FAKE NEWS CHALLENGE STAGE 1 : STANCE DETECTION

Sushmitha Suresh,
Department of Management Sciences,
University of Waterloo

Abstract

The goal of the Fake News Challenge is to explore how artificial intelligence technologies, particularly machine learning and natural language processing, might be leveraged to combat the fake news problem Assessing the veracity of a news story is a complex and cumbersome task, even for trained experts. Fortunately, the process can be broken down into steps or stages A helpful first step towards identifying fake news is to understand what other news organizations are saying about the topic We believe automating this process, called Stance Detection, could serve as a useful building block in an AI assisted fact checking pipeline So stage 1 of the Fake News Challenge (FNC 1) focuses on the task of Stance Detection. The baseline model achieved an accuracy score of 79.53% . With this baseline accuracy I went on try three simple classifiers and 4 recurrent neural networks. A simple logistic regression classifier gave an accuracy of 85.46% when ran on the unlabeled data. Considering this as my baseline model, I built two LSTMs, a unidirectional LSTM and a Bi-directional LSTM. Each of the LSTM implemented in W2V ad GloVe gave higher accuracy scores than Logistic Regression. The Bi-LSTM gave the highest accuracy of 92% among the other neural networks.

1 Introduction

Fake news, defined by the New York Times as "*a made-up story with an intention to deceive*", often for a secondary gain, is arguably one of the most serious challenges facing the news industry today. In a December Pew Research poll, 64% of US adults said that "*made-up news*" has caused a "*great deal of confusion*" about the facts of current events.

The larger problem, experts say, is less extreme but more insidious. Fake news, and the proliferation of raw opinion that passes for news, is creating confusion, punching holes in what is true, causing a kind of fun-house effect that leaves the reader doubting everything, including real news.

News that is fake or only marginally real has lurked online - and in supermarket tabloids - for years, but never before has it played such a prominent role in an American election and its aftermath. Narrowly defined, "fake news" means a made-up story with an intention to deceive, often geared toward getting clicks. But the issue has become a political battering ram, with the left accusing the right of trafficking in disinformation, and the right accusing the left of tarring conservatives as a way to try to censor websites. In the process, the definition of fake news has blurred.

While fake news became an issue during the highly charged 2016 presidential election campaign, Republicans and Democrats are about equally likely to say that these stories leave Americans deeply confused about current events. About six-in-ten Republicans say completely made-up news causes a great deal of confusion (57%), and about the same portion of Democrats say the same (64%). While a majority of those who make less than $30,000 a year say fake news causes a great deal of confusion (58%), this is a lower proportion than among those who make between $30,000 and $75,000 (65%) and those who make $75,000 or more (73%).

2 Problem Description

2.1 Fake News Challenge

The goal of the **Fake News Challenge** is to explore how artificial intelligence technologies,

particularly machine learning and natural language processing, might be leveraged to combat the fake news problem. AI technologies hold promise for significantly automating parts of the procedure human fact checkers use today to determine if a story is real or a hoax.

Assessing the veracity of a news story is a complex and cumbersome task, even for trained experts 3. Fortunately, the process can be broken down into steps or stages. A helpful first step towards identifying fake news is to understand what other news organizations are saying about the topic. Automating this process, called **Stance Detection**, could serve as a useful building block in an AI-assisted fact-checking pipeline. So, stage #1 of the **Fake News Challenge (FNC-1)**focuses on the task of Stance Detection.

Stance Detection involves estimating the relative perspective (or stance) of two pieces of text relative to a topic, claim or issue. The version of Stance Detection we have selected for FNC-1 extends the work of Ferreira & Vlachos 4. For FNC-1 we have chosen the task of estimating the stance of a body text from a news article relative to a headline. Specifically, the body text may agree, disagree, discuss or be unrelated to the headline.

2.2 Competetion's Baseline Model

A simple baseline model was developed using hand-coded features and a GradientBoosting classifier. The baseline implementation also included code for pre-processing text, splitting data carefully to avoid bleeding of articles between training and test, k-fold cross validation, scorer, and most of the crud to experiment with this data. The hand-crafted features included word/ngram overlap features, and indicator features for polarity and refutation. With these features and a gradient boosting classifier, the baseline achieves a weighted accuracy score of 79.53% with a 10-fold cross validation.

3 Related Work

3.1 Winner of the competetion : SOLAT in the SWEN

The team SOLAT IN THE SWEN decided to test how various cutting-edge machine learning techniques performed. After successfully implementing several different models, the team found that their results were best when combining multiple models in an ensemble. The team's final submission was an ensemble based on an 50/50 weighted average between gradient-boosted decision trees and a deep convolutional neural network. The team achieved an accuracy score of 82.02%.

3.2 First runner up of the competetion : Athene (UKP Lab)

The team used a we used the multilayer perceptron with bag-of-words features. A random search was carried out to optimize the hyper parameters. In order to further improve performance, an ensemble method consisting of 5 multilayer perceptron was used, whereby the labels had been predicted by hard voting. The team achieved n accuracy score of 81.97%.

3.3 Second runner up of the competetion : UCL Machine Reading

The teams work was based on a single, end-to-end system consisting of lexical as well as similarity features passed through a multi-layer perceptron with one hidden layer. The team used two simple bag-of-words representations for the text inputs: term frequency (TF) and term frequency-inverse document frequency (TF-IDF). The representations and feature thus extracted from the headline and body pairs consist of only the following:

- The TF vector of the headline;
- The TF vector of the body;
- The cosine similarity between the TF-IDF vectors of the headline and body.

The team was able to achieve an accuracy of 81.72%.

4 My Methodology

I developed three basic classifiers and four recurring neural networks generating seven different accuracy scores, each of the performance not varying drastically from one another. However, there were some notable and significant changes in the performance when RNNs were used.

4.1 Model 1 : Classifiers

I used three most basic yet effective classifiers : Logistic Regression, Random Forrest Classifier and Multinomial Naïve Bayes Classifier. My preprocessing of texts involved Tokenization , Normalization, Capitalization, Non-alphanumeric removal and Stemming-Lemmatization. The features that were extracted were **TF-IDF Vectors, Cosine Similarity and Word Overlap.**

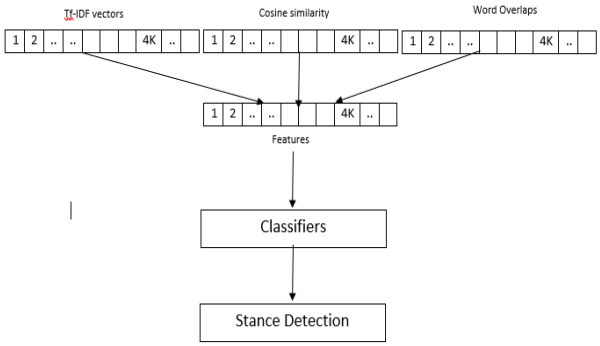

Fig 1: Classifier Architecture

Classifiers	Accuracy in %
LogisticRegression	85
RandomForestClassifier	77
MultinomialNB	56

Table 1: Qualitative Analysis of Classifiers.

While Naïve Bayes is expected to perform well in binary classification, our target variable has four different classes. In addition, the classifier makes a naïve assumption of absence of multicollinearity among the features used for prediction. Hence it is not very surprising that NB gave the least accuracy score.

RF is a versatile algorithm and be expected to outperform LR on many medium-sized tasks. It can handle categorical and real-valued features with ease—little to no preprocessing required. With proper cross-validation technique, they are readily tuned.

However, the increased accuracy for Logistic Regression can be attributed to the fact that for models with millions of sparse features, logistic regression will be much faster to train and execute and is less prone to overfitting.

Keeping Logistic Regression as my baseline model I went on build neural networks. And as expected the LSTMs performed much better than the classifiers with an increased accuracy of 90%.

4.2 Preprocessing

Pre-processing involves removal of punctuations and converting text to lower case. The headlines and bodies were each converted to sequences of words using tokenizer and the sequence of words were in turn converted to sequence of indices. The stances agree, disagree, unrelated and discuss were encoded as 1, 2, 3, and 4 respectively.

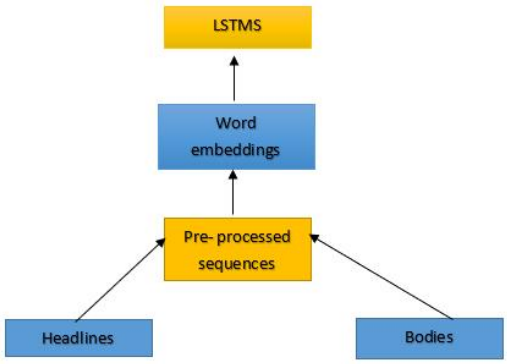

Fig 2 : Preprocessing steps

Then I went on to create the word embeddings either with W2V or Glove and check for vocabulary size of the embeddings to set the input dimensions right. An embedding matrix was created containing words in my vocabulary (the preprocess sequences from the previous step). For those words not in the embedding matrix, random initialization was used.

This pre-processing method remined the same for all the four models. However, the generation of word embeddings using the preprocessed indices differed with every model.

4.3 Model 2 : Uni-directional LSTM with W2V and GloVe

A recurrent neural network (RNN) is a class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence. This allows it to exhibit temporal dynamic behavior. Unlike feedforward neural networks, RNNs can use their internal state (memory) to process sequences of inputs.

Long short-term Memory (LSTM) is an RNN which can retain the internal states in the form of hidden layers. Widely used in text classification tasks, LSTMs have been known to perform better than a simple classification model.

In the first RNN, I used a word2vec word embeddings pretrained on google news. A word2vec trained on the training data set could have given a better accuracy but using the training data to create word2vec would have either overfitted the model or contained very less words. Whereas, the Google News embeddings contained a much larger vocabulary and was more reliable to not cause overfitting.

The second RNN was the same uni-directional LSTM but now with the GloVe word embeddings. GloVe is a much more principled approach to word embeddings that provides deep insights into word embeddings in general. Unlike word2vec – which learns by streaming sentences – GloVe learns based on a co-occurrence matrix and trains word vectors, so their differences predict co-occurrence ratios [1].

While GloVe has obvious advantages over word2Vec, there was no significance difference in accuracy score between the two word embeddings.

4.4 Model 2 : Bi-directional LSTM with W2V and GloVe

Using bidirectional LSTM will run the inputs in two ways, one from past to future and one from future to past [2]. The LSTM that runs backwards preserve information from the future and using the hidden states combined we can preserve both past and future Hence BiLSTMs show very good results as they can understand context better. The graph (Fig 6) shows that there is a constant increase in accuracy throughout the epochs.

4.5 Parameter Tuning

While there are several hyperparameters and ways to tune the hyperparameters, I have chosen my hyperparameters (Table 3) based on the previous work done in this FNC-1 challenge. Being an NLP beginner and having limitations of my previous experience in text classifications tasks, I decided to go with the parameter settings used in one of the previous papers published in 2017 [4] and the in-class tutorials. A dropout of 0.8 was used to avoid overfitting of the model. The output layer is masked using a 4 layered SoftMax activation. SoftMax function takes as input a C-dimensional vector z and outputs a C-dimensional vector y of real values between 0 and 1. This function is a normalized exponential and is defined as:

$$y_c = \varsigma(\mathbf{z})_c = \frac{e^{z_c}}{\sum_{d=1}^{C} e^{z_d}} \quad \text{for } c = 1 \cdots C$$

SoftMax Function

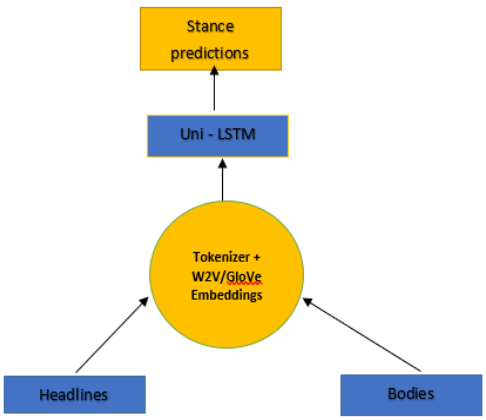

Fig 3: LSTM with W2V and GloVe

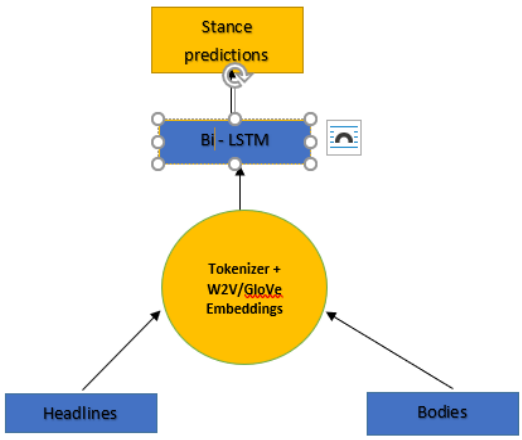

Fig 4: Bi-LSTM with W2V and GloVe

Parameters	Values
Sentence Length	170
Vocabulary Size	400000
LSTM_DIM	128
Embedding_DIM	300
Batch_Size	200
Epochs	10
Activation	Softmax
Dropout	0.8

Table 3 : Hyperparameters

RNN	Validation set	Test Set
W2V LSTM	81	85
GloVe LSTM	94	85
W2V Bi-LSTM	94	86
GloVe Bi -LSTM	92	96

Table 4: Validation vs Test set

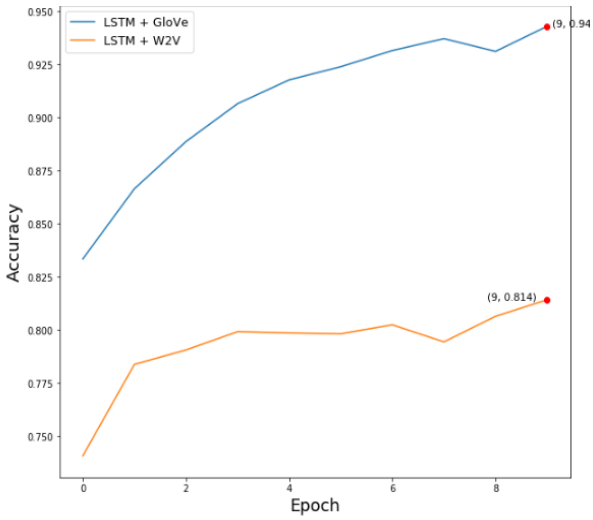

Fig 5: Epoch(10) difference between W2V and Glove with LSTM

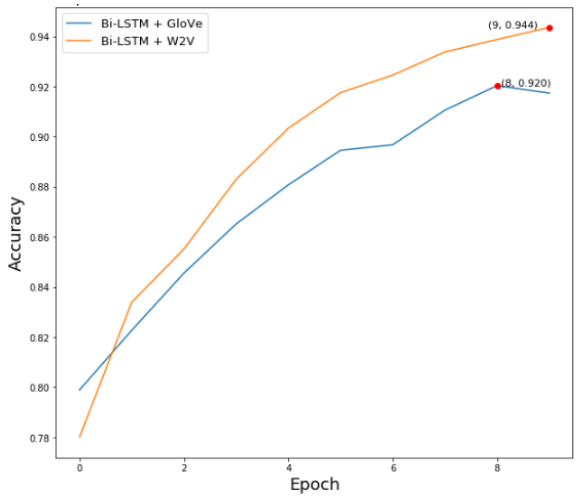

Fig 6: Epoch(10) difference between W2V and Glove with Bi-LSTM

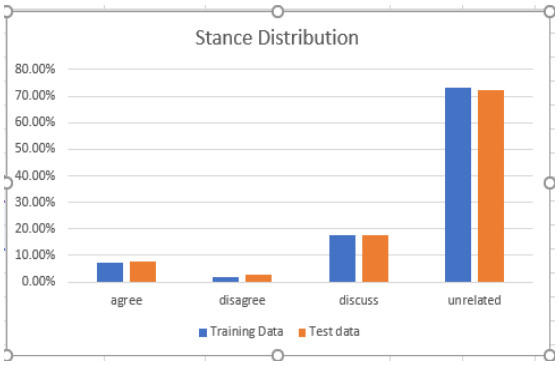

Fig 7: Train vs Test Stance Distribution

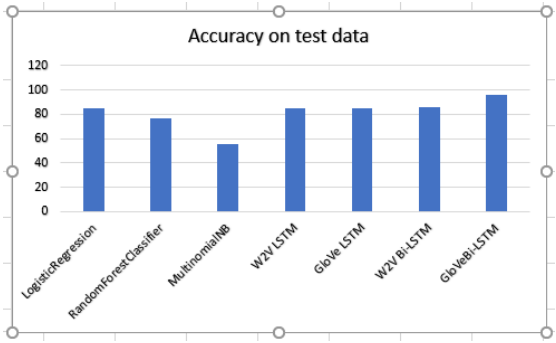

Fig 8: Accuracy score of all the models

5 Qualitative Analysis (Ablation)

5.1 LSTM vs Bi-LSTM

Both the LSTMs used an epoch of 10 and the corresponding validation accuracy was plotted to compare the performance of w2v and Glove.

In uni-LSTM GloVe clearly outperformed word2vec during the Epochs. However, on the competition dataset both showed almost the same accuracy.

In Bi-LSTM, both w2v and GloVe, almost had the same accuracy during Epochs. However, GloVe performed better on the competition dataset.

The performance of LSTMs (Table 4) with respect to word2vec and GloVe both on validation and test sets show us that all the four RNNs were consistent on the competition dataset except the Bi-LSTM with GloVe. Despite the performance of GloVe, the weighted score of Bi-LSTM with word2vec was 4748 when compared to other models that gave a score of around 3900. The reason for a low score can be attributed to the weight given to agree, disagree, discuss and the skewness in data (Fig 7).

5.2 Submission:

The performance of all the three models is shown in Fig 8. The Bi-LSTM with W2V was finally submitted in the Codalab, platform where the competition was conducted, and the weighted score was **4748.25.**

6 Future work

In addition to LSTM and BiLSTM there are two other techniques that can possibly perform better on the unknown data. While using Attention is believed to increase the accuracy, it is very unlikely to incorporate the mechanism in a classification task. Selecting states to attend to is more appropriate for translation tasks Hence the Attention mechanism is only stated as a suggestion and not implemented as a part of this project. Second approach is to use conditional encoding, one LSTM to encode the headlines and another LSTM to encode the bodies Finally, the last output vector of the LSTM of bodies is used to predict the stance of the headline body pairs.

7 References

[1] Jeffrey Pennington, Richard Socher, Christopher D. Manning, 2014. *GloVe: Global Vectors for Word Representation.* Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014) 12.

[2] Isabelle Augenstein, Tim Rocktaschel, Andreas Vlachos, ¨ and Kalina Bontcheva. 2016. *Stance Detection with Bidirectional Conditional Encoding*. In Proceedings of EMLNP.

[3] Benjamin Riedel, Isabelle Augenstein Georgios P. Spithourakis and Sebastian Riedel, 2017. *A simple but tough to beat baseline for the Fake News Challenge stance detection task.* ArXiv 1707 03264

[4] Stephen Pfohl, Oskar Triebe and Ferdinand Legros, 2017.*Stance Detection for the Fake News Challenge with Attention and Conditional Encoding.* Stanford CS224d Deep Learning for NLP final project.

[5] Andreas Hanselowski Avinesh PVS, Benjamin Schiller Felix Caspelherr Debanjan Chaudhuri, Christian M Meyer, and Iryna Gurevych, 2018. *A retrospective analysis of the fake news challenge stance detection task.* In Proceedings of the 27 th International Conference on Computational Linguistics, COLING 18 pages 1859 1874 Santa Fe, NM, USA.