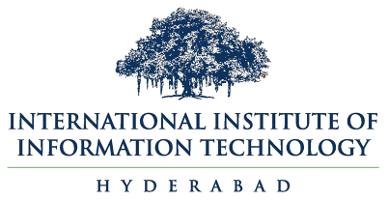
**Project Report**

**On**

**Fake News Detection (25)**

**Statistical Methods in AI (CSE 471)**



Submitted By: Submitted To:

Team Number: 34 Ravi Kiran Sarvadevbhatla

Team Members: Guided By:

Aman Sharma (2018201084) Murtuza Bohra

Ankush Nagpal (2018201083)

Kopal Agarwal (2018202015)

Shreya Upadhyay (2018201091)

**Abstract**

Identifying the veracity of a news article is an interesting problem while automating this process can be a challenging task. Detection of a news article as fake is still an open question as it is contingent on many factors which the current state-of-the-art models fail to incorporate. In this project report, we explore a subtask to fake news identification, and that is stance detection. Given a news article, the task is to determine the relevance of the body and its claim. We present a novel idea that combines the neural, statistical and external features to provide an efficient solution to this problem. We compute the neural embedding from the deep recurrent model, statistical features from the weighted n-gram bag-of-words model and hand crafted external features with the help of feature engineering heuristics. Finally, using deep neural layer all the features are combined, thereby classifying the headline-body news pair as agree, disagree, discuss, or unrelated. We compare our proposed technique with the current state-of-the-art models on the fake news challenge dataset.

**1. Introduction**

Fake news being a potential threat towards journalism and public discourse has created a buzz across the internet. With the advent of social media platforms such as Facebook and Twitter, it has become easier to propagate any information to the masses within minutes. While the propagation of information is proportional to growth of social media, there has been an aggravation in the authenticity of these news articles. These days it has become a lot easier to mislead the masses using a single Facebook or Twitter fake post. For an instance, in the US presidential election of 2016, the fake news has been cited as the foremost contributing factor that affected the outcome [1].

The root cause of this problem lies in the fact that none of the social networking sites use any automatic system that can identify the veracity of news owing across these platforms. A possible reason for this failure is the open domain nature of the problem that adds to the intricacies. The recently organized Fake News Challenge (FNC-1) [13] is an initiative in this direction. The aim of this challenge is to build an automatic system that has the capability to identify

whether a news article is fake or not. More specifically, given a news article the task is to evaluate the relatedness of the news body towards its headline. The relatedness or stance is the relative perspective of a news article towards a relative claim (shown in Table 1).

Headline “Robert Plant Ripped up $800M Led Zeppelin Reunion Contract “ Stance

Body 1 Led Zeppelin's Robert Plant turned down 500 MILLION to reform supergroup. Agree

Body 2 No, Robert Plant did not rip up an $800 million deal to get Led Zeppelin back together. Disagree

Body 3 Robert Plant reportedly tore up an $800 million Led Zeppelin reunion deal. Discuss

Body 4 Richard Branson's Virgin Galactic is set to launch SpaceShipTwo today. Unrelated

**Table 1: Headline-body pairs along with their relative stance.**

The idea behind building a countermeasure for fake news is to use machine learning and natural language processing (NLP) tools that can compute semantic and contextual similarity between the headline and the body, and classify the pairs into one of four categories. Deep learning models have been efficacious in solving many NLP problems that share similarities to fake news which includes but not limited to - computing semantic similarity between sentences [2, 3], community based question answering [4, 5], etc.

The basic building blocks of all deep models are recurrent networks such as recurrent neural networks

(RNN) [6], long short-term memory networks (LSTM) [7] and gated recurrent units (GRU) [8], and convolution networks such as convolution neural networks (CNN) [9]. A deep architecture encodes the given sequence of words into fixed length vector representation which can be used to score the relevance of two textual entities, in our case, relevance of each headline-body pair. Statistical information related to text can be encoded to vectors using the traditional bag-of-words (BOW) approach. The BOW approaches are often combined with term frequency (TF) and inverse document frequency (IDF), and ngrams that helps to encode more information related to the text [9, 10]. These approaches, however simple, have been used to ameliorate the performance of deep models in complex NLP problems such as community question answering [5] and answer sentence selection [11]. Sometimes, it is beneficial to leverage feature engineering heuristics when combined with statistical approaches. The feature engineering heuristics or the external features are used to aid the learning model to successfully converge to a global solution [12, 5, 13]. The external features includes common observations such as number of n-grams, number of words match between headline and the body, cosine similarity between the headline and the body vector, etc. The FNC-1 baseline also includes a combination of feature engineering heuristics that alone achieves a competitive performance, even outperforming several widely used deep learning architectures.

These days it is common to use pre-trained word embeddings such as Word2vec [14] along with deep models for NLP tasks. Similar to word embedding, the recurrent models have been used to encode an entire sentence to a vector. Some of the widely used sentence-to-vector models include doc2vec [15], paragraph2vec [16] and skip-thought vectors [17]. These deep recurrent models helps to capture the semantic and contextual information of the textual pairs, in our case, body and its claim. In our work, we use the skip-thought vector to encode the headline and the body, and combine it with external features and statistical approaches.

**2. Related Work**

In this section, we discuss some previous work that is in relation to fake news identiﬁcation such as rumor detection in news articles and hoax news identiﬁcation. We also discuss the use of deep learning architecture used by some of the researchers with whom our work shares some similarity. Fake news. From an NLP perspective, researchers have studied numerous aspects of credibility of online information. For example, [19] applied the time sensitive supervised approach by relying on the tweet content to address the credibility of a tweet in diﬀerent situations. [18] used LSTM in a similar problem of early rumor detection. In another work, [20] aimed at detecting the stance of tweets and determining the veracity of the given rumor with convolution neural networks. A submission [21] to the SemEval 2016 Twitter Stance Detection task focuses on creating a bag-of-words auto encoder, and training it over the tokenized tweets. FNC-1 submissions. In their work, [22] achieved a preliminary score of 0.8080, slightly above the competition baseline of 0.7950. They experimented on four basic models on which the ﬁnal result was evaluated: Bag Of Words (BOW), basic LSTM, LSTM with attention and conditional encoding LSTM with attention (CEA LSTM). In our work, instead of using the models separately, we combine the best of these models. Another team, [13], combined multiple models in an ensemble providing 50/50 weighted average between deep convolution neural network and a gradient boosted decision trees. Though this work seems to be similar to our work, the diﬀerence lies in the construction of ensemble of classiﬁers. In a similar attempt, a team [23] concatenated various features vectors and passed it through an MLP model. The work by [9], focuses on generating lexical and similarity features using (TF-IDF) representations of bag-of-words (BOW) which are then fed through a multi-layer perceptron (MLP) with one hidden layer. In their work, [24] divided the problem into two groups: unrelated and related. They were able to achieve 90% accuracy on the related/unrelated task by ﬁnding maximum and average Jaccard similarity score across all sentences in the article and choosing appropriate threshold values. A similar work of splitting the problem into two sub problems (related and unrelated) is also performed by [25]. The work by [26] focuses on the use of recurrent models for fake news stance detection.

**3. Implementation details**

We have used two models for fake news detection:

**3.1 Tree Model (Gradient Boosted Classifier)**

This model takes as input a few text-based features derived from the headline and body of an article. Then it feeds the features into Gradient Boosted Trees to predict the relation between the headline and the body (agree/disagree/discuss/unrelated).

**3.2 Features**

*3.2.a Preprocessing*

The labels (agree, disagree, discuss, unrelated) are encoded into numeric target values as (0, 1, 2, 3). The text of headline and body are then tokenized and stemmed. Finally Uni-grams, bi-grams and tri-grams are created out of the list of tokens. These grams and the original text are used by the following feature extractor modules.

*3.2.b Baseline Features*

The hand-crafted features include word/ngram overlap features i.e. cooccurrence (COOC) of word and character n-grams in the headline and the document as well as two lexicon-based features, which count the number of refuting (REFU) and polarity (POLA) words based on small word lists. For example, words like crime, accident, and scandal are often used with negative connotation. If such words are present in both the news headline, or are present in one while absent from the other, then, it is easier to identify such a pair as agree or disagree.

*3.2.c. TF-IDF Features:*

We use 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:

(i) The TF vector of the headline

(ii) The TF vector of the body

We tokenize the headline and body texts as well as derive the relevant vectors using scikit-learn. Different vocabularies are used for calculating the TF and TF-IDF vectors. For the TF vectors, we extract a vocabulary of the 4,000 most frequent words in the training set and exclude stop words (the scikit-learn stop words for the English language with negation terms removed). For the TF-IDF vectors, a vocabulary of the 4,000 most frequent words is defined on both the training and test sets and the same set of stop words is excluded.

The TF vectors are concatenated in a feature vector of total size 8,000 and fed into the classifier.

*3.2.d. Word2Vec Features*

This module utilizes the pre-trained word vectors from public sources, add them up to build vector representations of the headline and body. The word vectors were trained on a Google News corpus with 100 billion words and a vocabulary size of 3 million. The resulting word vectors can be used to find synonyms, predict the next word given the previous words, or to manipulate semantics. For the current problem constructing the vector representation out of word vectors could potentially overcome the ambiguities introduced by the fact that headline and body may use synonyms instead of exact words.

*3.2.e. Sentiment Features*

This modules uses the Sentiment Analyzer in the NLTK package to assign a sentiment polarity score to the headline and body separately. For example, negative score means the text shows a negative opinion of something. This score can be informative of whether the body is being positive about a subject while the headline is being negative. But it does not indicate whether it's the same subject that appears in the body and headline; however, this piece of information should be preserved in other features.

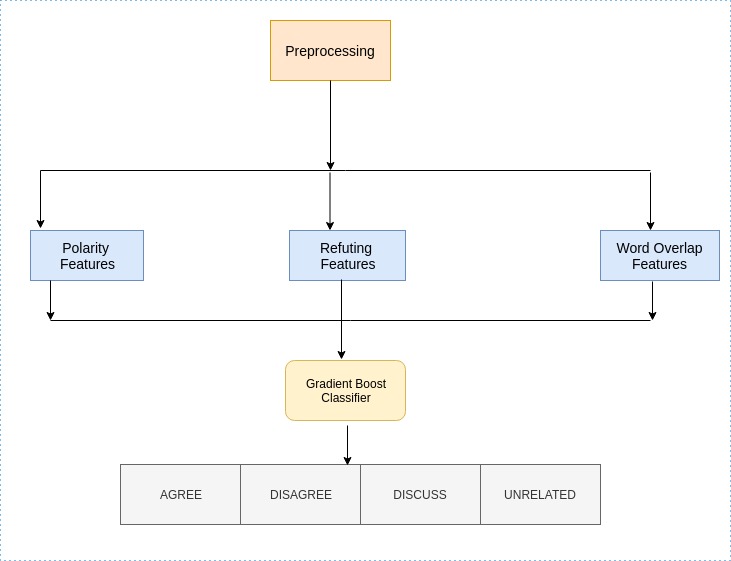
All above features are dumped into pickle files and finally they are fed to Gradient Boosted Classifier and performance of the model is evaluated.

**3.3. Deep Learning Model**

Above 4 features: TF-IDF, Sentiment, Word2Vec, and Baseline features are trained through separate Neural Network each. Then the outputs of each neural network are combined and fed to Softmax activation function to get final output class.

**4. Model Description**

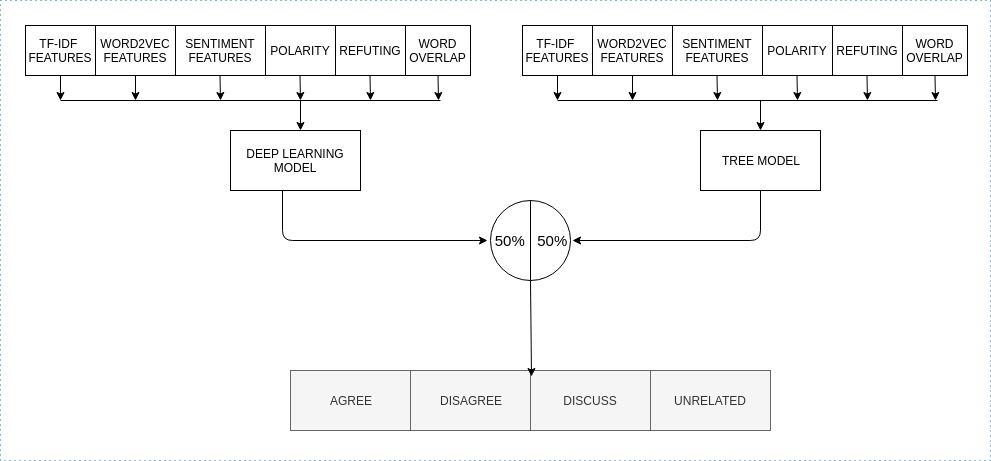
**4.1 Baseline Architecture**

****

**Figure 1:** Baseline Model Architecture

**4.2 Proposed Architecture**

The architecture proposed by us constitutes of a tree-based and a deep learning model. Both the models have been trained on the same set of features and a 50-50 weight is provided to both the models to ensure equal representation.

****

**Figure 2:** Detailed Architecture of the Proposed Model showing the component models

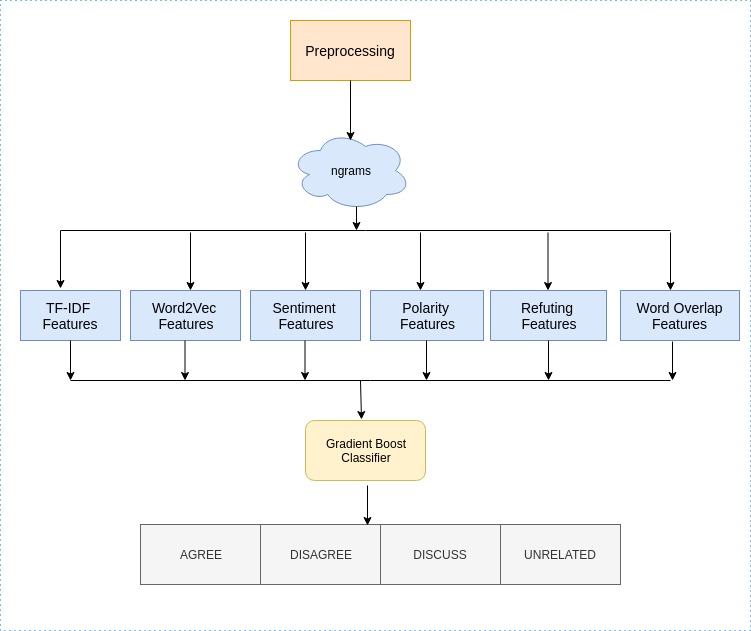
**4.2.a Tree-based-Model Architecture**

In addition to the provided baseline features, we are using some additional statistical features (TF-IDF, Sentiment Features) and neural embeddings (Word2Vec). The classifier being employed here is Gradient Tree Boosting and the following hyperparameters are being used with the classifier:

N\_Estimator = 200

Random\_state = 14128

Gradent Boosting builds an additive model in a forward stage-wise fashion; it allows for the optimization of arbitrary differentiable loss functions. In each stage n\_classes\_ regression trees are fit on the negative gradient of the binomial or multinomial deviance loss function.

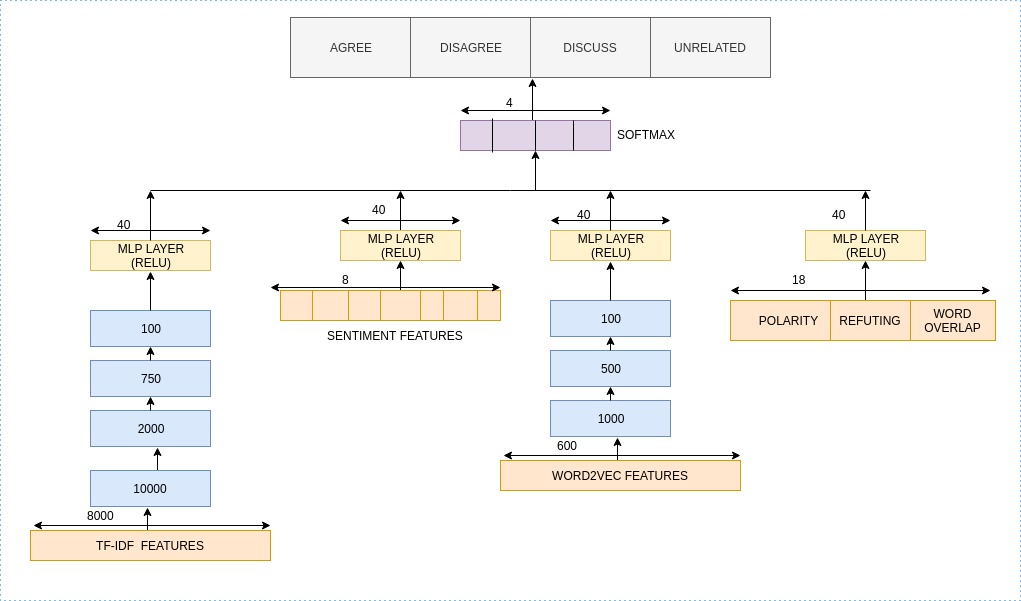
****

**Figure 3:** Architecture of the Tree Based Model

**4.2.b Deep Learning model Architecture**

We used 4 different models for the 4 particular type of features which are TF-IDF, Sentiment, Word2Vec and Baseline features respectively. Since TF-IDF and Word2vec neural embeddings have dimensions of 8000 and 600 respectively therefore, the number of hidden layers required is more as compared to sentiment and baseline features.

All of these sub models, collectively, form a single model that performs better when compared to independent runs of all these isolated models.

****

**Figure 4:** Architecture of the Deep learning Model

**5. Experimentations**

**5.1 Dataset Description**

We use the dataset provided in the FNC-1 challenge which is derived from the Emergent Dataset [15], provided by the fake news challenge administrators. The former consist of 49972 tuple with each tuple consisting of a headline-body pair followed by a corresponding class label stance of either agree, disagree, unrelated or discuss. Word counts roughly ranges between 8 to 40 for headlines and 600 to 7000 for article body. The distribution of FNC-1 dataset is shown in Table 2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| News Articles | Unrelated | Discuss | Agree | Disagree |
| 49972 | 73.17% | 17.83% | 7.36% | 1.68% |

**Table 2.** FNC-1 dataset description.

The ﬁnal results are evaluated over a test dataset provided by fake news organization consisting of 25413 samples.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Hyperparameters | TF-IDF vectors | Sentiment Features | Word2Vec Features | Baseline Features |
| MLP Layers | 6 | 2 | 5 | 2 |
| MLP Neurons | [8000, 10000, 2000, 750, 100, 40] | [8, 40] | [600, 1000, 50, 100, 40] | [18, 40] |
| Dropout | [0.5, 0.5, 0.25, -, -, -] | [-,-] | [0.25, 0.125, 0.1, 0.1, -] | [-,-] |
| Activation | Re-Lu | Re-Lu | Re-Lu | Re-Lu |

|  |  |
| --- | --- |
| MLP Layers | 1 |
| MLP Neurons | 4 |
| Activation | Softmax |
| Optimizer | Adam |
| Learning Rate | 0.001 |
| Batch Size | 32 |
| Loss | Categorical Cross-Entropy |

**Table 3.** Values of hyper-parameters. The ﬁrst half of the table shows the parameters used in architectures for extracting individual features. The second half shows the parameter setting of the feature combination layer that is shown in Figure 2.

**5.2 Training Parameters**

As shown in Figure 2, the proposed model computes the feature vectors separately and then combine these with the help of a MLP layer. We use categorical cross-entropy as the loss function to optimize our architecture with a softmax layer at the output which classify the given headline-body pair into agree, disagree, discuss, and unrelated. The hyper-parameter setting is shown in Table 3.

**5.4 Evaluation metrics**

From Table 2 it is evident that the FNC-1 dataset shows a heavy bias towards unrelated headline-body pairs. Recognizing this data bias and the simpler nature of the related/unrelated classiﬁcation problems, the organizers of FNC-1 introduced the following weighted accuracy score as their ﬁnal evaluation metric.

*Score1 =Accuracy Related, Unrelated*

*Score2 = Accuracy Agree, Disagree, Discuss*

*ScoreFNC = 0.25∗Score1 + 0.75∗Score2*

We use the ScoreFNC as the main evaluation criteria while comparing the proposed model with other related techniques. We also use the class-wise accuracy for further evaluation of the performance of all the techniques.

**6. Results**

**Tree based Model**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Agree | Disagree | Discuss | Unrelated | Overall |
| Agree | **1057** | **11** | **639** | **196** | **1903** |
| Disagree | **261** | **20** | **237** | **179** | **697** |
| Discuss | **1019** | **15** | **2969** | **504** | **4464** |
| Unrelated | **84** | **3** | **292** | **17979** | **18349** |
| Overall | **2421** | **49** | **4094** | **18849** | **25413** |

**Table 4:** Confusion Matrix of Tree Based Model run standalone

Accuracy: 86.46% (Test), 93.99% (Validation)

FNC Score: 77.59 (Test), 90.28 (validation)

**Deep Learning Model**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Agree | Disagree | Discuss | Unrelated | Overall |
| Agree | **799** | **6** | **816** | **282** | **1903** |
| Disagree | **130** | **2** | **317** | **248** | **697** |
| Discuss | **615** | **2** | **3241** | **606** | **4464** |
| Unrelated | **60** | **0** | **328** | **17961** | **18349** |
| Overall | **1604** | **10** | **4702** | **19097** | **25413** |

**Table 5:** Confusion Matrix of the Deep Learning Model when run Standalone

Accuracy: 86.58% (Test), 90.59% (Validation)

FNC Score: 77.277 (Test), 83.97 (validation)

**Baseline Model**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Agree | Disagree | Discuss | Unrelated | Overall |
| Agree | **86** | **3** | **1535** | **279** | **1903** |
| Disagree | **18** | **4** | **430** | **245** | **697** |
| Discuss | **136** | **17** | **3708** | **603** | **4464** |
| Unrelated | **9** | **1** | **567** | **17772** | **18349** |
| Overall | **249** | **25** | **6240** | **18899** | **25413** |

**Table 6:** Confusion Matrix of the Baseline

Accuracy: 80.26% (Test), 83.19% (Validation)

FNC Score: 75.320 (Test), 78.799 (validation)

**Proposed Model:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Agree | Disagree | Discuss | Unrelated | Overall |
| Agree | **951** | **3** | **713** | **236** | **1903** |
| Disagree | **190** | **9** | **276** | **222** | **697** |
| Discuss | **731** | **4** | **3199** | **530** | **4464** |
| Unrelated | **52** | **2** | **296** | **17999** | **18349** |
| Overall | **1924** | **18** | **4484** | **18987** | **25413** |

**Table 7:** Confusion Matrix of the Proposed Model (Combined)

Accuracy: 87.19% (test)

FNC Score: 78.42 (Test)

Precision: 0.86

Recall: 0.87

F1 Score: 0.85

**7. Work Division**

Detailed work distribution among the team members:

1. Ankush: TF-IDF features, n-grams, Gradient Boost Classifier

2. Kopal: Polarity and Refuting Features, Combined Model, Multi layer Perceptron Model

3. Aman: Preprocessing, Word2Vec Features, Multi layer Perceptron Model

4. Shreya: Preprocessing, Sentiment Features, Word Overlap Features

**References**

[1] NYTimes. As fake news spreads lies, more readers shrug at the truth. 2016.

[2] E. Agirre, D. Cer, M. Diab, A. Gonzalez-Agirre, and W. Guo. sem 2013 shared task: Semantic textual similarity, including a pilot on typedsimilarity. In In\* SEM 2013: The Second Joint Conference on Lexical and Computational Semantics. Association for Computational Linguistics. Citeseer, 2013.

[3] R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, and S. Fidler. Skip-thought vectors. In Advances in neural information processing systems, pages 3294–3302, 2015.

[4] L. Yang, Q. Ai, D. Spina, R.-C. Chen, L. Pang, W. B. Croft, J. Guo, and F. Scholer. Beyond factoid qa: Eﬀective methods for non-factoid answer sentence retrieval. In European Conference on Information Retrieval, pages 115–128. Springer, 2016.

[5] Y. Yang, W.-t. Yih, and C. Meek. Wikiqa: A challenge dataset for opendomain question answering. In EMNLP, pages 2013–2018, 2015.

[6] P. Neculoiu, M. Versteegh, M. Rotaru, and T. B. Amsterdam. Learning text similarity with siamese recurrent networks. ACL 2016, page 148, 2016.

[7] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555, 2014.

[8] H. He, K. Gimpel, and J. J. Lin. Multi-perspective sentence similarity modeling with convolutional neural networks. In EMNLP, pages 1576–1586, 2015.

[9] B. Riedel, I. Augenstein, G. P. Spithourakis, and S. Riedel. A simple but tough-to-beat baseline for the fake news challenge stance detection task. arXiv preprint arXiv:1707.03264, 2017.

[10] R. Davis and C. Proctor. Fake news, real consequences: Recruiting neural networks for the ﬁght against fake news. 2017.

[11] L. Yu, K. M. Hermann, P. Blunsom, and S. Pulman. Deep learning for answer sentence selection. arXiv preprint arXiv:1412.1632, 2014.

[12] L. Yang, Q. Ai, D. Spina, R.-C. Chen, L. Pang, W. B. Croft, J. Guo, and F. Scholer. Beyond factoid qa: Eﬀective methods for non-factoid answer sentence retrieval. In European Conference on Information Retrieval, pages 115–128. Springer, 2016.

[13] S. B. Yuxi Pan, Doug Sibley. Talos. http://blog.talosintelligence. com/2017/06/, 2017.

[14] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Eﬃcient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781, 2013.

[15] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems, pages 3111–3119, 2013.

[16] R. ˇReh˚uˇrek and P. Sojka. Software Framework for Topic Modelling with Large Corpora. In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, pages 45–50, Valletta, Malta, May 2010. ELRA. http://is.muni.cz/publication/884893/en.

[17] R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, and S. Fidler. Skip-thought vectors. In Advances in neural information processing systems, pages 3294–3302, 2015.

[18] T. Chen, L. Wu, X. Li, J. Zhang, H. Yin, and Y. Wang. Call attention to rumors: Deep attention based recurrent neural networks for early rumor detection. arXiv preprint arXiv:1704.05973, 2017.

[19] C. Castillo, M. Mendoza, and B. Poblete. Predicting information credibility in time-sensitive social media. Internet Research, 23(5):560–588, 2013.

[20] Y.-C. Chen, Z.-Y. Liu, and H.-Y. Kao. Ikm at semeval-2017 task 8: Convolutional neural networks for stance detection and rumor veriﬁcation. Proceedings of SemEval. ACL, 2017.

[21] I. Augenstein, A. Vlachos, and K. Bontcheva. Usfd at semeval-2016 task 6: Any-target stance detection on twitter with autoencoders. In SemEval@ NAACL-HLT, pages 389–393, 2016.

[22] S. Pfohl, O. Triebe, and F. Legros. Stance detection for the fake news challenge with attention and conditional encoding.

[23] B. S. Andreas Hanselowski, Avinesh PVS and F. Caspelherr. Team athene on the fake news challenge. 2017.

[24] A. K. Chaudhry, D. Baker, and P. Thun-Hohenstein. Stance detection for the fake news challenge: Identifying textual relationships with deep neural nets. 2017.

[25] S. Chopra, S. Jain, and J. M. Sholar. Towards automatic identiﬁcation of fake news: Headline-article stance detection with lstm attention models, 2017.

[26] K. Miller and A. Oswalt. Fake news headline classiﬁcation using neural networks with attention. 2017.