Stance Detection with Attention and Conditional Encoding

MSCI641 Fake News Challenge

Yuqing Zhao
University of Waterloo

Abstract

The Fake News Challenge is a public challenge, which encourages competitors to develop a stance detection tool that can effectively deal with fake news problem and could be incorporated into an AI-assisted fact-checking pipeline. (Pomerlau and Rao, 2016). The MSCI641 Fake News Challenge used the idea of the Fake News Challenge Stage one (FNC-1), and the dataset is provided on the CodaLab challenge website. I implemented four neural network models to deal with this problem: LSTMs (Baseline), Bidirectional LSTMs (Baseline), LSTMs with attention and conditional encoding LSTMs with attention (CEA LSTMs). The main goal of this report is to analyze the performances of these four models. The final results show that CELA outperformed than the other three models, with the accuracy of 96.60% and the competition weighted score of 1819.50. The second good model is LSTM with attention, with the accuracy 95.37% and the competition weighted score of 1817.0. Bidirectional LSTMs achieved 95.35% and the weighted score of 1156. Baseline LSTMs achieved the accuracy 90.61% and the weighted score of 1204.25. Furthermore, through the experiments, I found CEA LSTMs and LSTMs with attention performed well with long truncation length while the performance of baseline LSTM decreased as the truncation length increased. Moreover, I found the GloVe with basic LSTMs model outperformed than Word2Vec with basic LSTMs model.

1 Introduction

"Fake news, defined by the New York Times as 'a made-up story with an intention to deceive' (Tavernise, 2016) 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' had caused a 'great deal of confusion' about the facts of current events" (Barthel et al., 2016). At the same time, Stance Detection is a challenging problem which is to automatically classify the attitude expressed in a text towards a given target, which has been one of the most popular topics in Natural Language Processing field. The goal of the Fake News Challenge is to explore how artificial intelligence could help combat fake news. The first step of the Fake News Challenge is called "Stance Detection," and this is what our course project refers to (Pomerlau and Rao, 2016).

In my project, I used the concept of recurrent neural networks (RNNs) and implemented four neural network models: long short-term memory networks (LSTMs), bidirectional LSTMs, LSTM with attention mechanism and conditional encoding LSTMs with attention mechanism.

2 Related Work

2.1 LSTMs and BRNNs

Long Short-Term Memory architecture is a particular kind of RNN introduced by (Hochreiter and Schmidhuber, 1997), which enables to learn long-term dependencies. (Schuster et al., 1997) introduced bidirectional recurrent neural networks (BRNNs) with a principle that the neurons of a reg-

ular RNN is split into two directions, which enables to utilize input information from the past and future.

2.2 Stance Detection

The recognizing textual entailment is similar to stance detection task, which aims to process two sentences and determine these semantic relationships: either one of these two sentences is a logical consequence of the other one, these two sentences hold contradictory statements, or they are not related. To deal with reasoning about entailment task, (Rocktaschel et al. 2016) used attentive neural networks that attend over past output vectors, and the conditional encoding concept that the hypothesis conditioned on the premise rather than independently encoding these two parts. The authors have demonstrated conditional encoding can improve the model performance as it enables information to flow from the premise processing part to the hypothesis processing part. The authors also showed that conditional encoding LSTMs model with neural attention mechanisms outperformed than other state-of-art methods to deal with recognizing textual entailment. Due to previous works, it is reasonable to believe the effectiveness of conditional encoding scheme and neural attention mechanisms in stance detection task.

2.3 Previous Methods of FNC-1

Stanford students Pfohl et al. (2017) proposed conditional encoding attention LSTMs (CEA LSTMs) has effectively improved the LSTMs with attention mechanism model, with a competition score of 0.808. CEA LSTMs is a modified LSTMs model with attention mechanism and also conditional encoding scheme.

3 Approaches

3.1 Description of Data

Datasets including training dataset, validation dataset, test dataset and BodyID Lookup Table are provided on the CodaLab website. Training dataset and validation dataset contain Headline-Body pairs. Each article-body pair has been labeled with a stance, Unrelated, Agrees, Disagree and Discuss, while the test dataset lack labels and need competitors to predict each row and label it by Deep Learning models. There are two main challenges in FNC-1. First, BodyID Lookup Table contains sentences of various lengths. In the training dataset, the longest sentence length is 4876, and the shortest sentence length is 5. Another challenge is that the majority of "Stance" of training dataset is unrelated with **72.65**%. The training dataset contains **66,677** rows in total, and the details of each stance category are shown in Table 1.

	Unrelated	Agree	Disagree	discuss
Training data set	72.65%	7.41%	1.94%	18.00%

Table 1: Percentage of Each Stance Category in Training Dataset

3.2 Scoring System

Since the distribution of labels is highly unbalanced, prediction results will be evaluated by a weighted accuracy measure. Generally, the scoring system has two levels. In particular, classifying a Headline-Body as "Unrelated" correctly will contribute to 0.25 points. If the Headline-Body test pair is related, the score will be incremented by 0.25 if it labels the pair as any of the three classes: agrees, disagrees, or discusses. The evaluation score will so be 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. The scoring detail is shown in Figure 1 (Fake News Challenge website, 2016).

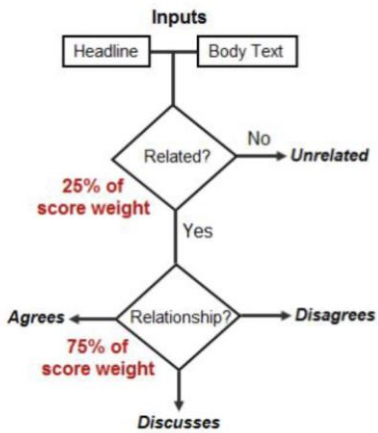

Figure 1: The Scoring System

3.3 Official Baseline Model

The official baseline model is using a GradientBoosting classifier with hand-crafted linguistic features including token overlap, polarity, and refutation. This model achieves a competition

score of 79.53% and competition weighted score of 2029.5. (FNC-1 baseline GitHub)

3.4 My Approaches

After thorough research about previous FNC-1 approaches, I developed four neural network models: Long short-term memory networks (LSTMs) which is a baseline of my project. Another baseline model is bidirectional LSTMs. To further improve the weighted score, I developed two advanced modified LSTMs models which are basic LSTMs in conjunction with attention mechanism and conditional encoding LSTMs with attention mechanism.

The code for these four models was developed under the environment of Python 3.6.6, Keras 2.2.2, and Tensorflow 1.9.0.

3.4.1 Pre-processing

The first step is to prepare text data for deep learning since raw text cannot be fed to deep learning models directly. The text of each headline and article body in the training dataset, validation dataset, and test dataset were processed by the *text_to_word_sequence function*, which is available from the Keras framework. By utilizing *text_to_word_sequence function*, the text is split into words, which is called the *tokenization* step. To further process tokens, I used *Tokenizer* API which is more sophisticated and can be fit and reused. After this step, every word map to a unique integer.

Since sentence length is varying but, the fixed length is needed to feed the model. Thus, before feeding to the embedding layer, it is essential to pad or truncate the sequence. Sentence length is a hyperparameter that may influence model performance and need to analyze it during experiments.

I used pre-trained word embeddings which aims to represent words as vectors. GloVe (Jeffrey al, et.) and Word2Vec (Tomas al et.) are two popular ways to do this. The dimension of GloVe that I used is 50 dimensions trained on a Twitter Corpus, while the dimension of Word2Vec is 300 trained on a Google News dataset. For the embedding layer, masking technique was used to ensure that the zero-padded input would not influence loss calculations.

3.4.2 Long Short-Term Memory Networks (LSTMs)

This is a standard LSTMs model (Hochreiter and Schmidhuber, 1997) which is a baseline of my project. Specifically, it processes a concatenation of tokens of the headline and body and produces the stance classification: “unrelated,” “agree,” “disagree,” “discuss” for each sample. The model structure is shown in figure 2.

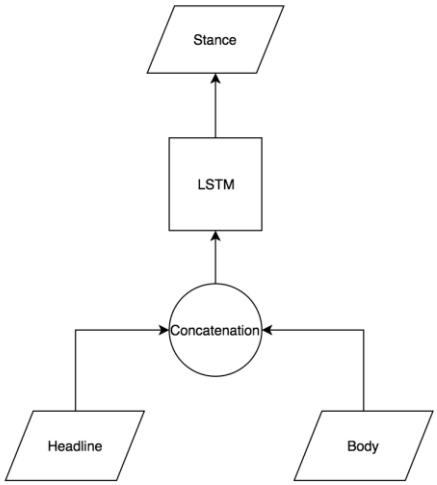

Figure 2: The Structure of LSTM Baseline Model.

3.4.3 Bidirectional LSTMs

Bidirectional LSTMs are an extension of regular LSTMs (Jason, 2017), which can improve the performance of models for sequence classification problem. The basic idea is introduced by (Schuster et al., 1997). Bidirectional LSTMs can be regarded as two LSTMs that the input sequence train on the first LSTM and a reversed copy of the input train on the second one. This is an additional baseline of my project. Specifically, the concatenation of the preprocessed headline and body is fed into bidirectional LSTMs. The structure of the model is shown in figure 3.

3.4.4 Attention Mechanism

Rocktaschel et al. (2016) presented the attention mechanism and implemented in TensorFlow. I mainly used the Keras framework to develop an attention mechanism, by customizing an attention layer class. The principle of attention mechanism is described below.

Let the attention window size is L such that first L states produced by the first LSTM. Let $Y \in \mathbb{R}^{k \times L}$ be a matrix consisting of output vectors $[h_1 \dots h_L]$ such that k is the dimension of the hidden layer. Let h_N is the last output vector (final state) such that N is total sequence length. e $\in \mathbb{R}^L$ be a vector of 1 s. W^y, W^h, W^p, W^x, w are the trainable matrixes that the attention layer should

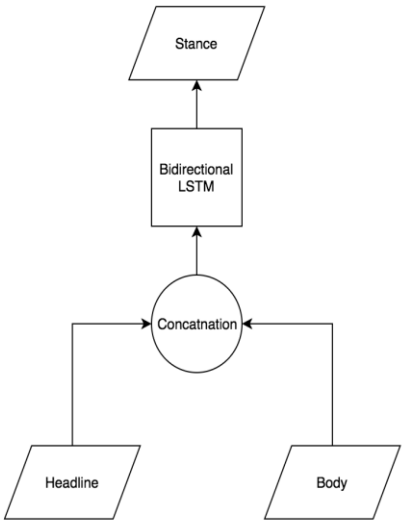

Figure 3: The structure of Bidirectional LSTM baseline model.

learn. h* is the final state that will pass to next layer. It is computed as follows.

$$M = \tanh(W^y Y + W^h h_N e_L^T) \quad (1)$$
$$\alpha = softmax(w^T M) \quad (2)$$
$$r = Y^\alpha \quad (3)$$
$$h^* = \tanh(W^P r + W^x h_N) \quad (4)$$

3.4.5 LSTMs with Attention

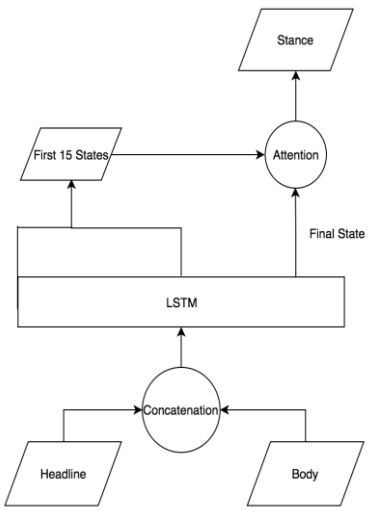

Figure 4: The Structure of LSTMs with attention mechanism model.

LSTMs with attention model is that LSTMs baseline model, which has mentioned in 3.4.2, is augmented with attention mechanism over the first L output vectors. The detailed information about this model is shown in figure 4.

3.4.6 Conditional Encoding LSTMs

A more complex model is CEA LSTMs which was proposed by Pfohl et al. (2017), and I implemented this by Keras framework. In this architecture, the headline and body are processed separately and fed into two LSTM models. The final hidden state of the first LSTM is used to initialize the second LSTM, which is called conditional encoding. Then, the attention mechanism attends the first L output vectors of the first LSTMs and inform the second LSTM which state need to attend to. The structure of this model is shown in figure 5.

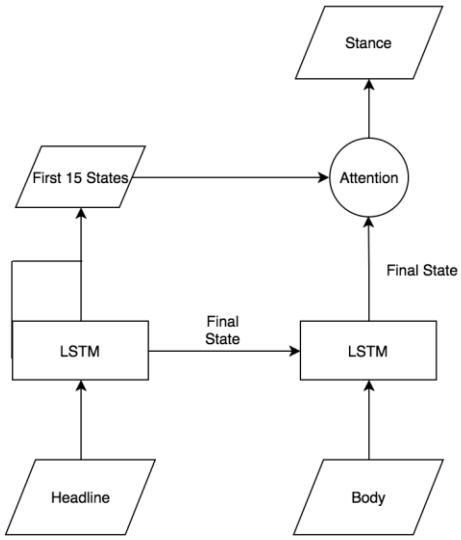

Figure 5: The Structure of CEA LSTM model.

4 Experiments and Results

At the beginning, I tried to use the provided validation dataset to do an evaluation of models. However, I found whatever models and special techniques I used, it was easy to overfit, and the results did not look good with validation accuracy of 72.98% and a weighted score of 1181. After that, I tried to use the provided training dataset only. Before training each model, the provided training dataset was split randomly into a training set (90%, 60009 samples) and a validation set (10%, 6668 samples).

As it is a 4-categories-classification task, experiments were implemented by minimizing cross-entropy loss, specifically *categorical_crossentropy* from Keras. The evaluation was performed on the validation dataset as mentioned previously. I monitored the performance of the validation accuracy and tuned the hyperparameters of each model.

After that, I selected the best parameter set for each model and used that model to predict stances for the unlabeled test dataset whose weighted score was further calculated on the CodaLab.

4.1 Parameter Tuning Experiments

They are many parameters such as dropout probability, truncations length that can be tuned. Due to the time limitation, I only considered some parameter sets. The default parameter setting is not an easy thing for NLP beginners, but I learned from a paper (Pfohl et al., 2017) which is also about FNC-1, and used their default parameter settings. Because the training dataset of (Pfohl et al., 2017) only contains 49972 samples, less than our dataset, so the final hyperparameters for each model are different from their models' after experiments. Since one main factor that impact sequence learning is the sequence length, I put truncation length in the first place when tuning different parameters.

For all models, the number of layer units is 100, the dropout probability of 0.8. Furthermore, Adam optimizer (Kingma and Ba, 2014) was used at all experiments. Initially, embedding vectors were set to be nontrainable since I was afraid the overfitting problem occurs. However, I find the performance was not suitable for all models, and then I set it trainable.

The fixed and tunable hyperparameter sets through experiments are shown in table 2.

Truncation Length	75, 150 (default), 300
Truncation Length of body (CEA LSTMs)	75, 150 (default), 300
Truncation Length of the Headline (CEA LSTMs)	15
Batch Size	128
Epoch	40
Vocabulary Size	40000
Learning Rate	0.001(default), 0.002, 0.003
Dropout	0.8
Attention Length	15
Units	100

Table 2: Hyperparameter Setting.

To further explain, since the 90% body sentence length is 306, the longest truncation length is set to be 300 although the longest body length is 4876. The longer truncation length means, the more time-consuming. The setting of vocabulary size is based on the experiment that the vocabulary is 35375 for all provided datasets.

4.2 Pretrained Word Representations

GloVe and Word2Vec are two popular way to represent words. They are more suitable for the different datasets. (Pfohl et al., 2017) Used GloVe to represent words in the FNC-1 task while (Rocktaschel et al. 2016) used Word2Vec in their approaches trained on a textual entailment dataset. To verify GloVe has better performance on FNC-1 task, I experimented comparing the performance difference between the baseline LSTMs with GloVe and it with Word2Vec. The result shown in figure 6 has demonstrated that LSTMs with GloVe outperformed than it with Word2Vec.

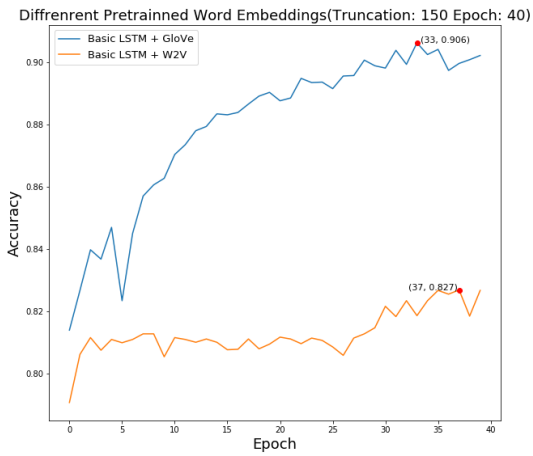

Figure 6: Baseline LSTMs Model with Different Pretrained Word Embeddings (evaluated on validation dataset).

4.3 Model Performance with Different Truncation Length

In order to investigate how different truncation length influence the model performance, I did experiments on 75, 150, 300 truncation words and evaluated the model performance by validation dataset accuracy. The results are shown in figure 7, 8 and 9, corresponding to 75 truncation length, 150 truncation length, and 300 truncation length respectively.

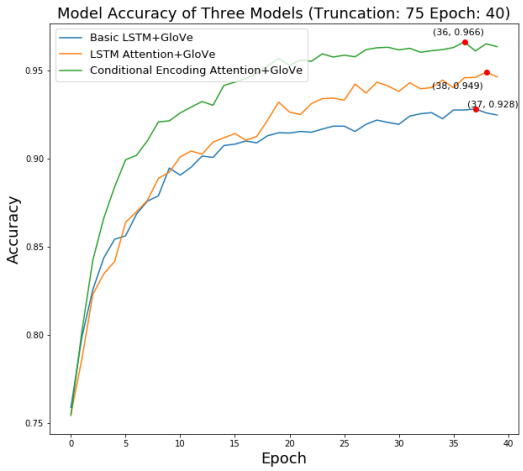

Figure 7: Model Accuracy with 75 Truncation Length (evaluated on the validation dataset).

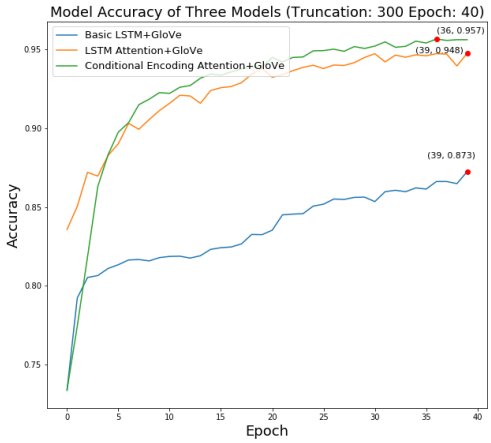

Figure 9: Model Accuracy with 300 Truncation Length (evaluated on the validation dataset).

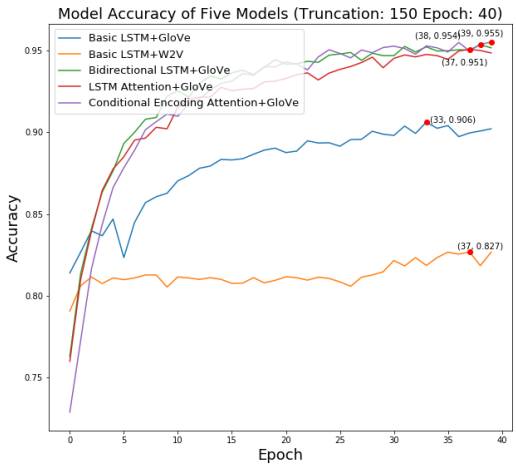

Figure 8: Model Accuracy with 150 Truncation Length (evaluated on the validation dataset).

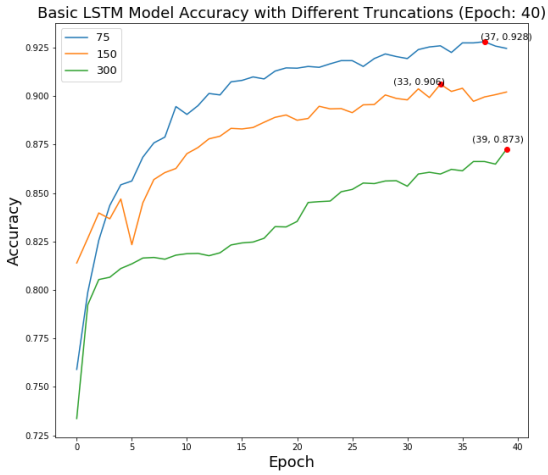

Figure 10: Model Accuracy of LSTM Baseline Model with different Truncation Length (evaluated on the validation dataset).

4.4 Swapping the Headline and the Body for CEA LSTMs Model

The default CEA model is that the first LSTM processes the headline and second LSTM processes the body. The attention mechanism attends to the first L states of first LSTM. To better investigate the model, I swaped the headline and body. Figure 10 shows the result.

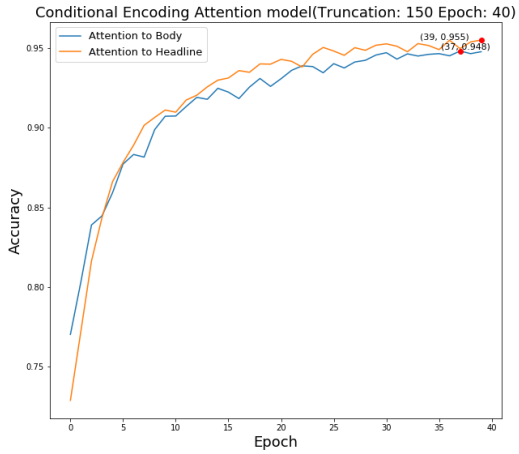

Figure 11: CEA LSTMs Model Accuracy.

It seems that the performance of these two models are quite similar, but the validation accuracy of default CEA model is 0.7% higher than the swapped one. This is probably because the length of a headline is quite short (90% headline length is below 16), and thus first L tokens of headline contain more comprehensive information than the tokens of the body.

4.5 Overfitting Techniques Investigation

Through the experiments, I found it is easy to overfit. Dropout is a traditional way to reduce overfitting. Also, adding activation layers before SoftMax output layer seems a right way. The result shows in figure 12.

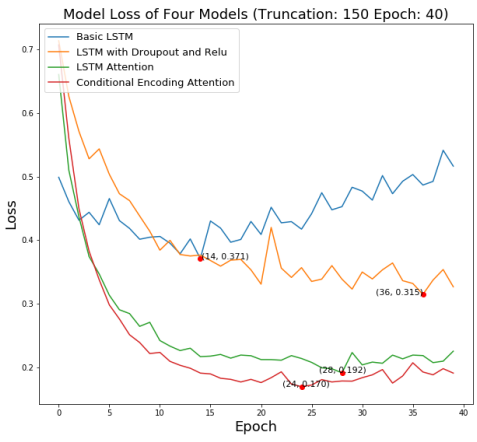

Figure 12: Model Loss.

4.6 Model Performance overview

Models	Accuracy	Weighted Score
LSTMs (baseline)	90.61%	1204.25
Bidirectional LSTMs (baseline)	95.35%	1156.00
LSTMs with Attention	95.37%	1817.00
Conditional Encoding Attention LSTMs	96.60%	1819.50

Table 3: Model Performance.

5 Conclusions and Future Work

5.1 Conclusions

- Conditional encoding LSTMs with attention mechanism (CEA LSTMs) outperformed than other three models whatever truncation length, with a highest validation accuracy of 96.60% and a competition weighted score of 1819.50.
- Attention mechanism and conditional Encoding can effectively utilize more information from long sequences without heavily influencing model performance.
- Dropout can effectively help with the overfitting problem.
- Basic LSTMs model with GloVe performed better than Basic LSTMs model with Word2Vec.
- Bidirectional LSTMs model got a high accuracy but got a low weighted score, since it performed bed in predicting agree, disagree, discuss labels.

5.2 Future Work

Even though I did lots of experiments, a gap between training accuracy and validation accuracy was always there (Shown in the figure). Perhaps, the way that validation data split from the provided training data set is not suitable, or the attention mechanism lacks masking zero technique. This problem is the main task for my future work.

Additionally, I plan to use more complex attention mechanisms such as word-by-word attention (Rocktaschel et al. 2016) and utilize bidirectional

conditional encoding (Augenstein et al., 2016). Furthermore, I plan to build a new model to further utilize context vectors from attention mechanisms as a new kind of feature to feed into a classifier.

Acknowledgments

I want to acknowledge and thank professor Olga Vechtomova and our course teaching assistant Hareesh Bahuleyan. The tutorial taught by Hareesh is good and helpful for my project.

References

Codalab. https://competitions.codalab.org/competitions/19111#phases.

FNC-1 github. https://github.com/FakeNewsChallenge/fnc-1-baseline.

Dean Pomerlau and Delip Rao. Post-facto Fake News Challenge.

Sabrina Tavernise. 2016. As Fake News Spreads Lies, More Readers Shrug at the Truth - The New York Times.

Michael Barthel, Amy Mitchell, and Jesse Holcomb. 2016. Many Americans Believe Fake News IsSowing Confusion— Pew Research Center.

Sepp Hochreiter and Jrgen Schmidhuber. 11 1997. Long Short-term Memory. Neural Comput.9(9):1735–1780.

Schuster, Mike, and Kuldip K. Paliwal. 1997. Bidirectional recurrent neural networks. Signal Processing, IEEE Transactions on 45.11. 2673-2681.2. [1] [1]

Tim Rockt¨aschel, Edward Grefenstette, Karl Moritz Hermann, Tom Ko˘cisk´y, and Phil Blunsom. 9 2015. Reasoning about Entailment with Neural Attention.

Stephen Pfohl, Oskar Triebe, and Ferdinand Legros. 2017. Stance Detection for the Fake News Challenge with Attention and Conditional Encoding

Isabelle Augenstein, Tim Rockt¨aschel, Andreas Vlachos, and Kalina Bontcheva. 2016. Stance Detection with Bidirectional Conditional Encoding. Empirical Methods in Natural Language Processing, (2010):876–885.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. GloVe: Global Vectors for Word Representation.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In NIPS, pages 3111–3119.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. GloVe: Global Vectors forWord Representation.

Jason Brownlee. 2017. How to Develop a Bidirectional LSTM for Sequence Classification in Python with Keras. https://machinelearningmastery.com/develop-bidirectional-lstm-sequence-classification-python-keras/.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.