# An End-to-End Multi-task Learning Model for Fact Checking

**Sizhen Li** and **Shuai Zhao** and **Bo Cheng**
State Key Laboratory of networking and switching technology,
Beijing university of posts and telecommunications

**Hao Yang**
2012 Labs, Huawei Technologies CO., LTD

## Abstract

With huge amount of information generated every day on the web, fact checking is an important and challenging task which can help people identify the authenticity of most claims as well as providing evidences selected from knowledge source like Wikipedia. Here we decompose this problem into two parts: an entity linking task (retrieving relative Wikipedia pages) and recognizing textual entailment between the claim and selected pages. In this paper, we present an end-to-end multi-task learning with bi-direction attention (EMBA) model to classify the claim as "supports", "refutes" or "not enough info" with respect to the pages retrieved and detect sentences as evidence at the same time. We conduct experiments on the FEVER (Fact Extraction and VERification) paper test dataset and shared task test dataset, a new public dataset for verification against textual sources. Experimental results show that our method achieves comparable performance compared with the baseline system.

## 1 Introduction

When we got news from newspapers and TVs which was thoroughly investigated and written by professional journalists, most of these messages are well-found and trustworthy. However, with the popularity of the internet, there are 2.5 quintillion bytes of data created each day at our current pace[1]. Everyone online is a producer as well as a recipient of these emerging information, and some of them are incorrect, fabricated or even with some evil purposes. Most time it is difficult for us to figure out the truth of those emerging news without professional background and enough investigation. Fact checking, which firstly has been produced and received a lot of attention in the industry of journalism, mainly verifying the speeches of public figures, is also important for other domains, e.g. wrong common-sense correction, rumor detection, content review etc.

With the increasing demand for automatic claim verification, several datasets for fact checking have been produced in recent years. Vlachos and Riedel (2014) are the first to release a public fake news detection and fact-checking dataset from two fact checking websites, the fact checking blog of Channel 4[2] and the True-O-Meter from PolitiFact[3]. This dataset only includes 221 statements. Similarly, from PolitiFact via its API, Wang (2017) collected LIAR dataset with 12.8K manually labeled short statements, which permits machine learning based methods used on this dataset. Both dataset don't include the original justification and evidence as it was not machine-readable. However, just verifying the claim based on the claim itself and without referring to any evidence sources is not reasonable and convincing.

In 2015, Silverman launched the Emergent Project[4], a real-time rumor tracker, part of a research project with the Tow Center for Digital Journalism[5] at Columbia University. Ferreira and Vlachos (2016) firstly proposed to use the data from Emergent Project as Emergent dataset for rumor debunking, which contains 300 rumored claims and 2,595 associated news articles. In 2017, the Fake news challenge (Pomerleau and Rao, 2017) consisted of 50K labeled claim-article pairs similarly derived from the Emergent Project. These two dataset stemmed from Emergent Project alleviate the fact checking task by detecting the relationship between claim-article pairs. However, in more common situation, we

[1] https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read/#62a15bba60ba

[2] http://blogs.channel4.com/factcheck/

[3] http://www.politifact.com/truth-o-meter/statements/

[4] http://www.emergent.info/

[5] https://towcenter.org/



*Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 138–144
Brussels, Belgium, November 1, 2018. 

are dealing with plenty of claims themselves online without associated articles which can help to verify the claims.

Fact Extraction and VERification (FEVER) dataset (Thorne et al., 2018) consists of 185,445 claims manually verified against the introductory sections of Wikipedia pages and classified as SUPPORTED, REFUTED or NOTENOUGHINFO. For the first two classes, the dataset provides combination of sentences forming the necessary evidences supporting or refuting the claim. Obviously, this dataset is more difficult than existing fact-checking datasets. In order to achieve higher FEVER score, a fact-checking system is required to classify the claim correctly as well as retrieving sentences among more than 5 million Wikipedia pages jointed as correct evidence supporting the judgement.

The baseline method of this task comprises of three components: document retrieval, sentence-level evidence selection and textual entailment. For the first two retrieval components, the baseline method uses document retrieval component of DrQA (Chen et al., 2017) which only relies on the unigram and bigram TF-IDF with vector similarity and don't understand semantics of the claim and pages. So, we find that it extracts lots of Wikipedia pages which are unrelated to the entities described in claims. Besides, similarity-based method prefer extracting supporting evidences than refuting evidences. For the recognizing textual entailment (RTE) module, on one hand, the previous retrieval results limit the performance of the RTE model. On the other hand, the selected sentences concatenated as evidences may also confuse the RTE model due to some contradictory information.

In this paper, we introduce an end-to-end multi-task learning with bi-direction attention (EMBA) model for FEVER task. We utilize the multi-task framework to jointly extract evidences and verify the claim because these two sub-tasks can be accomplished at the same time. For example, after selecting relative pages, we carefully scan these pages to find supporting or refuting evidences. If we find some, the claim can be labeled as SUPPORTS or REFUTES immediately. If not, the claim will be classified as NOTENOUGHINFO after we read pages completely. Our model is trained on claim-pages pairs by using attention mechanism in both directions, claim-to-pages and pages-to-claim, which provides complimentary information to each other. We obtain claim-aware sentence representation to predict the correct evidence position and the pages-aware claim representation to detect the relationship between the claim and the pages.

## 2 Related Work

Natural Language Inference (NLI) or Recognizing textual entailment (RTE) detects the relationship between the premise-hypothesis pairs as "entailment", "contradiction" and "not related". With the renaissance of neural network (Krizhevsky et al., 2012; Mikolov et al., 2010; Graves, 2012) and attention mechanism (Xu et al., 2015; Luong et al., 2015; Bahdanau et al., 2014), the popular framework for the RTE is "matching-aggregation" (Parikh et al., 2016; Wang et al., 2017). Under this framework, words of two sentences are firstly aligned, and then the aligning results with original vectors are aggregated into a new representation vector to make the final decision. The attention mechanism can empower this framework to capture more interactive features between two sentences. Compared to Fever task, RTE provides the sentence to verify against instead of having to retrieve it from knowledge source.

Another relative task is question answering (QA) and machine reading comprehension (MRC), for which approaches have recently been extended to handle large-scale resources such as Wikipedia (Chen et al., 2017). Similar to MRC task which needs to identify the answer span in a passage, FEVER task requires to detect the evidence sentences in Wikipedia pages. However, MRC model tends to identify the answer span based on the similarity and reasoning between the question and passage, while similarity-based method is more likely to ignore refuting evidence in pages. For example, a claim stating "Manchester by the Sea is distributed globally" can be refuted by retrieving "It began a limited release on November 18, 2016" as evidence.

## 3 Model

The FEVER dataset is derived from the Wikipedia pages. So, we assume each claim contains at least one entity in Wikipedia and the evidence can be retrieved from these relative pages. Thus, we decompose FEVER task into two components: (1) entity linking which detects Wikipedia entities in claim. We use the pages of identified entities

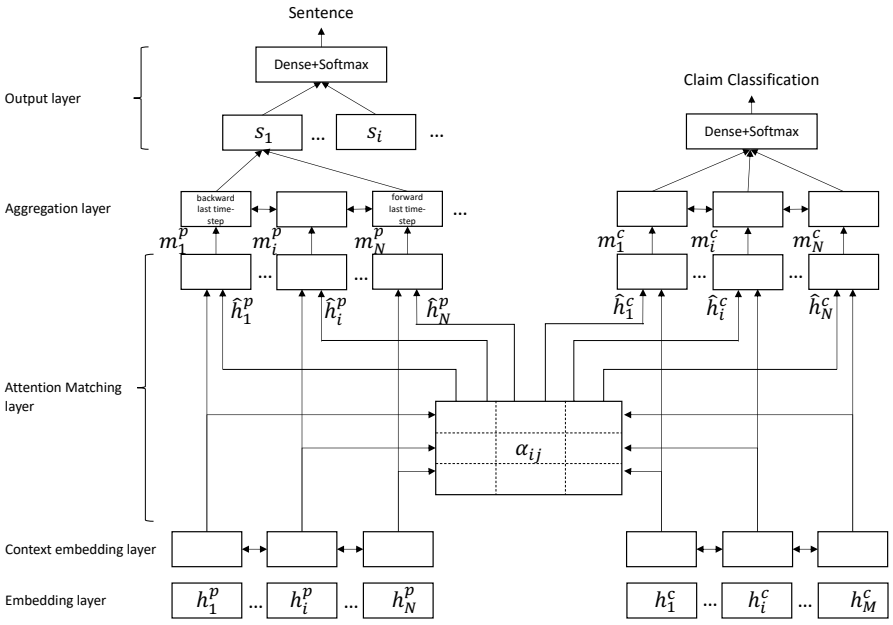


Figure 1: An End-to-End Multi-task Learning Model for Fact Checking

as relative pages. And (2) an end-to-end multi-task learning with bi-direction attention (EMBA) model (in Figure 1) which classify the claim as "supports", "refutes" or "not enough info" with respect to the pages retrieved and select sentences as evidence at the same time.

### 3.1 Entity Liking

S-MART is a Wikipedia entity linking tool for short and noisy text. For each claim, we use S-MART to retrieve the top 5 entities from Wikipedia. These entity pages are jointed together as the source pages then passed to select correct sentences. For a given claim, S-MART first retrieves all possible entities of Wikipedia by surface matching, and then ranks them using a statistical model, which is trained on the frequency counts with which the surface form occurs with the entity.

### 3.2 Sentence Extraction and Claim Verification

We now proceed to identify the correct sentences as evidence from relative pages and try to classify the claim as "supports", "refutes" or "not enough info" with respect to the pages retrieved at the same time. Inspired by the recent success of attention mechanism in NLI (Wang et al., 2017) and MRC (Seo et al., 2016; Tan et al., 2017), we propose an end-to-end multi-task learning with bi-direction attention (EMBA) model, which exploits both pages-to-claim attention to verify the claim and claim-to-pages attention to predict the evidence sentence position respectively. Our model consists of:

**Embedding layer**: This layer represents each word in a fixed-size vector with two components: a word embedding and a character-level embedding. For word embedding, pre-trained word vectors, Glove (Pennington et al., 2014), provides the fixed-size embedding of each word. For character embedding, following Kim (Kim, 2014), characters of each words are embedded into fixed-size embedding, then fed into a Convolutional Neural Network (CNN). The character and word embedding vectors are concatenated together and passed to a Highway Network (Srivastava et al., 2015). The output of this layer are two sequences of word vectors of claim and pages.

**Context embedding layer**: The purpose of this layer is to incorporate contextual information into the presentation of each word of claim and passage. We utilize a bi-directional LSTM (BiLSTM) on the top of the embedding provided by the previous layers to encode contextual embedding for each word.

**Attention matching layer**: In this layer, we compute attention in two directions: from pages to claim as well as from claim to pages. To obtain these attention mechanisms, we first calculate

a shared similarity matrix between the contextual embedding of each word of the claim $\mathbf{h}_i^c$ and each word of the pages $\mathbf{h}_j^p$:

$$\alpha_{ij} = \mathbf{w}[\mathbf{h}_i^c; \mathbf{h}_j^p; \mathbf{h}_i^c \circ \mathbf{h}_j^p] \quad (1)$$

where $\alpha_{ij}$ represents the attention weights on the i-th claim word by j-th pages word, w is a trainable weight vector, $\circ$ is elementwise multiplication, [;] is vector concatenation across row, and implicit multiplication is matrix multiplication.

*Claim-to-pages attention* Claim-to-pages attention represents which claim words are most relevant to each word of pages. To obtain attended pages vector, we take $\alpha_{ij}$ as the weight of $\mathbf{h}_j^p$ and weighted sum all the contextual embedding of pages:

$$\tilde{\mathbf{h}}_j = \sum_{i=1}^{N} \alpha_{ij}\mathbf{h}_i / \sum_{i=1}^{N} \alpha_{ij} \quad (2)$$

Finally, we match each contextual embedding with its corresponding attention vector to obtain the claim-aware representation of each word of pages:

$$\mathbf{m}_j = \mathbf{W}[\mathbf{h}_j; \tilde{\mathbf{h}}_j; \mathbf{h}_j \circ \tilde{\mathbf{h}}_j] \quad (3)$$

*Pages-to-claim attention* Pages-to-claim attention represents which pages words are most relevant to each claim word. Similar to claim-to-pages attention, the attended claim vector and the pages-aware representation of each pages word are calculated by:

$$\tilde{\mathbf{h}}_i = \sum_{j=1}^{N} \alpha_{ij}\mathbf{h}_j / \sum_{j=1}^{N} \alpha_{ij} \quad (4)$$

$$\mathbf{m}_i = \mathbf{W}[\mathbf{h}_i; \tilde{\mathbf{h}}_i; \mathbf{h}_i \circ \tilde{\mathbf{h}}_i] \quad (5)$$

**Aggregation layer**: The input to the aggregation layer is two sequences of matching vectors, the claim-aware pages word representation and pages-aware claim word representation. The goal of the modeling layer is to capture the interaction among the pages words conditioned on the claim as well as the claim words conditioned on the passage words. This is different from the contextual embedding layer, which captures the interaction among context information independent of matching information.

**Sentence selection layer**: The FEVER task requires the model to retrieve sentences of the passage as evidence to verify the claim. The sentence representation $\mathbf{s}_t$ is obtained by concatenating vectors from the last time-step of the previous layer BiLSTM models output sequences. We calculate the probability distribution of the evidence position over the whole pages by:

$$p_t = softmax(\mathbf{w}\mathbf{s}_t) \quad (6)$$

For this sub-task, the objective function is to minimize the negative log probabilities of true evidence index:

$$L_s = -\sum_{t=1}^{N} [y_t \log p_t + (1 - y_t) \log(1 - p_t)] \quad (7)$$

where $y_t \in 0, 1$ denotes a label, $y_t = 1$ means the t-th sentence is a correct evidence, other $y_t = 0$.

**Claim verification layer**: The input of this layer is pages-aware claim representation produced from the matching layer and the output is a 3-way classification, predicting whether the claim is SUPPORTED, REFUTED or NOTENOUGHINFO by the pages. We utilize multiple convolution layers, with the output of 3 for classification. We optimize the objective function:

$$L_c = -\sum_{i=1}^{k} y_i \log \hat{g}_i \quad (8)$$

Where k is the number of claims. $y_t \in 0, 1, 2$ denotes a label, meaning the i-th claim is SUPPORTED, REFUTED, and NOTENOUGHINFO by the pages respectively.

**Training**: The model is trained by minimizing joint objective function:

$$L = L_s + \alpha * L_c \quad (9)$$

where $\alpha$ is the hyper-parameter for weights of two loss functions.

## 4 Experiments

In this section, we evaluate our model on FEVER paper test dataset and shared task test dataset.

### 4.1 Model Details

The model architecture used for this task is depicted in Figure 1. The nonlinearity function $f = \tanh$ is employed. We use 100 1D filters for CNN char embedding, each with a width of 5. The hidden state size (d) of the model is 100. We use the Adam (Kingma and Ba, 2014) optimizer, with a minibatch size of 32 and an initial learning rate of 0.001. A dropout rate of 0.2 is used for the

| | EMBA (paper test) | Baseline (paper dev) |
|---|---|---|
| **Evidence Score** | 30.34 | 32.57 |
| **Label Accuracy** | 45.06 | - |
| **Evidence Precision** | 46.12 | - |
| **Evidence Recall** | 42.84 | - |
| **Evidence F1** | 44.42 | - |

Table 1: our EMBA model results on the FEVER paper test dataset, Baseline method results on the paper dev dataset.

| Evidence F1 | Label Accuracy | FEVER Score |
|---|---|---|
| 39.73 | 45.38 | 29.22 |

Table 2: Model results on the FEVER shared task test dataset.

CNN, all LSTM layers, and the linear transformation. The parameters are initialized by the techniques described in (Glorot, 2010). The max value used for max-norm regularization is 5. The $L_c$ loss weight is set to $\alpha = 0.5$.

### 4.2 Experimental Results

We use the official evaluation script[6] to compute the evidence F1 score, label accuracy and FEVER score. As shown in Table 1, our method achieves comparable performance on FEVER paper test dataset comparing with the baseline method on FEVER paper dev dataset. The result shows that jointly verifying a claim and retrieving evidences at same time can be as good as pipelined model. Our method results on the FEVER paper shared task test dataset is showed in Table 2. Besides, We calculate and present the confusion matrix of claim classification results on the FEVER paper test dataset in Table 3. Our model isn't good at identifying the unrelated relationship between claim and pages retrieved. Our model sentence selection performance is recorded in Table 4. We can see that our model doesn't perform well for retrieving evidence. Though with low evidence precision, our model average accuracy without requirement to provide correct evidence (51.97%) is similar to 52.09% accuracy of baseline method, which means that claim verification module and the sentence extraction module are relatively independent in our model.

[6]https://github.com/sheffieldnlp/fever-baselines

| | NEI | REFUTES | SUPPORTS | recall |
|---|---|---|---|---|
| **NEI** | 1285 | 1432 | 390 | 41% |
| **REFUTES** | 992 | 1937 | 155 | 62.8% |
| **SUPPORTS** | 942 | 860 | 1284 | 41.6% |
| **precision** | 39.9% | 45.8% | 70.2% | |

Table 3: confusion matrix for claim classification. (NEI = "not enough info")

| | Evidence precision | Evidence recall | Evidence F1 |
|---|---|---|---|
| **SUPPORTS** | 22.07% | 40.08% | 28.47% |
| **REFUTES** | 23.8% | 45.18% | 31.18% |

Table 4: sentences retrieval performance.

### 4.3 Error Analysis

We investigate the predicted results on the paper test dataset and show several error causes as followings.

**Document retrieval** We use entity linking tool to retrieve relative Wikipedia pages. Some entity mentions in claims are linked incorrectly, hence we cannot obtain the desired pages containing the correct evidence sentences. The S-MART tool returned correct entities for 70% claims of paper test dataset. A better entity retrieval method should be researched for the FEVER task.

**Pages length** After document retrieval, the relative pages are concatenated and passed through EMBA model. However, in order to train and predict effectively, the length of the pages is limited to 800 tokens. So, if there are many relative pages and the position of the evidence sentence is near the end of the page, these correct sentences would be cut off.

**Evidence composition** Some claims require composition of evidences from multiple pages. Furthermore, the selection of second page relies on the correct retrieval of the first page and sentence. For example, claim "Deepika Padukone has been in at least one Indian films" can be supported by combination of "She starred roles in Yeh Jawaani Hai Deewani" and "Yeh Jawaani Hai Deewani is an Indian film" from "Deepika Padukone" and "Yeh Jawaani Hai Deewani" Wikipedia pages respectively. The second page couldn't be found correctly if we don't select the first sentence exactly. 18% claims in train dataset belong to this situation.

## 5 Conclusion

We propose a novel end-to-end multi-task learning with bi-direction attention (EMBA) model to detect sentences as evidence and classify the claim as "supports", "refutes" or "not enough info" with respect to the pages retrieved at the same time. EMBA uses attention mechanism in both directions to capture interactive features between claim and pages retrieved. Model obtains claim-aware sentence representation to predict the correct evidence position and the pages-aware claim representation to detect the relationship between the claim and the pages. Experimental results on the FEVER paper test dataset show that our approach achieve comparable performance comparing with the baseline method. There are several promising directions that worth researching in the future. For instance, in sentence selection layer, the model just predicts whether a sentence is an evidence. Further, we can try to instantly predict whether a sentence is "supporting", "refuting" or "not related with" the claim. What's more, the hyperparameter $\alpha$ for joint loss function is fixed. A good value for this parameter can achieve one plus one is greater than two. We can try to learn this parameter value during training the model.

## 6 acknowledgement

This work is supported by National Natural Science Foundation of China (Grant No. 61501048), Beijing Natural Science Foundation (Grant No. 4182042), Fundamental Research Funds for the Central Universities (No. 2017RC12), and National Natural Science Foundation of China (U1536111, U1536112).

## References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *Computer Science*.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*.

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*, pages 1163–1168.

Yoshua Bengio Glorot, Xavier. 2010. Understanding the difficulty of training deep feedforward neural networks. In *international conference on artificial intelligence and statistics*, pages 249–256.

Alex Graves. 2012. *Long Short-Term Memory*. Springer Berlin Heidelberg.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *Computer Science*.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *International Conference on Neural Information Processing Systems*, pages 1097–1105.

Minh Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *Computer Science*.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *INTERSPEECH 2010, Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September*, pages 1045–1048.

Ankur P Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. *arXiv preprint arXiv:1606.01933*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Dean Pomerleau and Delip Rao. 2017. *Fake News Challenge http://fakenewschallenge.org*.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Highway networks. *arXiv preprint arXiv:1505.00387*.

Chuanqi Tan, Furu Wei, Nan Yang, Bowen Du, Weifeng Lv, and Ming Zhou. 2017. S-net: From answer extraction to answer generation for machine reading comprehension. *arXiv preprint arXiv:1706.04815*.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*.

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. pages 422–426.

Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral multi-perspective matching for natural language sentences.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. *Computer Science*, pages 2048–2057.