"News Title can be Deceptive" Title Body Consistency Detection for News Articles using Text Entailment

Tanik Saikh, kingshuk Basak, Asif Ekbal, and Pushpak Bhattacharyya
{1821cs08, kingshuk.mtcs16, asif, pb}@iitp.ac.in

Indian Institute of Technology Patna, India

Abstract. News Title (NT) and News Body (NB) consistency detection is a demanding problem in Fake News Detection. In this paper, we formulate consistency detection between NT and NB from the perspective of Textual Entailment (TE), and propose various deep learning based methods for solving this problem. Inconsistency between NT and NB can affect the purpose of the news and alter the view of the reader towards the news contents. We develop various models based on *Multilayer Perceptron (MLP)*, *Convolutional Neural Networks (CNN)*, *Long Short-Term Memory (LSTM)*, and a combination of *CNN* and *LSTM*. Evaluation of the proposed approaches on a recently released benchmark dataset demonstrate the effectiveness of our approaches.

Keywords: News Title Body Consistency Detection · Textual Entailment · Fake News Detection · Deep Learning.

1 Introduction

In recent times, there has been a phenomenal growth in web information due to the presence of numerous websites, blogs, and social media sites. The number of news sources is growing daily, and often multiple reportings are available for a particular event (from the different sources). Many of these reporting are legitimate, whereas something could also be misleading (or, fake). Fake news is defined as a *"made-up stories with an intention to deceive"* [41], i.e. the task of fake news detection system can be defined as to compute the probability of a news article being fake [8]. A robust automatic fake news detection system may consist of several modules, *viz. i. finding out whether the textual content of a news article is true or not*, *ii. determining the relationship between headline/title and the body text* and *iii. evaluating the intrinsic prejudice of a written text*. Each of these modules has its own challenges. Fake news detection systems should have the ability to reason about arguments with common sense knowledge, which relates to a system which considers recognizing semantic phenomena as what typically TE does. In this paper, we are taking the second challenge into account.

In a news article, the headline (i.e. title) is the most important part. Titles give an overview of the overall contents of the news body. Headlines are catchy

and sometimes could also be deceptive. Misleading news titles can lead one away from the correct path or direction. More often, people just read the title of the news and make the judgment. The title can affect peoples' way of reading an article. Hence detection of a misleading or inconsistent title is very important these days. Title-Body consistency detection problem can be seen as if reading the content (body) of the news one can get an idea that the information provided in the body of the news is same as the title of the news. In a simple way, we can sum it up as whether the news body infers the news title or not. It corresponds to the relation between the NB and NT. We can correlate this *Title-Body* inference task with a very popular task in Natural Language Processing (NLP), namely *Natural Language Inference (NLI)/ TE* [9, 24]. The task is conceptually similar to the task of TE. The concept of TE was first introduced in the shared task for Recognizing Textual Entailment-1 (RTE-1) in the year of 2005 and defined as an unidirectional relation between the two texts called as *Text(T)* and *Hypothesis(H)*. It is defined as: ***T** entails **H** if, typically, a human reading of **T** would infer that **H** is most likely to be true* [9], i.e. to judge whether H is the logical consequence of T or not. TE can be viewed as a generic task which captures major semantic inference needed across many NLP applications, namely *Text Summarization* [19], *Question Answering* [12], *Information Retrieval* [25], *Information Extraction* [13], *Machine Translation evaluation* [10], *Novelty detection* [22] and many more.

In this paper, we frame the problem of Title-Body consistency detection with respect to TE. Title-Body consistency detection is highly relevant to TE in the following sense: considering the body of news as Text(**T**) and headline/title of news as Hypothesis (**H**) of TE. If a hypothesis (title) can be inferred from a text (news body) then it can be considered as textually entailed (also as a consistent title). On the other hand, if a hypothesis (title) cannot be inferred from a text (news body) then it can be considered as a not-entailed (also known as an inconsistent title). We consider NB as T and NT as H [1]. With this intuition, we formulate the title-body consistency problem from the viewpoint of TE and setup our experiments subsequently. Overall the problem is: for a given news title/headline (NT) and an article body (NB), the task is to determine the stance of the headline with respect to the article. The problem can also be treated as stance classification between the news bodies and headlines. The labels are *agree, disagree, discuss and unrelated*. This dataset is released as a part of *Fake News Challenge stage 1 (FNC-I): Stance Detection*[2][30]. Motivations and contributions of our current work stem from the following:

- The problem of stance classification between NB and NT has not been attempted much, especially from the viewpoint of TE.
- We investigate the appropriateness of deep learning models for determining the title-body consistency within the framework of TE. To the best of our knowledge this has not been attempted so far in the literature.

[1] We use these terms interchangeably throughout the paper
[2] http://www.fakenewschallenge.org/

We propose various deep learning based models for *Title-Body* consistency detection. The models are based on MLP, CNN [16], LSTM [14], and the combination of both.

2 Related Work

Dealing with fake news and fact-checking of an article is a very challenging and interesting problem to human being, in particular, to news and social media industries. Literature reveals that there are an ample number of works carried out towards this direction and also there are rooms to explore. The authors in [4] proposed a novel corpus that represents an unified view of stance detection, stance rationale, relevant document retrieval, and fact checking. The task defined in [27] solved the problem of stance detection on Fake News dataset by applying an end-to-end memory network as proposed in [39] which includes convolutional neural network, recurrent neural network and the similarity matrix. [37] proposed a model for stance detection named as *360 stance detector*. Given a news search query and a topic, the task of the tool is to retrieve the news articles related to the query and analyze their stance. The task defined in [33] performed meticulous linguistic analysis in the context of political fact-checking and fake news detection. They compared the language of real news with the language of satire, hoaxes, and propaganda with the aim to find the linguistic characteristic of suspicious texts. Previous computational works [42, 7] have posed fact-checking system by exploiting the entailment concept from the knowledge bases. [5] conducted an investigation into the unique linguistic styles found in clickbait articles. [20] examined impact, characteristics and detection of hoax documents on Wikipedia. [36] differentiated between various fake news types and defined there are three fake news tasks based on their types *viz: serious-fabrications, large-scale hoaxes, humorous fakes*. The work reported in [3] built a classification model, which is able to predict whether a piece of text is fake based on its content. They proposed different models, namely Logistic Regression, Two-layer Feed Forward Neural Network, Recurrent Neural Network, Long short-term memory, Gated Recurrent Units, Bi-directional RNN with LSTM, CNN with max Pooling, Attention Augmented CNN. The GRU model yields the best F1 score of 84% among these models. They reported the evaluation results on the Kaggle Dataset [3].

The task narrated in [29] applied the concept of neural attention and conditional encoding to LSTM and obtained an accuracy of 80.8%, outperforming with 1.3% margin over the reported baseline of 79.5%. Altogether, they used four models, namely, Bag-of-Words(BOW), basic LSTM, LSTM with attention, Conditional encoding LSTM with attention. The system reported in [32] explored the various neural network architectures for stance detection in news article. They made use of transfer learning on *Stanford Natural Language Inference(SNLI)* [6] corpus, which consists of T and H pairs. They trained a conditional encoding model (as utilized by [35] for RTE on SNLI dataset) and evaluated on the fake

[3] https://www.kaggle.com/mrisdal/fake-news/data

news dataset. Apart from the above-cited works, there are some other works that can be found in the literature [31, 23, 47]. Recently, [44] examined PolitiFact data and also made use of various meta-data features for the prediction. [11] posited a novel dataset namely, *Emergent*, which was driven from the digital Journalism project, namely *Emergent* [38]. The Stance Detection dataset for FNC-1 is also derived from this *Emergent* dataset.

3 Methodology

Our first proposed model is based on MLP. Later we develop models based on the followings: CNN, LSTM and a combined model of both. In the following subsections, we will discuss these models.

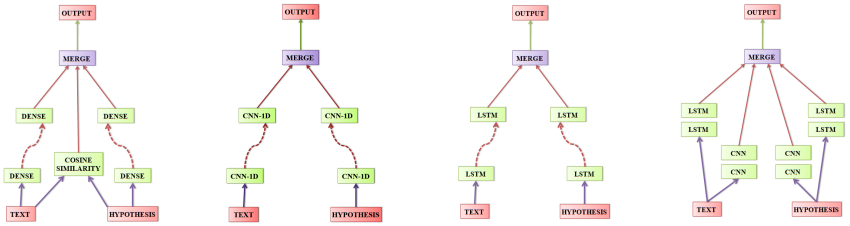

Fig. 1. Proposed models architecture

3.1 Multilayer Perceptron Model (MLP)

The architecture of this model is shown in Figure (1: MLP). In this model, 300-dimension vector representation of NT and NB are taken as input and fed to a two separate dense layers [4]. In this model, we employ 7 such dense layers one after another. Adding of such 7 dense layers is fully empirical. We keep on adding dense layers, starting from one layer, perform experiments, note the results, and observe that the results are interesting. But after 7 dense layers the result remain same, then we freeze the model with 7 dense layers[5]. Say for example, we obtain u and v as the final outputs from 7-dense layers for NT and NB, respectively. We concatenate (u;v), u and v. We also compute cosine similarity [6] between the vector representations of NT and NB to capture the semantic information of the pair of texts. Further, we concatenate the value of cosine similarity with *(u;v)* to get the final set of output neurons. The justification of concatenation of cosine similarity in this way is also empirical and also to capture semantic similarity between NB and NT. Finally, the output neurons are given to the output layer.

[4] dense layer indicates feed-forward neural network, we use these terms interchangeably through out the paper

[5] For space constraint, we are not able to show all the seven layers in the diagram, the dotted lines indicate other layers.

[6] https://en.wikipedia.org/wiki/Cosine_similarity

3.2 Convolutional Neural Networks Model(CNN)

Literature shows that CNNs [17] are the best performer for text classification. An ample number of works have been published for text similarity including NLI [15, 45]. In the traditional feed-forward neural network, each output interacts with each input, but CNN's structure imposes local interaction between the inputs within a filter size m. CNNs perform well in feature extraction. We utilize that quality to extract features automatically from NT and NB to capture better relationships between them. Overall, CNNs are generally hierarchical, and as the task is a classification problem, we are tempted to use this model. CNNs consist of one or more convolution and pooling layers followed by one or more dense layers. The vector representations of both the NT and NB are fed into CNNs which are 1-dimensional vector. So, we make use of CNN-1D (Convolutional Neural Network with 1 dimension). The architecture of this model is shown in Figure(1: CNN).[7] Working of Convolutional Layer relies on the following formulas:

$$a_i^l = b_i^l + \sum_{i=1}^{m^{l-1}} conv1D(w_{i,j}^l, s_i^{l-1}) \tag{1}$$

$$s_i^l = f_l(a_i^l) \tag{2}$$

where, a_i^l is the input of the i^{th} feature signal of layer l, b_i^l is the bias of the feature signal of layer l, s_i^l is the output from the layer l of feature i, $w_{i,j}^l$ is the weight vector of kernel or filter between the j^{th} feature in $(l-1)^{th}$ layer and i^{th} feature in the l^{th} layer, $f_l(.)$ is the activation function for l^{th} layer.
The vectors of T and H are given to a series of four CNN-1D with a filter size of 2. On top of convolutional layer, we build a Max-k pooling layer, where the value of k is 2. Intuitively, we want to capture the top-k values from each convolutional filter. By Max-k pooling, we are keeping maximum k values for each filter, which indicates the highest degree that a filter matches the input sequence. On top of Max-pooling, there is a flatten layer. We merge the Flatten layer's output for T(NB) and H(NT), which are further fed into the output layer for classification.

Table 1. Results (accuracy) with different number of CNN model

Model with # of CNN	Accuracy
One	85.35
Two	91.42
Three	89.46
Four	**92.50**

We keep on adding CNN and perform experiment at each step. We note the result on adding each CNN. The final model is a combination of four CNNs as

[7] Due to space limitations, all the CNNs applied are not shown, the dotted lines indicate the same.

the model was found to be over-fitted after this. Table 1 shows the evaluation results. It shows that even after increasing the number of CNNs, we do not observe any improvement.

3.3 Long Short-Term Memory Model (LSTM)

LSTMs [14] are a special kinds of recurrent neural network (RNN), which can learn long-term dependencies by managing vanishing or exploding gradient problem very smartly. In general, RNNs are sequential. Recently, it has been successfully fostered for solving various NLP tasks, including Machine Translation [40], Language Modeling [46] and also for TE/NLI [6, 43, 35]. As it has been successfully applied to TE/NLI, we also leverage this network to tackle our problem. However, we make use of multiple LSTMs to utilize the goodness of multiple RNNs.
The vector representation is given to the LSTM, with each component of the input vector is assigned to each time-step of the LSTM. The output of the first LSTM goes to the input of the second LSTM and so on. The production at each time-step from the first LSTM is the input to each time-step of the second LSTM. We stack 3 LSTMs for both the NT and NB. Outputs of the LSTMs for both the NT and NB are merged and fed into output layer.

The architecture of the proposed model is shown in Figure (1: LSTM)[8]. It models the word sequence as follows:

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f) \tag{3}$$
$$i_t = \sigma(W_t.[h_{t-1}, x_t] + b_i) \tag{4}$$
$$\widetilde{C_t} = tanh(W_c.[h_{t-1}, x_t] + b_c) \tag{5}$$
$$O_t = \sigma(W_o.[h_{t-1}, x_t] + b_o) \tag{6}$$
$$C_t = f_t * C_{t-1} + i_t * \widetilde{C_t} \tag{7}$$
$$h_t = O_t * tanh(C_t) \tag{8}$$

where, x_t and h_t represent the input and output, respectively at the time-step t. C_t represents the cell state, with value 1 if it is fully active or 0 for inactive. σ is the sigmoid activation function. f_t, i_t, o_t and $\widetilde{C_t}$ are the outputs for the forget gate, input gate, output gate and the candidate gate, respectively. W_j and b_j are the weight and bias, respectively for the individual gates (*input gate, output gate, candidate gate, forget gate*).

3.4 Combined CNN and LSTM Model

It is reported that CNN performs better for some applications, whereas RNN performs better for the other sets of applications. In general, CNNs are hierarchical and RNNs (LSTM) are sequential. In particular, CNN can extract the

[8] the dotted lines indicate the other CNNs in the Figure, for space limitations we avoid this.

features and LSTM can learn sequential relations. To avail the richness of both the models we exploit the combination of both. We combine the outputs of both the CNN and LSTM. Vector representations of NT and NB are given to two stacked CNN and two stacked LSTM as inputs. Outputs of two stacked CNN and two stacked LSTM for NT and that for NB are concatenated and fed into the final layer with softmax activation function. The architecture of such model is shown in Figure (1: CNN+LSTM).

3.5 Modeling

Let, F_j be the vector representation of the input sequence in the last layer. F_j is the final layer with *Softmax* (Equ: 9) activation function to fit the number of classes. *Softmax* gives the prediction probability distribution for each class. If z, j and K are the vectors to the output layer, indexes of the output unit and the size of the input vector, respectively, then the *Softmax* function can be written as,

$$\sigma(z)_j = \frac{\exp^{z_j}}{\sum_{k=1}^{K} \exp^{z_k}} \tag{9}$$

If we represent the output vector as y_j, then it can be written as,

$$y_j = Softmax(w_j.F_j + b_j) \tag{10}$$

where, w_j and b_j are the weight matrices for the fully connected layer and the bias term, respectively. We use *Leaky ReLU, with 0.02 negative slope value* 3.5, as an activation function for the hidden layers.

$$f(n) = \begin{cases} n & \text{if } n>0 \\ \alpha.n & \text{if } n<0 \text{ and small value of } \alpha \end{cases}$$

We apply *Mean Square Error* for loss function on the prediction to minimize the error and *Adam* [18] as an optimizer (with an initial learning rate of 0.2) for all the experiments that we carried out.

4 Experiments

In this section, we discuss the details of experimental setup, results obtained, and discussions.

4.1 Data

We use the datasets of FNC-I. It contains *News title (NT)*, *News body (NB)* and *labels: agree, unrelated, discuss, disagree*. The labels essentially depict the relation between NT and NB. The statistics and the class distribution of the data are shown in Table 2. There are multiple bodies(documents) with different stance/class corresponding to a particular headline. We show one such example in Table 3. We correlate this problem with the problem of TE/NLI, and use

the classes as defined in [6]. We correlate the classes of FNC with NLI: *"Agree with Entailment", "Disagree with Contradiction" and "Discuss, Unrelated with Neutral"* after performing a rigorous analysis of the training instances contained in both the datasets, namely Standord Natural Language Inference (SNLI) [9] and the datasets of FNC-I.

Table 2. Distribution of classes in training and test set of FNC

Dataset	Classes				
	Example pairs	**Unrelated**	**Discuss**	**Agree**	**Disagree**
Training	49972	0.73131	0.17828	0.0736012	0.0168094
Test	25431	0.722032	0.17466	0.074833	0.027427

The Table 2 shows that the data is very imbalanced. To mitigate this problem, the shared task organizers came up with a scoring scheme. The scheme is basically two levels weighted scoring system. In the first level 25% weight is given for classifying *Unrelated* and *Related (Agree, Disagree and Discuss)* and 75% weight is given for classifying *Related* pairs as *Agree, Disagree and Discuss*. The score is called FNC-1.

Table 3. An example from the training dataset of FNC-I

Headline: El-Sisi denies claims he'll give Sinai land to Palestinians	
Stance	**News Body**
Agree	Al-Sisi has denied Israeli reports stating that .....
Disagree	Israel's Army Radio substantiated earlier claims that al-Sisi had offered Abbas an extended Gaza Strip.....
Discuss	Spokesman for Palestinian President Abbas,Tayeb Abdel Rahim, claimed that al-Sisi had not made an offer to extend the Gaza Strip,.....
Unrelated	Reports of more attacks in Yakutia bears on people 53-year-old Igor Nerungri Vorozhbitsyn ignored, they say, for his life more.....

4.2 Experimental Setup

Here, we discuss the experimental procedures. First, we perform data pre-processing followed by the vectorization of each word contained in NB/NT and after that the whole NB/NT.

Data Pre-Processing: We perform following pre-processing operations (by the library in python environment) on the dataset: removal of non-ASCII characters (Unicode data), URL's punctuations (Regular expression), stop-words removal from NT and NB (NLTK), replacement of all the numbers with their textual representation (*i.e. 200 is replaced by two hundred*) (Inflect), conversion of all

[9] https://nlp.stanford.edu/projects/snli/

the words into it's lower case form (inbuilt function) and chop words into its base form (LancasterStemmer).

Word Embedding: Distributed representations of words/sentences/documents [26, 21] are very helpful in any deep learning assisted language processing tasks, because it is very much efficient to capture hidden semantic structure. There are various word embedding methods. We make use of GloVe [10] [28] word embedding, which is pre-trained on the combined Wikipedia 2014 + Gigaword 5th Edition corpora for English words, where each word is expressed by a 300-dimensional vector(x_i). It is a count based model, leveraging word-occurrence matrix, which provides comprehensive lexical representation of the input.
Embedding of NT and NB: NTs and NBs are the collections of words. We consider the vector representation of each word contained in NT and NB using GloVe [11]. Out of vocabulary words are initialized as zeros. We set the dimension of the vector as 300 for all the experiments that we perform. Concatenation of word vectors for creating sentence embedding often suffers from the problem of high dimensionality. So, we take the average of all the words' vector representations contained in NT/NB to get the representation of NT/NB.

4.3 Results and Discussion

In this section, we report the evaluation results on the test set of FNC-I dataset in terms of FNC-1 score, overall F1, per class F1 etc., and compare the results with the state-of-the-art models. Results are shown in Table 4. We calculate the FNC-1 score following the guidelines as provided by the task organizer. The models based on CNN and the combined model perform better compared to the others. Using CNN, we extract features at the paragraph level, and these features are utilized to determine the relationship between NT and NB. LSTM reads and captures the sequential relationship between the words in the sentence and yields the final representation of the total paragraph at the last time step and individual relations at each time step. But, as we calculate the encoded representation for NT and NB into a single vector, we believe LSTM fails to learn any new sequential relation. This might be one of the reasons why CNN performs better compared to LSTM. However, the LSTM model produces the best F1 score for the disagree class. However, the combination of CNN and LSTM yields the best performance. It has been observed that CNN performs for a few examples and LSTM performs better for the other set of examples, i.e. there were few examples where CNN performs correctly but LSTM fails and the vice-versa[12].

Comparison with the existing systems: There were 50 out of 80 participants who were able to submit their systems exploiting various techniques. The first three best performing systems namely Talos Intelligences SOLAT in the

[10] http://nlp.stanford.edu/projects/glove/
[11] We find 2691 number of unique words whose WordEmbeddings are not present.
[12] for space limitations, we are avoiding showing those examples

Table 4. Results of state-of-the-art systems, proposed models, baseline model, and the human judgements

Sl No.	System/Team Name	FNC-1	F1	Agree	Disagree	Discuss	Unrelated
Best three systems							
1	**SOLAT in the SWEN**	0.8202	0.582	0.539	0.035	**0.760**	**0.994**
2	**Athene (UKP Lab)**	0.8197	0.604	0.487	0.151	**0.780**	**0.996**
3	**UCL Machine Reading**	0.8172	0.583	0.479	0.114	0.747	0.989
Proposed Models							
4	**MLP**	0.444	0.2848	0.091	0.0	0.1402	0.9077
5	**CNN**	0.7213	0.4777	0.3462	0.0	0.6328	0.9315
6	**LSTM**	0.5604	0.4150	0.0073	**0.5954**	0.1084	0.9489
7	**CNN+LSTM**	0.769	0.570	0.436	0.187	0.712	0.944
Baseline and Human Evaluation							
8	**Official Baseline**	**0.7520**	X	X	X	X	X
9	**HUMAN UPPER BOUND**	0.859	0.754	0.588	0.667	0.765	0.997

SWEN team [2], Team Athene[1], and The UCL Machine Reading (UCLMR) team [34] obtained the FNC scores as .8202, .8197, and .8172 respectively.

The best system consists of two sub-systems. The first one is like: embeddings of the headline and body are given to respective two one-dimensional convolutional neural networks. The respective outputs further feeding into an MLP with three hidden layers. The second one is gradient boosted decision trees based having five features. The second best team presented an MLPs based model having six hidden layers and a softmax layer with multiple handcrafted features. The third best system is also based on MLP but having one hidden layer with features like term-frequency of unigram, cosine similarity between TF-Inverse document frequency (IDF) vector of headline and body. Finally, they concatenated these two features. The proposed system obtained the best FNC-1 score with the combination of CNN and LSTM which is close to the state-of-the art FNC-1 score and able to beat the official baseline. However, our system is much simpler compared to the existing systems in the sense that our proposed models follow an end-to-end architecture and which avoids any manual feature engineering.

4.4 Error Analysis

We perform a very detailed analysis of the errors encountered by our proposed system.We describe the most commonly occurring errors below:

– The dataset is highly unbalanced and biased, and majority of the example pairs are with the *Unrelated* classes. So our system tends to predict the *Unrelated* class. We foster the measure of generating synthetic data to overcome this. It is observed that the data remain unbalanced even after creating synthetic data. Further creation of synthetic data causes over-fitting and deteriorate the accuracy.

- The presence of negation words like *"No/Not"* may change the meaning of a whole sentence even though it shares maximum lexical contents with the comparing sentence.
- It is observed that news texts are having an ample number of named entities and multi-word expressions. Specific techniques to handle these named entities and multi-word expressions need to be investigated.

5 Conclusion and Future Work

In this paper, we have proposed an approach to detect consistency between news title and news body by leveraging the concept of TE and deep learning. Evaluation on the benchmark dataset shows that a the combined model yields the best performance, which is closer to the state-of-the-art, but outperforms official baseline. The experiments divulge that TE is a good indicator to detect the consistency between NB and NT. Our future focus will be in the following directions.

- proposing an ensemble model with not only cosine similarity features but also external features like sentiment, TF-IDF, and other TE based features etc.
- a deep learning model with attention mechanism to get the knowledge of words which are significant compared to the others.
- an automated fake news detection system in multi modal scenario, as there are various fake news in different forms like *pictures, audios, videos in addition to textual* in numerous websites, blogs, and social media sites.

References

1. Andreas Hanselowski, Avinesh PVS, B.S., Caspelherr., F.: Description of the System Developed by Team Athene in the FNC-1, 2017. (2017)
2. Baird Sean, S.D., Yuxi., P.: Talos Targets Disinformation with Fake News Challenge Victory (2017)
3. Bajaj, S.: The Pope Has a New Baby! Fake News Detection using Deep Learning (2017)
4. Baly, R., Mohtarami, M., Glass, J., Màrquez, L., Moschitti, A., Nakov, P.: Integrating Stance Detection and Fact Checking in a Unified Corpus. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers). pp. 21–27. Association for Computational Linguistics, New Orleans, Louisiana (2018)
5. Biyani, P., Tsioutsiouliklis, K., Blackmer, J.: "8 Amazing Secrets for Getting More Clicks": Detecting Clickbaits in News Streams Using Article Informality. In: AAAI. pp. 94–100. Phoenix, Arizona, USA (2016)
6. Bowman, S.R., Angeli, G., Potts, C., Manning, C.D.: A Large Annotated Corpus for Learning Natural Language Inference. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 632–642. Association for Computational Linguistics, Lisbon, Portugal (2015)

7. Ciampaglia, G.L., Shiralkar, P., Rocha, L.M., Bollen, J., Menczer, F., Flammini, A.: Computational Fact Checking from Knowledge Networks. PloS one **10**(6), e0128–193 (2015)
8. Conroy, N.J., Rubin, V.L., Chen, Y.: Automatic Deception Detection: Methods for Finding Fake News. Proceedings of the Association for Information Science and Technology **52**(1), 1–4 (2015)
9. Dagan, I., Glickman, O., Magnini, B.: The PASCAL Recognising Textual Entailment Challenge. In: Proceedings of the First International Conference on Machine Learning Challenges: Evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment. pp. 177–190. MLCW'05, Springer-Verlag, Berlin, Heidelberg (2006)
10. Doddington, G.: Automatic Evaluation Of Machine Translation Quality Using N-gram Co-occurrence Statistics. In: Proceedings of the Second International Conference on Human Language Technology Research. pp. 138–145. HLT '02, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2002)
11. Ferreira, W., Vlachos, A.: Emergent: A Novel Data-set for Stance Classification. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 1163–1168. Association for Computational Linguistics, San Diego, California (2016)
12. Green, Jr., B.F., Wolf, A.K., Chomsky, C., Laughery, K.: Baseball: An Automatic Question-Answerer. In: Papers Presented at the May 9-11, 1961, Western Joint IRE-AIEE-ACM Computer Conference. pp. 219–224. IRE-AIEE-ACM '61 (Western), ACM, New York, NY, USA (1961)
13. Grishman, R.: Information Extraction: Techniques And Challenges. In: International Summer School on Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology. pp. 10–27. SCIE '97, Springer-Verlag, London, UK, UK (1997)
14. Hochreiter, S., Schmidhuber, J.: Long Short-Term Memory. In: Natural Language Inference. vol. 9(8), p. 17351780. Neural computation. (1997)
15. Hu, B., Lu, Z., Li, H., Chen, Q.: Convolutional neural network architectures for matching natural language sentences. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 27, pp. 2042–2050. Curran Associates, Inc., Palais des Congrs de Montral, Montral, CANADA (2014)
16. Kim, Y.: Convolutional Neural Networks for Sentence Classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1746–1751. Association for Computational Linguistics, Doha, Qatar (2014)
17. Kim, Y.: Convolutional Neural Networks for Sentence Classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1746–1751. Association for Computational Linguistics, Doha, Qatar (2014)
18. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. arXiv preprint arXiv:1412.6980 (2014)
19. Knight, K., Marcu", D.: Summarization Beyond Sentence Extraction: A Probabilistic Approach To Sentence Compression. Artificial Intelligence **139**(1), 91 – 107 (2002)
20. Kumar, S., West, R., Leskovec, J.: Disinformation on the Web: Impact, Characteristics, and Detection of Wikipedia Hoaxes. In: Proceedings of the 25th international conference on World Wide Web. pp. 591–602. International World Wide Web Conferences Steering Committee (2016)

21. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: Xing, E.P., Jebara, T. (eds.) Proceedings of the 31st International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 32, pp. 1188–1196. PMLR, Bejing, China (22–24 Jun 2014)
22. Li, X., Croft, W.B.: Novelty Detection Based on Sentence Level Patterns. In: Proceedings of the 2005 ACM CIKM International Conference on Information and Knowledge Management, Bremen, Germany, October 31 - November 5, 2005. pp. 744–751 (2005)
23. Lukasik, M., Cohn, T., Bontcheva, K.: Classifying Tweet Level Judgements of Rumours in Social Media. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 2590–2595. Association for Computational Linguistics, Lisbon, Portugal (2015)
24. MacCartney, B., Manning, C.D.: Natural Logic for Textual Inference. In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing. pp. 193–200. RTE '07, Prague, Czech Republic (2007)
25. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, New York, NY, USA (2008)
26. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed Representations of Words and Phrases and their Compositionality. In: Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 26, pp. 3111–3119. Curran Associates, Inc., Lake Tahoe, Nevada, USA (2013)
27. Mohtarami, M., Baly, R., Glass, J., Nakov, P., Màrquez, L., Moschitti, A.: Automatic Stance Detection using End-to-End Memory Networks. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). pp. 767–776. Association for Computational Linguistics, New Orleans, Louisiana (2018)
28. Pennington, J., Socher, R., Manning, C.: Glove: Global Vectors for Word Representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1532–1543. Association for Computational Linguistics, Doha, Qatar (2014)
29. Pfohl, S., Triebe, O., Legros, F.: Stance Detection for the Fake News Challenge with Attention and Conditional Encoding (2017)
30. Pomerleau, D., Rao., D.: The Fake News Challenge: Exploring how Artificial Intelligence Technologies could be Leveraged to Combat Fake News. (2017)
31. Qazvinian, V., Rosengren, E., Radev, D.R., Mei, Q.: Rumor has it: Identifying Misinformation in Microblogs. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing. pp. 1589–1599. Association for Computational Linguistics, Edinburgh, Scotland, UK. (2011)
32. Rakholia, N., Bhargava, S.: Is it true?–Deep Learning for Stance Detection in News (2017)
33. Rashkin, H., Choi, E., Jang, J.Y., Volkova, S., Choi, Y.: Truth of Varying Shades: Analyzing Language in Fake News and Political Fact-Checking. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. pp. 2931–2937. Association for Computational Linguistics, Copenhagen, Denmark (2017)
34. Benjamin Riedel, Isabelle Augenstein, Georgios P. Spithourakis, Sebastian Riedel: A Simple but Tough-to-Beat Baseline for the Fake News Challenge Stance Detection Task. CoRR **abs/1707.03264** (2017)

35. Rocktäschel, T., Grefenstette, E., Hermann, K.M., Kocisky, T., Blunsom, P.: Reasoning about Entailment with Neural Attention. In: International Conference on Learning Representations (ICLR) (2016)
36. Rubin, V.L., Chen, Y., Conroy, N.J.: Deception Detection for News: Three Types of Fakes. In: Information Science with Impact: Research in and for the Community - Proceedings of the 78th ASIS&T Annual Meeting, ASIST 2015, St. Louis, Missouri, Missouri, USA, October 6-10, 2015. pp. 1–4 (2015)
37. Ruder, S., Glover, J., Mehrabani, A., Ghaffari, P.: 360 Stance Detection. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations. pp. 31–35. Association for Computational Linguistics, New Orleans, Louisiana (2018)
38. Silverman, C.: Lies, damn lies and viral content (2015)
39. Sukhbaatar, S., szlam, a., Weston, J., Fergus, R.: End-to-end memory networks. In: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (eds.) Advances in Neural Information Processing Systems 28, pp. 2440–2448. MONTREAL, CANADA (2015)
40. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to Sequence Learning with Neural Networks. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 27, pp. 3104–3112. Montral, QC, Canada (2014)
41. Tavernise, S.: As Fake News Spreads Lies, More Readers Shrug at the Truth. New York Times **6** (2016)
42. Vlachos, A., Riedel, S.: Fact checking: Task definition and dataset construction. In: Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science. pp. 18–22 (2014)
43. Wang, S., Jiang, J.: Learning Natural Language Inference with LSTM. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 1442–1451. Association for Computational Linguistics, San Diego, California (2016)
44. Wang, W.Y.: "liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 422–426. Association for Computational Linguistics, Vancouver, Canada (2017)
45. Yin, W., Schütze, H., Xiang, B., Zhou, B.: Abcnn: Attention-based convolutional neural network for modeling sentence pairs. arXiv preprint arXiv:1512.05193 (2015)
46. Zaremba, W., Sutskever, I., Vinyals, O.: Recurrent Neural Network Regularization. arXiv preprint arXiv:1409.2329 (2014)
47. Zhao, Z., Resnick, P., Mei, Q.: Enquiring Minds: Early Detection of Rumors in Social Media from Enquiry Posts. In: Proceedings of the 24th International Conference on World Wide Web. pp. 1395–1405. International World Wide Web Conferences Steering Committee, Florence, Italy (2015)