Validation of Facts Against Textual Sources

Vamsi Krishna Pendyala Simran sinha Satya Prakash Shriya Reddy Anupam Jamatia
Department of Computer Science and Engineering
National Institute of Technology
Agartala, Tripura, India
{krishnapendiala, simsinha32}@gmail.com
{satyaprakash30497, shriyavipul90, anupamjamatia}@gmail.com

Abstract

In today's world, the spreading of fake news has become facile through social media which diffuses rapidly and can be believed easily. Fact Checkers or Fact Verifiers are the need of the hour. In this paper, we propose a system which would verify a claim(fact) against a textual source provided and classify the claim to be true, false, out-of-context or inappropriate with respect to that source. This would help us to verify a fact as well as know about the source of our knowledge base against which the fact is being verified. We used a two-step approach to achieve our goal. First step is about retrieving the evidence related to the claims from the textual source. Next step is the classification of the claim as true, false, inappropriate and out of context with respect to the evidence using a modified version of textual entailment module. The accuracy of the best performing system is 64.95%.

1 Introduction

Fact Checking is one of the biggest buzzwords of this era. Many a time, the news transmitted through social media can be moulded and altered when so many people share information and it is hard to discern between fact and fiction. So there is a need to verify every piece of information we observe in our day-to-day life to be true or false. A solution to this problem is, we check each claim or fact manually against a reliable source and then label the claim or fact to be either true or false which is time consuming for large data. Vlachos and Riedel (2014) discussed the fact verification process as an ordinal text classification task, where they created a data-set using the manually annotated data present on sites like PolitiFact, FactCheck, FullFact. The FakeNews Challenge[1] by Riedel et al. (2017) addresses fact-checking as a simple instance detection problem, which mainly checks whether the given instance is in accordance with the article or not. Recently Thorne et al. (2018a) published a huge data set to deeply understand the process of large scale fact-checking. All of the above approaches rely completely upon the source against which data is to be verified, but in some cases, the source text might contain limited or no amount of information about the claim and a claim might be *Out-of-Context* of this source text. This leads to a problem of classifying a claim to be within the scope/context of the source text or not. Concretely, we do need a system which would not only classify a claim to be true or false but also check whether the source text is sufficient enough to classify the claim. To address this issue, in this paper, we come with an approach to verify facts or claims against a reliable source and classify them into 4 different classes.
In our approach, a claim or fact is classified as *True* (if a proper supporting evidence is available from the source text), *False* (if a contradicting evidence is available from the source text), *Inappropriate* (nothing can be concluded about the claim based upon evidence retrieved) or *Out-of-Context* (out of the scope of the source text). Prior to the classification, the evidence is retrieved from the

[1] www.fakenewschallenge.org/

Proceedings of Recent Advances in Natural Language Processing, pages 895–903,
Varna, Bulgaria, Sep 2–4, 2019.
https://doi.org/10.26615/978-954-452-056-4_104

textual source related to a given claim. If the textual source has no knowledge about the claim, it would be labelled as '*Out-of-Context*' for example if we have a source text about sports and we need to verify a claim related to politics then the claim is *Out-of-Context*. Although if it has some information about the claim it would further be classified as *True*, *False* or *Inappropriate*. The result would be '*True*' if the source supports the claim, it would be '*False*' if the source opposes the claim and it would be labelled as '*Inappropriate*' if the evidence is not sufficient to conclude the classification of the claim, e.g., knowing that Michael Jackson was a singer, we cannot infer whether his children will be singers. This is something inappropriate to the information provided. Hence the claim 'Michael Jackson's children would be singers' is an inappropriate claim for the evidence 'Michael Jackson was a Singer'. This labelling would not only help in knowing about the claim but also helps us to know about our textual source as well. The labels *Inappropriate* and *Out-of-Context* claim tell us whether the textual source has enough information to conclude about the claim or not, if a claim is *Out-of-Context* then the domain of our source text can be expanded to answer the claim. *Inappropriate* means that we do have required knowledge about the claim and also cannot conclude anything based upon this available knowledge. The best performing version of our system is seen to be giving 63.06% accuracy.

In Section 2 we have mentioned various works done by different authors related to fact extraction and verification. Section 3 explains the dataset collection and the system architecture to classify claims against a textual source. Section 4 describes the stages of this experiment which are evidence retrieval, similarity measures used to obtain the relation between claim and evidence, the first level of classification of the claim into in-context and out-context, then the further classification of in-context claims to *True*, *False* and *Inappropriate* claim. In Section 5 we discussed error analysis of our system along with the results obtained on using different classification algorithms on our data for classification purpose and comparative measure between different models. In Section 6 we conclude about our model with its practical usage in the real world.

2 Related Work

There has been a substantial amount of work done in the field of fact verification. Vlachos and Riedel (2014) provided the first dataset related to fact verification containing 211 labelled claims in the political domain with evidence hyperlinks. An alternative is Wang (2017) which released a dataset called LIAR dataset for detecting fake news, which contains 12.8K claims labelled manually using POLTIFACT.COM[2] on different context. Alhindi et al. (2018) extended the LIAR dataset and labelled a claim using the speaker related metadata without using the evidence. Basically, they used the justification given by the humans at the end of the article in the summary. Modelling the extracted justification along with the claim yielded better results rather than using a machine learning model for binary classification and a six way classification. Ferreira and Vlachos (2016) later presented a new modified dataset known as Emergent, where they had 300 claims and 2,595 related articles and they came to the conclusion that fact verification can also be treated as Natural Language Inference Task, as they used textual entailment to predict whether the article supports the claim or not. The latest large scale dataset is prepared Thorne et al. (2018a) which was annotated manually and used for verification against textual sources. It contains 185,441 claims generated from Wikipedia. These claims were classified Supported, Refuted and Not Enough Info by annotators. In this, Recognizing Textual Entailment (RTE) component was proceeded by an evidence retrieval module. The accuracy measured found to be 31.87% if evidence was taken into consideration and 50.91% if evidence is ignored. The only drawback of this system is its restriction to the Wikipedia domain. Thorne and Vlachos (2018) conducted a survey on automated fact checking research stemming using natural language processing and other related fields. According to this survey, the inputs for verification system play a vital role. Evidence retrieval plays a vital role in solving the fact verification problem. Fact checking requires the apt evidence against which sentences can be predicted to be true or false. Chen et al. (2017a) provides a framework for open domain question answering upon Wikipedia and SQuAD data set. This involved machine reading along with the document

[2] www.politifact.com

retrieval and then identifying the answers. We deal with a similar retrieval problem like open domain Question Answering, which would be succeeded by verification using textual entailment. Natural Language Inference is basically a task to find out whether a hypothesis entails, contradicts or is neutral about the claim. There have been recent developments in these fields like the SNLI dataset for learning natural language inference built by Bowman et al. (2015). Different neural NLI models (Nie and Bansal (2017); Parikh et al. (2016); Chen et al. (2017b); Gong et al. (2018)) that achieve promising performance. Parikh et al. (2016) has the highest accuracy on the Stanford Natural Language Inference task. We used the similar approach as the FEVER (Thorne et al., 2018c) but, instead of a three class classification we have extended it to a four class classification namely *True*, *False*, *Inappropriate* and *Out-of-Context*. These two modules combined together help us in validation of a fact. In 2018, a shared task known as the FEVER Shared Task (Thorne et al., 2018b) was held which dealt with the fact verification problem. The FEVER shared task is closely related to our work as it uses the same two modules. The following are some of the systems that participated in the FEVER shared task: Nie et al. (2018) have scored the maximum of 64% accuracy in the FEVER shared task, in which they used the neural semantic matching networks. For both the evidence retrieval and RTE model they enhanced the working using the neural networks. Hidey and Diab (2018) known as the Team Sweepers, made the evidence retrieval system better using lexical tagging and syntactic similarity. They used multitask learning and trained both the components together and set the parameters in a way using reinforcement learning so that it can first find sentences related to the claim and then find their relation with the claim. DeFactoNLP (Reddy et al., 2018) aimed at retrieving the evidence for the valuation of the claim from Wikipedia. The retrieval of documents which is considered as evidence is done by TF-IDF vectors of the claim and the sentences in the documents followed by inputting them to a textual entailment recognition module. Then the Random forest classifier is used for the classification of the claim. Lee et al. (2018) have introduced a method by developing a neural ranker using decomposable attention model and lexical tagging instead of TF-IDF for evidence retrieval part. Lexical tagging is done by using two lexical tags name such as Parts-of-Speech and Named Entity Recognition to enhance the performance.

3 System Architecture

In this section, we discuss the overview of our system and our approach for classifying a claim based upon a particular source text. Our approach is divided into two stages: Evidence Retrieval and Classification of claim as *True*, *False*, *Inappropriate* or *Out-of-Context* using a textual entailment module. The reason for using textual entailment is because it precisely gives us a relationship between an evidence and a claim. In the first stage i.e., evidence retrieval, given a claim, we find its TF-IDF vectors corresponding to the source text against which it is being verified.

Later we find out the cosine similarity between the TF-IDF vector of a claim to each of the sentences present in the source text. Then, we filter out the top four sentences which have the highest cosine similarity values and consider this as the evidence for that particular claim as discussed in section 3.2. The reason for considering top 4 sentences is that they closely correspond to the nearest sentences to the claim. In the next stage, the extracted evidence and the present claim are passed into a Textual Entailment module which returns the probabilities of two texts entailing, contradicting or neutral towards each other. These probabilities along with other variables discussed later are used as a feature vector for our classification model. The entire claim classification process is explained in 3.3. Section 3.1 describes the process of preparation of the dataset.

3.1 Dataset

Due to the uniqueness of our classification, we were supposed to either prepare our own dataset or modify an existing standard dataset for serving our purpose. Here, we have done both, we modified a dataset known as the SICK dataset and prepared a new dataset called the NITA dataset.

3.1.1 SICK Dataset

The main target of our experiment was to classify a claim based upon its evidence, so we required a dataset consisting of sentence pairs and a correlation between these two sentences. Hence we used a publicly available dataset known as the SICK dataset. The SICK-2014 dataset (Marelli et al., 2014) was introduced as Task 1 of the SemEval

ID	Sentence1	Sentence2	Entailment	Score
23	A group of kids is playing in a yard and an old man is standing in the background	A group of boys in a yard is playing and a man is standing in the background	yes	4.5
14	A brown dog is attacking another animal in front of the man in pants.	Two dogs are fighting.	unknown	3.5
13	Two dogs are wrestling and hugging.	There is no dog wrestling and hugging.	no	3.3

Table 1: Sample SICK dataset with entailment labels and relatedness scores.

Source Text	Claim/Fact	Label
The Lion King	Mufasa is Father of Simba	True
The Lion King	Ram killed Ravan	Out of Context
The Lion King	Cats hate Lions	Inappropriate Claim.

Table 2: Sample NITA dataset.

2014 conference and in contrast to SNLI (Bowman et al., 2015), it is geared at specifical benchmarking semantic compositional methods, aiming to capture only similarities on purely language and common knowledge level, without relying on domain knowledge, and there are no named entities or multi-word idioms. It consists of total 10,000 pairs of sentences.

We modified the SICK dataset as per our classification by adding two more columns to it. We manually labelled a claim and an evidence pair to be in-context or out-context based upon their relatedness score as given in the dataset, which indicates the semantic similarity of these pair of sentenced. Firstly, we labelled all the pairs with relatedness score less than 3 as *Out-of-Context* and other claims as *True*, *False* or *Inappropriate* based upon their textual entailment labels provided by the SICK dataset.

3.1.2 NITA Dataset

After considering SICK dataset we even wanted to develop our own dataset consisting of source texts and claims along with their labels as follows:

- **Source Text Collection**: We collected some short stories and articles related to *sports, movies, mythology, moral stories, Wikipedia articles* in English language and considered them as source texts. The total number of source texts collected in this way turned out to be 53 .
- **Claim Generation**: Corresponding to these 53 stories/articles/textual content, we prepared a total of 928 claims. The purpose was to generate claims about a single fact which could be arbitrarily complex and allowed for a variety of expressions for the entities. The claims were generated based upon every source text. For example, consider "The rabbit tortoise race" as the source text, one of the claims related to this source text can be "Rabbit won the race".
- **Claim Labelling**: Classifying whether a claim is *True*, *False*, *Inappropriate* or *Out-of-Context* based on the evidence from source text was done at this stage. We checked every claim manually with respect to its source text and labelled the claim accordingly. The labelling is done as per the meaning of each label which was discussed in the introduction section.
- **Dataset Validation**: Considering the complexity of labelling of claims, we considered validation of the data set generated by us. For this purpose we tried to analyse the labels we gave to each claim, where labels generated by one person were analysed by other to establish an inter annotator agreement. We considered around 30% that is 240 claims for this validation process and calculated the Fleiss k score (Fleiss, 1971) to be 0.876.

LABEL	No. of Claims
TRUE	170
FALSE	170
OUT-OF-CONTEXT	420
INAPPROPRIATE	168
Total No. of Claims	**928**

Table 3: NITA Dataset Splitting based upon Labels

3.2 Evidence Retrieval

We used the concept of Document Retrieval from the DrQA system (Chen et al., 2017a). Firstly, we find out the Term Frequency-Inverse Document Frequency (TFIDF) vectors (Hiemstra, 2000) for a claim and the sentences of the source text. We then calculate the cosine similarity between the claim and each sentence. Thereafter, we pick the top four similar sentences based on the cosine similarity between the bigram TF-IDF vectors of the sentences and the claim. These sentences are finally chosen as possible sources of *evidence*. Now, we are left with claim and evidence pairs.

3.3 Classification

In this final stage, we classify all the claims to be *True*, *False*, *Inappropriate* or *Out-of-Context* using machine learning classification models. The features for this classification are obtained by passing a claim and an evidence to a textual entailment module in order to obtain probabilities of entailment, contradiction and neutrality between claim and evidence. The RTE is the process of determining whether a text fragment (Hypothesis H) can be inferred from another fragment (Text T) (Sammons et al., 2012). The RTE module receives the claim and the set of possible evidences from the previous stages. Let there be 'n' possible sources of evidence for verifying a claim. For the i^{th} possible evidence, let s_i denote the probability of it entailing the claim, let r_i denote the probability of it contradicting the claim, and let u_i be the probability of it being uninformative. The RTE module calculates each of these probabilities. The SNLI corpus (Bowman et al., 2015) is used for training the RTE model. This corpus is composed of sentence pairs T, H where T corresponds to the literal description of an image and H is a manually created sentence. If H can be inferred from T, the "Entailment" label is assigned to the pair. If H contradicts the information in T, the pair is labelled as "Contradiction". Otherwise, the label 'Neutral' is assigned. We chose to employ the state-of-the-art RTE by (Parikh et al., 2016). We selected this because at the time of development of this work, it was one of the best performing systems on the task with publicly available code.

For a particular claim *c* and an evidence *e* let s_i denote the probability of it entailing the claim, let r_i denote the probability of it contradicting the claim, and let u_i be the probability of it being uninformative returned by textual entailment. Below are some variables we considered for our convenience:

$$cs_i = \begin{cases} 1 & \text{if } s_i \geq r_i \text{ and } s_i \geq u_i \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

$$cr_i = \begin{cases} 1 & \text{if } r_i \geq s_i \text{ and } r_i \geq u_i \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

$$cu_i = \begin{cases} 1 & \text{if } u_i \geq s_i \text{ and } u_i \geq r_i \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

$$CosineSimilarity = \left\{ cos(\theta) = \frac{C.E}{||C||||E||} \right. \tag{4}$$

The similarity variable used here is cosine similarity between claim and evidence. The value C and E denote the vector notation of claim *c* and evidence *e* based upon their word frequency. Consider the cosine similarity between claim and evidence *i* to be S_i. Using above variables we form a feature vector for each claim and evidence pair for the classification model *i* as:

$$\textit{feature vector} = < s_i, r_i, u_i, cs_i, cr_i, cu_i, S_i >$$

The above *feature vector* give us an understanding of how closely two statements are related i.e., a relationship between claim and evidence. Some statements which are a negation to each other may have high cosine similarity but then their contradiction probability would be high which would help the learning algorithm to classify claims accurately. We used both the datasets i.e., the SICK dataset and the NITA dataset, along with the above mentioned feature vector for training and testing purpose of various machine learning classification models like Naïve Bayes, Logistic Regression, Support Vector Machine, Random Forest and Multi-Level Perceptron. These models were used as they are widely used in the industry for practical applications.

4 Experiments

As mentioned in section 3, our model of fact verification consists of two stages:

1. *Retrieving evidence related to the claim from the source text.*

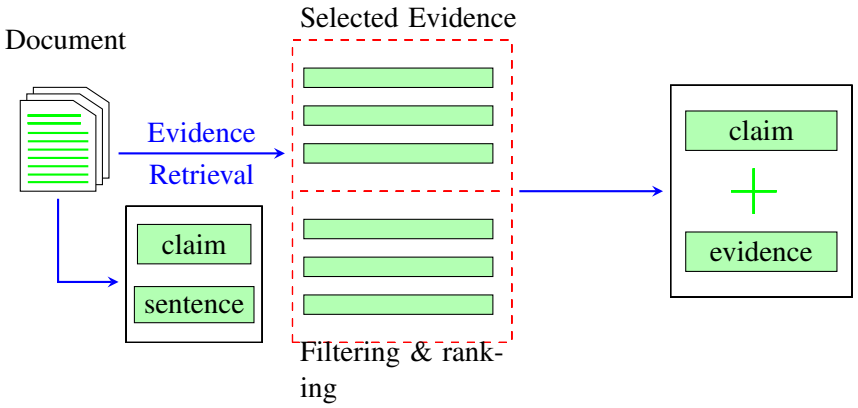

Figure 1: System Overview: Document Retrieval, Sentence Selection, and Claim Verification.

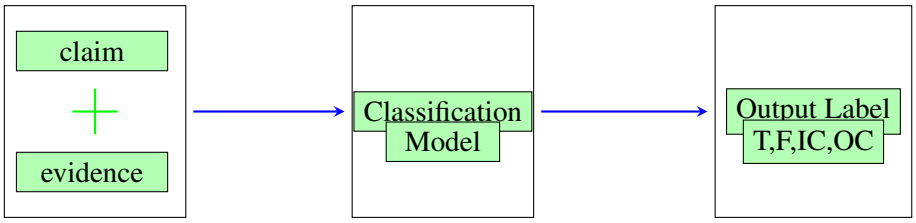

Figure 2: System Overview: Claim Classification Process.

Relatedness		**Entailment**	
[1-2) range	923 (10%)	NEUTRAL	5595 (57%)
[2-3) range	1373 (14%)	CONTRADICTION	1424 (14%)
[3-4) range	3872 (39%)	ENTAILMENT	2821 (29%)
[4-5) range	3672 (37%)		

Table 4: Distribution of SICK sentence pairs for each gold relatedness level and entailment label.

2. *Classifying a claim to be True, False, Inappropriate and Out-of-Context with respect to the source text.*

4.1 Evidence Retrieval

We used the uni-gram TF-IDF vector of sentences of the source text and the claim and computed the cosine similarity between various sentences from source text and claim. Based upon their cosine similarities, we selected a concatenation of top four sentences as evidence for the claim from the source text. After the evidence is retrieved, we are now left with a claim and an evidence upon which classification of claim is to be carried out in the next stage. In NITA dataset, we have a column describing the name of source text from which we wish to derive evidence from the claim.

4.2 Classification of Claims

In this stage we classify the claims using textual entailment module. For accomplishing the textual entailment task, we used the decomposable attention model developed by Parikh et al. (2016). This model was trained and tested upon the Stanford Natural Language Inference (SNLI) Corpus and has a test accuracy of 86.8% . We used this model to obtain probabilities of entailment, contradiction, and neutrality between a claim and evidence and further developed more variables as discussed in Section 3. The feature vector along with modified SICK data was passed into various classification models. We used the same approach for the NITA dataset. The test and train dataset split for both the above experiments was 60% training, 20% cross validation and 20% testing. The results of experiments on both the datasets are as in Table 5 which consists of the weighted average of all

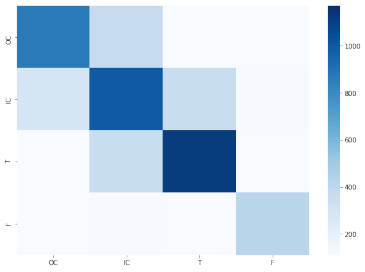

Figure 3: Random Forest Confusion Matrix for SICK Dataset

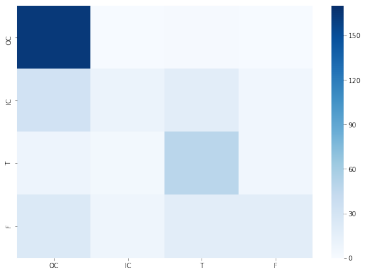

Figure 4: Random Forest Confusion Matrix for NITA Dataset

the results.

SICK dataset consists of pair of sentences and their relationship, hence evidence retrieval part for this dataset is skipped. The highest accuracy model observed with both the datasets is the "Random Forests" model with 100 trees and a maximum depth of 5.

5 Error Analysis

On observing the results it was found that the false claims were the most error-prone and have the least correct results. The SICK dataset gave better results in comparison to the NITA dataset prepared by us which is depicted by the random forest confusion matrix in Figure 3 and 4. The reason behind this could be that the evidence retrieved in the evidence retrieval part was not appropriate consisting of punctuation symbols and the other unwanted context in its text. On observing the results it was found that the false claims were the most error-prone and have the least correct results. The possible reasons for this could be the low probability rate of contradiction returned by the textual entailment module and also high cosine similarity, because in some instances the claim and evidence pair can be a negation of each other and hence have high word similarity. Next, upon observing results produced by the classification model, we saw that most inappropriate claims were classified as true and some true claims were classified as inappropriate. This ambiguity is mainly due to evidence supporting a claim partially, the probability of entailment for this would be high but due to variance in cosine similarity between claim and evidence there can arise an ambiguity. The overall system performance is at par with other existing fact verification systems as mentioned in section in terms of accuracy. Further modifications to improve the performance of the system are discussed in the next section .

6 Conclusion and Future Scope

The uniqueness in our approach is classifying a fact into four classes, this not only gives information about the fact whether it is true or false but also gives us an insight whether the source text we are using is limited. The *Out-of-Context* label particularly tries to validate the scope of our source text whether the source is enough to classify a particular fact/claim.

Here we discussed about the modification of the SICK dataset as per our requirement along with our approach of carrying out the process of classifying our claims. Compared to the existing fact verification systems such as FEVER systems which classifies a claim only into 3 classes, our model classifies a claim into 4 classes giving additional information. Our system can have many practical applications like subjective paper correction, fake news identifier, social media fact checking, etc. We believe that our system will provide a stimulating challenge for claim/fact extraction and verification systems and be effective for knowing about the scope of the source.

In future, we wish to tackle the problem of restricting our system for a particular source text, by enabling the system to extract evidence from a larger source like the internet itself by using various APIs provided by prominent search engines such as Google API to get appropriate evidence. Next thing we wish to implement further as a modification in our system is come up with a larger dataset comprising of our 4 class classification labels to train our system for better accuracy.

Data-set	Model	Precision	Recall	F1-score	Accuracy
SICK	Naive Bayes	0.53	0.53	0.50	0.535
	SVM	0.58	0.58	0.57	0.582
	Random Forest	0.63	0.63	0.63	0.630
	Logistic Regression	0.57	0.58	0.57	0.577
	MLP	0.63	0.62	0.63	0.624
NITA	Naive Bayes	0.61	0.63	0.59	0.631
	SVM	0.62	0.65	0.61	0.649
	Random Forest	0.61	0.65	0.61	0.649
	Logistic Regression	0.61	0.65	0.60	0.649
	MLP	0.61	0.64	0.61	0.644

Table 5: Classification result using various classification models

References

Tariq Alhindi, Savvas Petridis, and Smaranda Muresan. 2018. Where is your evidence: Improving fact-checking by justification modeling. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*. pages 85–90.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, pages 632–642.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017a. Reading wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Vancouver, Canada, pages 1870–1879.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017b. Enhanced LSTM for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Vancouver, Canada, pages 1657–1668.

William Ferreira and Andreas Vlachos. 2016. 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*. Association for Computational Linguistics, San Diego, California, pages 1163–1168.

J. L. Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin* 76(5):378–382.

Yichen Gong, Heng Luo, and Jian Zhang. 2018. Natural language inference over interaction space. In *International Conference on Learning Representations*.

Christopher Hidey and Mona Diab. 2018. Team SWEEPer: Joint sentence extraction and fact checking with pointer networks. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*. Brussels, Belgium, pages 150–155.

Djoerd Hiemstra. 2000. A probabilistic justification for using tfxidf term weighting in information retrieval. *International Journal on Digital Libraries* 3(2):131–139.

Nayeon Lee, Chien-Sheng Wu, and Pascale Fung. 2018. Improving large-scale fact-checking using decomposable attention models and lexical tagging. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. pages 1133–1138.

Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli. 2014. Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. Association for Computational Linguistics, Dublin, Ireland.

Yixin Nie and Mohit Bansal. 2017. Shortcut-stacked sentence encoders for multi-domain inference. In *Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP*. Association for Computational Linguistics, Copenhagen, Denmark, pages 41–45.

Yixin Nie, Haonan Chen, and Mohit Bansal. 2018. Combining fact extraction and verification with neural semantic matching networks. *arXiv preprint arXiv:1811.07039* .

Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, pages 2249–2255.

Aniketh Janardhan Reddy, Gil Rocha, and Diego Esteves. 2018. Defactonlp: Fact verification using entity recognition, tfidf vector comparison and decomposable attention. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*. Association for Computational Linguistics, pages 132–137.

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 preprint arXiv:1707.03264* .

Mark Sammons, Vinod Vydiswaran, and Dan Roth. 2012. Recognizing textual entailment. *Multilingual Natural Language Applications: From Theory to Practice* pages 209–258.

James Thorne and Andreas Vlachos. 2018. Automated fact checking: Task formulations, methods and future directions. In *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, pages 3346–3359.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018a. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, pages 809–819.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018b. The Fact Extraction and VERification (FEVER) shared task. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018c. Proceedings of the first workshop on fact extraction and verification (fever). In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*.

Andreas Vlachos and Sebastian Riedel. 2014. Fact checking: Task definition and dataset construction. In *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*. pages 18–22.

William Yang Wang. 2017. "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)*. Association for Computational Linguistics, Vancouver, Canada, pages 422–426.