number of layers in our MLP and we concluded that the total number of hidden layers should be 3 as it produced the best results and the accuracies reduced slightly and didn't improve as we increased the number of layers. The graph for the same is shown in Figure 10. We also ran the above tests by adding a dropout of 0.5, however we couldn't improve on our results as shown in Figure 11. Finally we ran our model for different number of nodes in the hidden layers and concluded that 600 nodes in the first layers, 600 in the second and 300 in the third worked the best.

8 RESULTS

In order to generate our results, we first tune the hyper parameters and show the results of our three best models on the test set in table 3. The test set is different from the train set and it is only used to evaluate the performance of our model.

The model that performed the best was the TF-IDF Representation with Global features into an MLP. This model was able to achieve an accuracy of 82.8% beating the baseline by 3.8% . The hyper parameters for this model are, vocabulary size of 5000 words for the TF-IDF representation and three hidden layers of size 600, 600 and 300 respectively. The confusion matrix for this configuration is shown in Figure 7. The confusion matrix for the results with a dropout of 0.5 for each layer is shown in Figure 8. The overall accuracy reduces to a value of 81.78% after using dropout. So we can say our final model doesn't implement dropout as the accuracy reduces after using dropout.

Furthermore, we also ran several standard machine learning models such as SVM, Naive Bayes, Gradient Boosting after allocating a high weighting to rare samples and we get the FNC-1 scores as shown in Table 2. The gradient boosting technique achieves the maximum accuracy of all these classifiers and the confusion matrix for this shown in Figure 9.

9 CONCLUSIONS

Using the TF-IDF Representation of the combined headline and the body along with the global features as an input to our MLP. we beat the baseline on the FNC-1 score by 3.8%. Assuming the fact that the baseline was already achieving an FNC-1 score of 79%, we think 3.8% is a considerable improvement. Furthermore, I would like to point out that we were also able to beat the Winner of FNC-1 Challenge by 0.8% which we believe is a considerable achievement for our model.

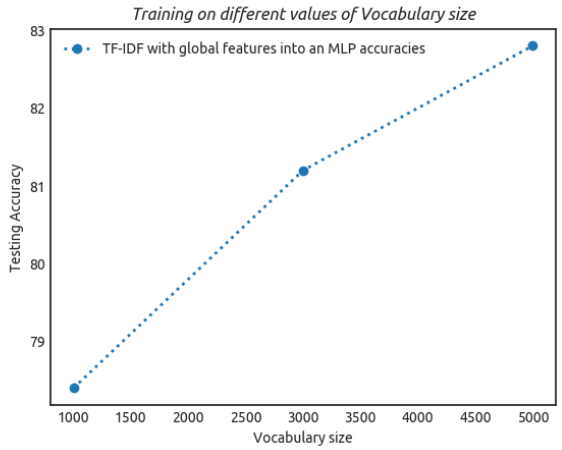

Figure 12: Testing accuracies for different values of the vocabulary size for the TF-IDF representation

After implement several variations of recurrent neural networks namely Bi-directional LSTM's, LSTM's, GRU's, we conclude we can not beat the baseline by hyper parameter tuning using only these models for classification. This came in as shock as recurrent neural networks have been achieving state-of-the-art results in a lot of natural-language processing problems. We think RNN's might be under-performing because of the following reasons, there may not be enough data for an RNN to generalize, even though Bi-LSTM's, GRU's and LSTM's remember information it might so happen that they forget important information that the TF-IDF is capturing.

As for the future, we plan improving our currrent TF-IDF with global features model in a lot of ways. One of the first things we would like to do is to train a word embedding specific to the fake news challenge and use the appropriate word embeddings. In addition, we also plan to develop a sentence encoder using the word embeddings we develop from the fake news data. In addition, we also plan to add further hand-written features to our global features in order to improve our accuracy.

Even though, we could not achieve a very high accuracy with the a complicated neural architectures, we successfully achieved our primary goal of improving on the baseline FNC-1 score. We fulfill the target by using a TF-IDF Representation of the combined headline and article along with the global features as an input to our MLP. By tuning our hyper parameters, we were able to achieve an FNC-1 score of 82.8% outperforming the baseline by a full 3.8%. We believe these results are a step in the direction of detecting fake news automatically.

A APPENDIX

A.1 Data Characteristics

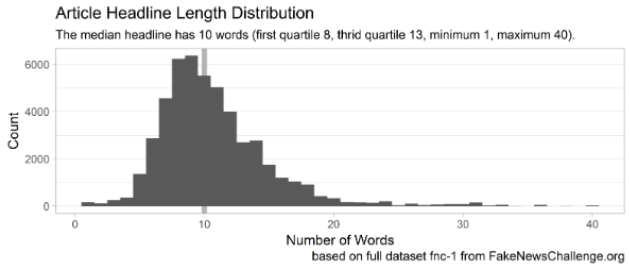

Figure 13: Distribution of article headline lengths[25]

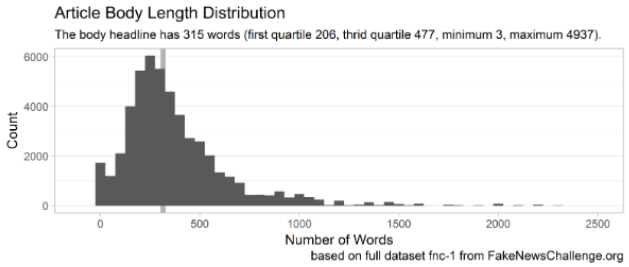

Figure 14: Distribution of article body lengths[25]

ACKNOWLEDGMENTS

The authors would like to thank Professor Jiawei Han and the teaching assistants for guiding us throughout the project.

REFERENCES

[1] Nikita Nangia Adina Williams and Samuel R. Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. CoRR abs/1704.05426.

[2] L. Lee B. Pang. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In Proceedings of ACL.

[3] L. Lee B. Pang. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with re- spect to rating scales. In Proceedings of ACL.

[4] V. Zhong C. Xiong and R. Socher. 2016. Dynamic coattention networks for question answering. arXiv preprint arXiv:1611.01604.

[5] The Conversation. [n. d.]. The real consequences of fake news. ([n. d.]). http://theconversation.com/the-real-consequences-of-fake-news-81179

[6] Richard Davis and Chris Proctor. 2017. Fake News, Real Consequences: Recruiting Neural Networks for the Fight Against Fake News. (2017). https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1174/reports/2761239.pdf

[7] W. Ferreira and A. Vlachos. 2016. Emergent: a novel data-set for stance classification,. In *in Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. ACL.

[8] Kazi Saidul Hasan and Vincent Ng. 2016. Stance Classification of Ideological Debates: Data, Models, Features, and Constraints. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, ACL*.

[9] Xiaoyu Bai Marc Tanti Lonneke van der Plas Hoa Trong Vu, Thuong-Hai Pham and Albert Gatt. 2017. LCT-MALTAâĂŹs submission to RepEval 2017 shared task. The Second Workshop on Evaluating Vector Space Representations for NLP.

[10] A. Vlachos I. Augenstein, T. Rocktaschel and K. Bontcheva. 2016. Stance detection with bidirectional conditional encoding. In *arXiv preprint arXiv:1606.05464*.

[11] C. Cardie J. Wiebe, T. Wilson. 2005. Annotating Expressions of Opinions and Emotions in Language. Language Resources and Evaluation, 39(2-3): 165âĂŞ210.

[12] Y. Kim. 2014. Convolutional neural networks for sentence classification. In *arXiv preprint arXiv:1408.5882*.
[13] B. Liu M. Hu. 2004. Mining and Summarizing Cus- tomer Reviews. In Proceedings of ACM SIGKDD.
[14] Barbara Poblete Marcelo Mendoza and Carlos Castillo. 2010. Twitter Under Crisis: Can We Trust What We RT?. In *In Proceedings of the First Workshop on Social Media Analytics (SOMAâĂŹ2010), pages 71âĂŞ79, New York, NY, USA*. ACM.
[15] Hieu Pham Minh-Thang Luong and Christopher D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. arXiv preprint arXiv:1508.04025, 2015.
[16] A. Krizhevsky I. Sutskever R. Salakhutdinov N. Srivastava, G. Hinton. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. In *Journal of Machine Learning Research*.
[17] Yimin Chen Niall J. Conroy, Victoria L. Rubin. 2015. Automatic deception detection: Methods for finding fake news. Proceedings of the Association for Information Science and Technology.
[18] Angeliki Lazaridou Nikita Nangia, Adina Williams and Samuel R. Bowman. 2017. The RepEval 2017 Shared Task: Multi-Genre Natural Language Inference with Sentence Representations. arXiv:1707.08172v1 [cs.CL] 25 Jul 2017.
[19] R. Abbott J. E. Fox Tree R. Bowmani P. Anand, M. Walker and M. Minor. 2011. Cats rule and dogs drool!: Classifying stance in online debate. In *WASSA*.
[20] D. Pomerleau and D. Rao. 2016. Fake news challenge. (2016). http://www.fakenewschallenge.org/
[21] Zhen-Hua Ling Si Wei Hui Jiang Qian Chen, Xiaodan Zhu and Diana Inkpen. 2017. Recurrent neural network-based sentence encoder with gated attention for natural language inference. The Second Workshop on Evaluating Vector Space Representations for NLP.
[22] L. Bottou M. Karlen K. Kavukcuglu P. Kuksa R. Collobert, J. Weston. 2011. Natural Language Processing (Almost) from Scratch. Journal of Machine Learning Research 12:2493âĂŞ2537.
[23] J. Wu J. Chuang C. Manning A. Ng C. Potts R. Socher, A. Perelygin. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Tree- bank. In Proceedings of EMNLP 2013.
[24] P. Sobhani X. Zhu S. M. Mohammad, S. Kiritchenko and C. Cherry. 2016. Semeval-2016 task 6: Detecting stance in tweets. In *in Proceedings of SemEval*.
[25] Ferdinand Legros Stephen Pfohl, Oskar Trieb. [n. d.]. Stance Detection for the Fake News Challenge with Attention and Conditional Encoding. ([n. d.]). https://web.stanford.edu/class/cs224n/reports/2748568.pdf
[26] Karl Moritz Hermann TomaÂťs KoËČciskËČy Tim Rocktaschel, Edward Grefenstette and Phil Blunsom. 2016. Reasoning about Entailment with Neural Attention. In *In International Conference on Learning Representations*. ICLR.
[27] P. Vincent Y. Bengio, R. Ducharme. 2003. Neural Probabilitistic Language Model. In *Journal of Machine Learning Research 3:1137âĂŞ1155*.
[28] Lei Lin Yang Liu, Chengjie Sun and Xiaolong Wang. 2016. Learning Natural Language Inference using Bidirectional LSTM model and Inner-Attention.