Hierarchical Multi-head Attentive Network for Evidence-aware Fake News Detection

Nguyen Vo
Worcester Polytechnic Institute
Computer Science Department
Worcester, MA, USA, 01609
nkvo@wpi.edu

Kyumin Lee
Worcester Polytechnic Institute
Computer Science Department
Worcester, MA, USA, 01609
kmlee@wpi.edu

Abstract

The widespread of fake news and misinformation in various domains ranging from politics, economics to public health has posed an urgent need to automatically fact-check information. A recent trend in fake news detection is to utilize evidence from external sources. However, existing evidence-aware fake news detection methods focused on either only word-level attention or evidence-level attention, which may result in suboptimal performance. In this paper, we propose a Hierarchical Multi-head Attentive Network to fact-check textual claims. Our model jointly combines multi-head word-level attention and multi-head document-level attention, which aid explanation in both word-level and evidence-level. Experiments on two real-word datasets show that our model outperforms seven state-of-the-art baselines. Improvements over baselines are from 6% to 18%. Our source code and datasets are released at https://github.com/nguyenvo09/EACL2021.

1 Introduction

The proliferation of biased news, misleading claims, disinformation and fake news has caused heightened negative effects on modern society in various domains ranging from politics, economics to public health. A recent study showed that maliciously fabricated and partisan stories possibly caused citizens' misperception about political candidates (Allcott and Gentzkow, 2017) during the 2016 U.S. presidential elections. In economics, the spread of fake news has manipulated stock price (Kogan et al., 2019). For example, $139 billion was wiped out when the Associated Press (AP)'s hacked Twitter account posted rumor about White House explosion with Barack Obama's injury. Recently, misinformation has caused infodemics in public health (Ashoka, 2020) and even led to people's fatalities in the physical world (Alluri, 2019).

To reduce the spread of misinformation and its detrimental influences, many fact-checking systems have been developed to fact-check textual claims. It is estimated that the number of fact-checking outlets has increased 400% in 60 countries since 2014 (Stencel, 2019). Several fact-checking systems such as *snopes.com* and *politifact.com* are widely used by both online users and major corporations. Facebook (CNN, 2020) recently incorporated third-party fact-checking sites to social media posts and Google integrated fact-checking articles to their search engine (Wang et al., 2018). These fact-checking systems debunk claims by manually assess their credibility based on collected webpages used as evidence. However, this manual process is laborious and unscalable to handle the large volume of produced false claims on communication platforms. Therefore, in this paper, our goal is to build an automatic fake news detection system to fact-check textual claims based on collected evidence to speed up fact-checking process of the above fact-checking sites.

To detect fake news, researchers proposed to use linguistics and textual content (Castillo et al., 2011; Zhao et al., 2015; Liu et al., 2015). Since textual claims are usually deliberately written to deceive readers, it is hard to detect fake news by solely relying on the content claims. Therefore, multiple works utilized other signals such as temporal spreading patterns (Liu and Wu, 2018), network structures (Wu and Liu, 2018; Vo and Lee, 2018; Shu et al., 2020) and users' feedbacks (Vo and Lee, 2019; Shu et al., 2019; Vo and Lee, 2020a). However, limited work used external webpages as documents which could provide interpretive explanation to users. Several recent work (Popat et al., 2018; Ma et al., 2019; Vo and Lee, 2020b) started to utilize documents to fact-check textual claims. Popat et al. (2018) used word-level attention in documents but treated all documents with equal im-

Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, pages 965–975
April 19 - 23, 2021.

portance whereas Ma et al. (2019) only focused on which documents are more crucial without considering what words help explain credibility of textual claims.

Observing drawbacks of the existing work, we propose Hierarchical Multi-head Attentive Network which jointly utilizes word attention and evidence attention. Overall semantics of a document may be generated by multiple parts of the document. Therefore, we propose a multi-head word attention mechanism to capture different semantic contributions of words to the meaning of the documents. Since a document may have different semantic aspects corresponding to various information related to credibility of a claim, we propose a multi-head document-level attention mechanism to capture contributions of the different semantic aspects of the documents. In our attention mechanism, we also use speakers and publishers information to further improve effectiveness of our model. To our knowledge, our work is the first applying multi-head attention mechanism for both words and documents in evidence-aware fake news detection. Our work makes the following contributions:

- We propose a novel hierarchical multi-head attention network which jointly combines word attention and evidence attention for evidence-aware fake news detection.
- We propose a novel multi-head attention mechanism to capture important words and evidence.
- Experiments on two public datasets demonstrate the effectiveness and generality of our model over state-of-the-art fake news detection techniques.

2 Related Work

Many methods have been proposed to detect fake news in recent years. These methods can be placed into three groups: (1) human-based fact-checking sites (e.g. Snopes.com, Politifact.com), (2) machine learning based methods and (3) hybrid systems (e.g. content moderation on social media sites). In machine-learning-based methods, researchers mainly used linguistics and textual content (Zellers et al., 2019; Zhao et al., 2015; Wang, 2017; Shu et al., 2019), temporal spreading patterns (Liu and Wu, 2018), network structures (Wu and Liu, 2018; Vo and Lee, 2018; You et al., 2019), users' feedbacks (Vo and Lee, 2019; Shu et al., 2019) and multimodal signals (Gupta et al., 2013; Vo and Lee, 2020b). Recently, researchers focus on fact-checking claims based on evidence from different sources. Thorne and Vlachos (2017) and Vlachos and Riedel (2015) fact-check claims using subject-predicate-object triplets extracted from knowledge graph as evidence. Chen et al. (2020) assess claims' credibility using tabular data. Our work is closely related to fact verification task (Thorne et al., 2018; Nie et al., 2019; Soleimani et al., 2020) which aims to classify a pair of a claim and an evidence extracted from Wikipedia into three classes: *supported*, *refuted*, or *not enough info*. For fact verification task, Nie et al. (2019) used ELMo (Peters et al., 2018) to extract contextual embeddings of words and used a modified ESIM model (Chen et al., 2017). Soleimani et al. (2020) used BERT model (Devlin et al., 2018) to retrieve and verify claims. Zhou et al. (2019) used graph based models for semantic reasoning. Our work is different from these work since our goal is to classify a pair of a claim and a list of relevant evidence into *true* or *false*.

Our work is close to existing work about evidence-aware fake news detection (Popat et al., 2018; Ma et al., 2019; Wu et al., 2020; Mishra and Setty, 2019). Popat et al. (2018) used an average pooling layer to derive claims' representation to attend to words in evidence, Mishra and Setty (2019) focused on words and sentences in each evidence, and Ma et al. (2019) proposed a semantic entailment model to attend to important evidence. However, to the best of our knowledge, our work is the first jointly using multi-head attention mechanisms to focus on important words in each evidence and important evidence from a set of relevant articles. Our attention mechanism is different from these work since we use multiple attention heads to capture different semantic contributions of words and evidence.

3 Problem Statement

We denote an evidence-based fact-checking dataset $\mathcal{C}$ as a collection of tuples $(c, s, \mathcal{D}, \mathcal{P})$ where c is a textual claim originated from a speaker s, $\mathcal{D} = \{d_i\}_{i=1}^{k}$ is a collection of k documents[1] relevant to the claim c and $\mathcal{P} = \{p_i\}_{i=1}^{k}$ is the corresponding publishers of documents in $\mathcal{D}$. Note, $|\mathcal{D}| = |\mathcal{P}|$. Our goal is to classify each tuple $(c, s, \mathcal{D}, \mathcal{P})$ into a pre-defined class (i.e. true news/fake news).

[1] We use the term "documents", "articles", and "evidence" interchangeably.

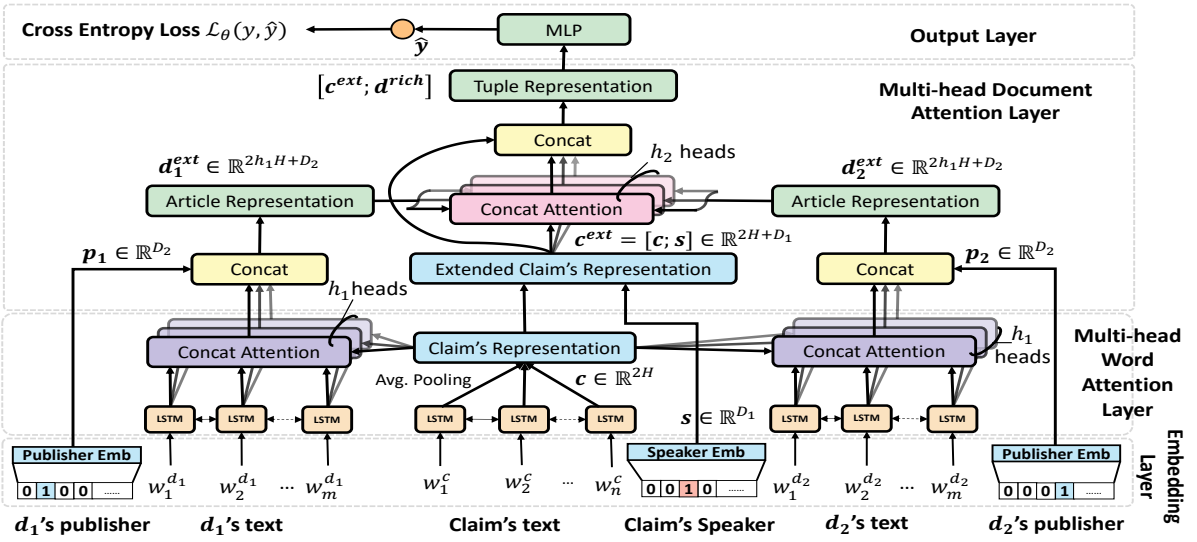


Figure 1: The architecture of our proposed model **MAC** in which we show a claim c, two associated relevant articles d_1 and d_2 and sources of the claim and the two documents. h_1 and h_2 are the number of heads of word-level attention and document-level attention respectively.

4 Framework

In this section, we describe our Hierarchical Multi-head Attentive Network for Fact-Checking (**MAC**) which jointly considers word-level attention and document-level attention. Our framework consists of four main components: (1) embedding layer, (2) multi-head word attention layer, (3) multi-head document attention layer and (4) output layer. These components are illustrated in Fig. 1 where we show a claim and two documents as an example.

4.1 Embedding Layer

Each claim c is modeled as a sequence of n words $[w_1^c, w_2^c, ..., w_n^c]$ and d_i is viewed as another sequence of m words $[w_1^d, w_2^d, ..., w_m^d]$. Each word w_i^c and w_j^d will be projected into D-dimensional vectors $\mathbf{e}_i^c$ and $\mathbf{e}_j^d$ respectively by an embedding matrix $\mathbf{W}_e \in \mathbb{R}^{V \times D}$ where V is the vocabulary size. Each speaker s and publisher p_i modeled as one-hot vectors are transformed into dense vectors $\mathbf{s} \in \mathbb{R}^{D_1}$ and $\mathbf{p}_i \in \mathbb{R}^{D_2}$ respectively by using two matrices $\mathbf{W}_s \in \mathbb{R}^{S \times D_1}$ and $\mathbf{W}_p \in \mathbb{R}^{P \times D_2}$, where S and P are the number of speakers and publishers in a training set respectively. Both $\mathbf{W}_s$ and $\mathbf{W}_p$ are uniformly initialized in $[-0.2, 0.2]$. Note that, both matrices $\mathbf{W}_s$ and $\mathbf{W}_p$ are jointly learned with other parameters of our MAC.

4.2 Multi-head Word Attention Layer

We input word embeddings $\mathbf{e}_i^c$ of the claim c into a bidirectional LSTM (Graves et al., 2005) which helps generate contextual representation $\mathbf{h}_i$ of each token as follows: $\mathbf{h}_i^c = [\overleftarrow{\mathbf{h}}_i; \overrightarrow{\mathbf{h}}_i] \in \mathbb{R}^{2H}$, where $\overleftarrow{\mathbf{h}}_i$ and $\overrightarrow{\mathbf{h}}_i$ are hidden states in forward and backward pass of the BiLSTM, symbol ; means concatenation and H is hidden size. We derive claim's representation in $\mathbb{R}^{2H}$ by an average pooling layer as follows:

$$\mathbf{c} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{h}_i^c \quad (1)$$

Applying a similar process on the top of each document d_i with a different BiLSTM, we have contextual representation $\mathbf{h}_j^d \in \mathbb{R}^{2H}$ for each word in d_i. After going through BiLSTM, d_i is modeled as matrix $\mathbf{H} = [\mathbf{h}_1^d \oplus \mathbf{h}_2^d \oplus ... \oplus \mathbf{h}_m^d] \in \mathbb{R}^{m \times 2H}$ where $\oplus$ denotes stacking.

To understand what information in a document helps us fact-check a claim, we need to guide our model to focus on crucial keywords or phrases of the document. Drawing inspiration from (Luong et al., 2015), we firstly replicate vector $\mathbf{c}$ (Eq.1) m times to create matrix $\mathbf{C}_1 \in \mathbb{R}^{m \times 2H}$ and propose an attention mechanism to attend to important words in the document d_i as follows:

$$\mathbf{a}_1 = softmax\big(\tanh\big([\mathbf{H}; \mathbf{C}_1] \cdot \mathbf{W}_1\big) \cdot \mathbf{w}_2\big) \quad (2)$$

where $\mathbf{w}_2 \in \mathbb{R}^{a_1}$, $\mathbf{W}_1 \in \mathbb{R}^{4H \times a_1}$, $[\mathbf{H}; \mathbf{C}_1]$ is concatenation of two matrices on the last dimension and $\mathbf{a}_1 \in \mathbb{R}^m$ is attention distribution on m words. However, the overall semantics of the document might be generated by multiple parts of the document (Lin et al., 2017). Therefore, we propose a

multi-head word attention mechanism to capture different semantic contributions of words by extending vector $\mathbf{w}_2$ into a matrix $\mathbf{W}_2 \in \mathbb{R}^{a_1 \times h_1}$ where h_1 is the number of attention heads shown in Fig. 1. We modify Eq. 2 as follows:

$$\mathbf{A}_1 = softmax_{col}\big(\tanh([\mathbf{H}; \mathbf{C}_1] \cdot \mathbf{W}_1) \cdot \mathbf{W}_2\big) \quad (3)$$

where $\mathbf{A}_1 \in \mathbb{R}^{m \times h_1}$ and each column of $\mathbf{A}_1$ has been normalized by the softmax operation. Intuitively, $\mathbf{A}_1$ stands for h_1 different attention distributions on top of m words of the document d_i, helping us capture different aspects of the document. After computing $\mathbf{A}_1$, we derive representation of document d_i as follows:

$$\mathbf{d}_i = flatten(\mathbf{A}_1^T \cdot \mathbf{H}) \quad (4)$$

where $\mathbf{d}_i \in \mathbb{R}^{h_1 2H}$ and function *flatten(.)* flattens $\mathbf{A}_1^T \cdot \mathbf{H}$ into a vector. We also implemented a more sophisticated multi-head attention in (Vaswani et al., 2017) but did not achieve good results.

4.3 Multi-head Document Attention Layer

This layer consists of three components as follows: (1) extending representations of claims, (2) extending representations of evidence and (3) multi-head document attention mechanism.

Extending representations of claims. So far the representation of the claim $\mathbf{c}$ (Eq. 1) is only from textual content. In reality, a speaker who made a claim may impact credibility of the claim. For example, claims from some politicians are controversial and inaccurate (Allcott and Gentzkow, 2017). Therefore, we enrich vector $\mathbf{c}$ by concatenating it with speaker's embedding $\mathbf{s}$ to generate $\mathbf{c}^{ext} \in \mathbb{R}^x$, where $x = 2H + D_1$ as shown in Eq. 5.

$$\mathbf{c}^{ext} = [\mathbf{c}; \mathbf{s}] \in \mathbb{R}^x \quad (5)$$

Extending representations of evidence. Intuitively, an article published by *nytimes.com* might be more reliable than a piece of news published by *breitbart.com* which is known to be a less credible site. Therefore, to capture more information, we further enrich representations of evidence with publishers' information by concatenating $\mathbf{d}_i$ (Eq. 4) with its publisher's embedding $\mathbf{p}_i$ as follows:

$$\mathbf{d}_i^{ext} = [\mathbf{d}_i; \mathbf{p}_i] \in \mathbb{R}^y \quad (6)$$

where $y = 2h_1H + D_2$. From Eq. 6, we can generate representations of k relevant articles and stack them as shown in Eq. 7.

$$\mathbf{D} = [\mathbf{d}_1^{ext} \oplus ... \oplus \mathbf{d}_k^{ext}] \in \mathbb{R}^{k \times y} \quad (7)$$

Multi-head Document Attention Mechanism. In real life, a journalist from *snopes.com* and *politifact.com* may use all k articles relevant to the claim c to fact-check it but she may focus on some key articles to determine the verdict of the claim c while other articles may have negligible information. To capture such intuition, we need to downgrade uninformative documents and concentrate on more meaningful articles. Similar to Section 4.2, we use multi-head attention mechanism which produces different attention distributions representing diverse contributions of articles toward determining veracity of the claim c.

We firstly create matrix $\mathbf{C}_2 \in \mathbb{R}^{k \times x}$ by replicating vector $\mathbf{c}^{ext}$ (Eq. 5) k times. Secondly, the matrix $\mathbf{C}_2$ is concatenated with matrix $\mathbf{D}$ (Eq. 7) on the last dimension of the two matrices denoted as $[\mathbf{D}; \mathbf{C}_2] \in \mathbb{R}^{k \times (x+y)}$.

Our proposed multi-head document-level attention mechanism applies h_2 different attention heads as shown in Eq. 8.

$$\mathbf{A}_2 = softmax_{col}(\tanh([\mathbf{D}; \mathbf{C}_2] \cdot \mathbf{W}_3) \cdot \mathbf{W}_4) \quad (8)$$

where $\mathbf{W}_3 \in \mathbb{R}^{(x+y) \times a_2}$, $\mathbf{W}_4 \in \mathbb{R}^{a_2 \times h_2}$. The matrix $\mathbf{A}_2 \in \mathbb{R}^{k \times h_2}$, where each of its column is normalized by the softmax operator, is a collection of h_2 different attention distributions on k documents. Using attention weights, we can generate attended representation of k evidence denoted as $\mathbf{d}^{rich} \in \mathbb{R}^{h_2 y}$ as shown in Eq. 9.

$$\mathbf{d}^{rich} = flatten(\mathbf{A}_2^T \cdot \mathbf{D}) \quad (9)$$

where *flatten(.)* function flattens $\mathbf{A}_2^T \cdot \mathbf{D}$ into a vector. We finally generate representation of a tuple $(c, s, \mathcal{D}, \mathcal{P})$ by concatenating vector $\mathbf{c}^{ext}$ (Eq. 5) and vector $\mathbf{d}^{rich}$ (Eq. 9), denoted as $[\mathbf{c}^{ext}; \mathbf{d}^{rich}]$.

To the best of our knowledge, our work is the first work utilizing multi-head attention mechanism integrated with speakers and publishers information to capture various semantic contributions of evidence toward fact-checking process.

4.4 Output Layer

In this layer, we input tuple representation $[\mathbf{c}^{ext}; \mathbf{d}^{rich}]$ into a multilayer perceptron (MLP) to compute probability $\hat{y}$ that the claim c is a true news as follows:

$$\hat{y} = \sigma\big(\mathbf{W}_6 \cdot \big(\mathbf{W}_5 \cdot [\mathbf{c}^{ext}; \mathbf{d}^{rich}] + \mathbf{b}_5\big) + \mathbf{b}_6\big) \quad (10)$$

where $\mathbf{W}_5, \mathbf{W}_6, \mathbf{b}_5, \mathbf{b}_6$ are weights and biases of the MLP, and $\sigma(.)$ is the sigmoid function. We optimize our model by minimizing the standard

Table 1: Statistics of our experimental datasets

	Snopes	PolitiFact
True claims	1,164	1,867
False claims	3,177	1,701
|Speakers|	N/A	664
|Documents|	29,242	29,556
|Publishers|	12,236	4,542

cross-entropy as shown on the top of Fig. 1.

$$\mathcal{L}_\theta(y, \hat{y}) = -\big(y \log \hat{y} + (1-y) \log(1-\hat{y})\big) \quad (11)$$

where $y \in \{0, 1\}$ is the ground truth label of a tuple $(c, s, \mathcal{D}, \mathcal{P})$. During training, we sample a mini batch of 32 tuples and compute average loss from the tuples.

5 Experiments

5.1 Datasets

We employed two public datasets released by (Popat et al., 2018). Each of these datasets is a collections of tuples $(c, s, \mathcal{D}, \mathcal{P}, y)$ where each textual claim c and its credible label y are collected from two major fact-checking websites `snopes.com` and `politifact.com`. The articles pertinent to the claim c are retrieved by using search engines. Each Snopes claim was labeled as *true* or *false* while in Politifact, there were originally six labels: *true*, *mostly true*, *half true*, *false*, *mostly false*, *pants on fire*. Following (Popat et al., 2018), we merge *true*, *mostly true* and *half true* into *true claims* and the rest are into *false claims*. Details of our datasets are presented in Table 1. Note that Snopes does not have speakers' information.

5.2 Baselines

We compare our MAC model with seven state-of-the-art baselines divided into two groups. The first group of the baselines only used textual content of claims, and the second group of the baselines utilized relevant articles to fact-check textual claims. A related method (Mishra and Setty, 2019) used subject information of articles (e.g. politics, entertainment), which was not available in our datasets. We tried to compare with it but achieved poor results perhaps due to missing information. Therefore, we do not report its result in this paper. Details of the baselines are shown as follows:

Using only claims' text:

- **BERT** (Devlin et al., 2018) is a pre-trained language model achieving state-of-the-art results on many NLP tasks. The representation of [CLS] token is inputted to a trainable linear layer to classify claims.
- **LSTM-Last** is a model proposed in (Rashkin et al., 2017). **LSTM-Last** takes the last hidden state of the LSTM as representations of claims. These representations will be inputted to a linear layer for classification.
- **LSTM-Avg** is another model proposed in (Rashkin et al., 2017) which used an average pooling layer on top of hidden states to derive representations of claims.
- **CNN** (Wang, 2017) is a state-of-the-art model which applied 1D-convolutional neural network on word vectors of claims.

Using both claims' text and articles' text:

- **DeClare** (Popat et al., 2018) computes credibility score of each pair of a claim c and a document d_i. The overall credible rating is averaged from all k relevant articles.
- **HAN** (Ma et al., 2019) is a hierarchical attention network based on representations of relevant documents. It uses attention mechanisms to determine which document is more important without considering which word in a document should be focused on.
- **NSMN** (Nie et al., 2019) is a state-of-the-art model designed to determine stance of a document d_i with respect to claim c. We apply NSMN on our dataset by predicting score of each pair (c, d_i) and computing average score based on documents in $\mathcal{D}$ same as DeClare.

Note that, we also applied BERT, LSTM-Last, LSTM-Avg and CNN by using both claims' text and articles' text. For each of these baselines, we concatenated a claim's text and a document's text, and input the concatenated content into the baseline to compute likelihood that the claim is fake news. We computed average probability based on all documents of the claim and used it as final prediction. However, we did not observe considerable improvements of these baselines. In addition to deep-learning-based baselines, we compared our MAC with other feature-based techniques (e.g. SVM). As expected, these traditional techniques had inferior performance compared with neural models. Therefore, we only report the seven baselines' performance.

Table 2: Performance of MAC and baselines on Snopes dataset. MAC outperforms baselines significantly with p-value<0.05 by one-sided paired Wilcoxon test.

Method Types	Methods				True News as Positive			Fake News as Positive		
		AUC	F1 Macro	F1 Micro	F1	Precision	Recall	F1	Precision	Recall
Using only claims' text	BERT	0.60852	0.56096	0.69806	0.31574	0.40318	0.26050	0.80618	0.76011	0.85839
	LSTM-Avg	0.69124	0.62100	0.71877	0.42953	0.48415	0.39692	0.81246	0.79139	0.83671
	LSTM-Last	0.70142	0.63122	0.72415	0.44650	0.48935	0.41412	0.81594	0.79594	0.83776
	TextCNN	0.70537	0.63081	0.72005	0.45001	0.48164	0.43035	0.81160	0.79882	0.82622
Using both claims' text & articles' text	HAN	0.70365	0.62510	0.72800	0.42884	0.49192	0.38161	0.82136	0.79058	0.85490
	NSMN	0.77270	0.68006	0.76127	0.51954	0.57558	0.48182	0.84058	0.82011	0.86364
	DeClare	0.81036	0.72445	0.78813	0.59250	0.61235	0.58096	0.85640	0.85023	0.86399
Ours	MAC	**0.88715**	**0.78660**	**0.83316**	**0.68738**	**0.69975**	**0.68601**	**0.88581**	**0.88617**	**0.88706**
Imprv. over the best baseline		9.47%	8.58%	5.71%	16.01%	14.27%	18.08%	3.43%	4.23%	2.67%

5.3 Experimental Settings

For each dataset, we randomly select 10% number of claims from each class to form a validation set, which is used for tuning hyper-parameters. We report 5-fold stratified cross validation results on the remaining 90% of the data. We train our model and baselines on 4-folds and test them on the remaining fold. We use AUC, macro/micro F1, class-specific F1, Precision and Recall as evaluation metrics. To mitigate overfitting and reduce training time, we early stop training process on the validation set when F1 macro on the validation data continuously decreases in 10 epochs. When we get the same F1 macro between consecutive epochs, we rely on AUC for early stopping.

For fair comparisons, we use Adam optimizer (Kingma and Ba, 2014) with learning rate 0.001 and regularize parameters of all methods with ℓ_2 norm and weight decay $\lambda = 0.001$. As the maximum lengths of claims and articles in words are 30 and 100 respectively for both datasets, we set $n = 30$ and $m = 100$. For HAN and our model, we set $k = 30$ since the number of articles for each claim is at most 30 in both datasets. Batch size is set to 32 and we trained all models until convergence. We tune all models including ours with hidden size H chosen from $\{64, 128, 300\}$, pre-trained word-embeddings are from Glove (Pennington et al., 2014) with $D = 300$. Both D_1 and D_2 are tuned from $\{128, 256\}$. The number of attention heads h_1 and h_2 is chosen from $\{1, 2, 3, 4, 5\}$, a_1 and a_2 are equal to $2 \times H$. In addition to Glove, we also utilized contextual embeddings from pre-trained language models such as ELMo and BERT but achieved comparable performances. We implemented all methods in PyTorch 0.4.1 and run experiments on an NVIDIA GTX 1080.

5.4 Performance of MAC and baselines

We show experimental results of our model and baselines in Tables 2 and 3. In Table 2, MAC outperforms all baselines with significance level $p < 0.05$ by using one-sided paired Wilcoxon test on Snopes dataset. MAC achieves the best result when $h_1 = 5, h_2 = 2, H = 300$ and $D_1 = D_2 = 128$. In Table 3, MAC also significantly outperforms all baselines with $p < 0.05$ according to one-sided paired Wilcoxon test on PolitiFact dataset. The hyperparameters we selected for MAC are $h_1 = 3, h_2 = 1, H = 300$ and $D_1 = D_2 = 128$.

For baselines, BERT is used as a static encoder. We tried to fine tune it but even achieve worse results. This might be because we do not have sufficient data to tune it. For both HAN and DeClare, since both papers do not release their source code, we tried our best to reproduce results from these two models. HAN model derived representation of each document by using the last hidden state of a GRU (Chung et al., 2014) without any attention mechanism on words to downgrade unimportant words (e.g. stop words), leading to poor representations of documents. Therefore, document-level attention mechanism in HAN model did not perform well. Similar patterns can be observed in two baselines LSTM-Avg and LSTM-Last. DeClare performed best among baselines, indicating the importance of applying word-level attention on words to reduce impact of less informative words.

We can see that our MAC outperforms all baselines in all metrics. When viewing *true news* as *positive class*, our MAC has an average increase of 16.0% and 7.1% over the best baselines on Snopes and PolitiFact respectively. We also have an increase of 4.7% improvements over baselines with a maximum improvements of 10.1% in PolitiFact

Table 3: Performance of MAC and baselines on PolitiFact dataset. MAC outperforms baselines with statistical significance level p-value<0.05 by one-sided paired Wilcoxon test.

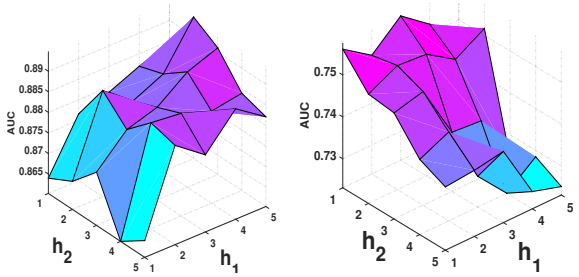
Method Types	Methods				True News as Positive			Fake News as Positive		
		AUC	F1 Macro	F1 Micro	F1	Precision	Recall	F1	Precision	Recall
Using only claims' text	BERT	0.58822	0.56021	0.56446	0.56364	0.59206	0.54968	0.55678	0.54354	0.58069
	LSTM-Avg	0.65465	0.60564	0.60866	0.61821	0.63192	0.61267	0.59307	0.59046	0.60425
	LSTM-Last	0.64289	0.60196	0.60493	0.61703	0.62634	0.61456	0.58690	0.58763	0.59434
	TextCNN	0.65152	0.60380	0.60740	0.61521	0.63010	0.61030	0.59238	0.59049	0.60421
Using both claims' text & articles' text	HAN	0.63201	0.58655	0.59121	0.59193	0.61502	0.58290	0.58117	0.57573	0.60034
	NSMN	0.64237	0.60211	0.60431	0.61123	0.63051	0.59912	0.59299	0.58213	0.60999
	DeClare	0.70642	0.65213	0.65350	0.67230	0.66548	0.67997	0.63195	0.64053	0.62444
Ours	MAC	**0.75756**	**0.68642**	**0.69116**	**0.71786**	**0.68856**	**0.75493**	**0.65498**	**0.70546**	**0.62576**
Imprv. over the best baseline		7.24%	5.26%	5.76%	6.78%	3.47%	11.02%	3.64%	10.14%	0.21%

Table 4: Impact of word attention and evidence attention on our MAC in two datasets

Methods	Snopes		PolitiFact	
	AUC	F1 Macro	AUC	F1 Macro
Only Word Att	0.87278	0.77831	0.74483	0.67818
Only Evidence Att	0.82531	0.72885	0.71790	0.65187
Word & Doc Att	**0.88715**	**0.78660**	**0.75756**	**0.68642**

Table 5: Impact of speakers and publishers on performance of MAC in two datasets

Methods	Snopes		PolitiFact	
	AUC	F1 Macro	AUC	F1 Macro
Text Only	0.88186	0.77146	0.72401	0.66844
Text + Publishers	**0.88715**	**0.78660**	0.72645	0.66984
Text + Speakers			0.75202	0.68483
Text + Pubs + Spkrs			**0.75756**	**0.68642**



(a) Snopes (b) PolitiFact

Figure 2: Sensitivity of MAC with respect to number of heads in word-level attention h_1 and the number of heads in document-level attention h_2

when considering *fake news* as *negative class*. In terms of AUC, average improvements of MAC over the baselines are 7.9% and 6.1% on Snopes and PolitiFact respectively. Improvements of MAC over baselines can be explained by our multi-head attention mechanism shown in Eq. 3 and Eq. 8. After attending to words in documents, we can generate better representations of documents/evidence, leading to more effective document-level attention compared with HAN model.

5.5 Ablation Studies

Impact of Word Attention and Evidence Attention. We study the impact of attention layers on performance of MAC by (1) using only word attention and replacing evidence attention with an average pooling layer on top of documents' representations and (2) using only evidence attention and replacing word attention with an average pooling layer on top of words' representations. As we can see in Table 4, using only word attention performs much better than using only evidence attention. This is because without downgrading less informative words in evidence, irrelevant information can be captured, leading to low quality representations of evidence. This experiment aligns with our observation that HAN model, which used only evidence attention, did not perform well. When combining both attention mechanisms hierarchically, we consistently achieve best results on two datasets in Table 4. In particular, the model *Word & Doc Att* outperformed both *Only Evidence Att* and *Only Evidence Att* significantly with p-value < 0.05. This result indicates that it is crucial to combine word-level attention and document-level attention to improve the performance of evidence-aware fake news detection task.

Impact of Speakers and Publishers on MAC. To study how speakers and publishers impact performance of MAC, we experiment four models: (1) using text only (Text Only), (2) using text and publishers (Text + Publishers), (3) using text and speakers (Text + Speakers) and (4) using text, publishers and speakers (Text + Pubs + Spkrs). In Table 5, Text + Publishers has better performance then using only text in both datasets. In PolitiFact, Text + Speakers achieves 2∼3% improvements over Text + Publishers, indicating that speakers who made claims are

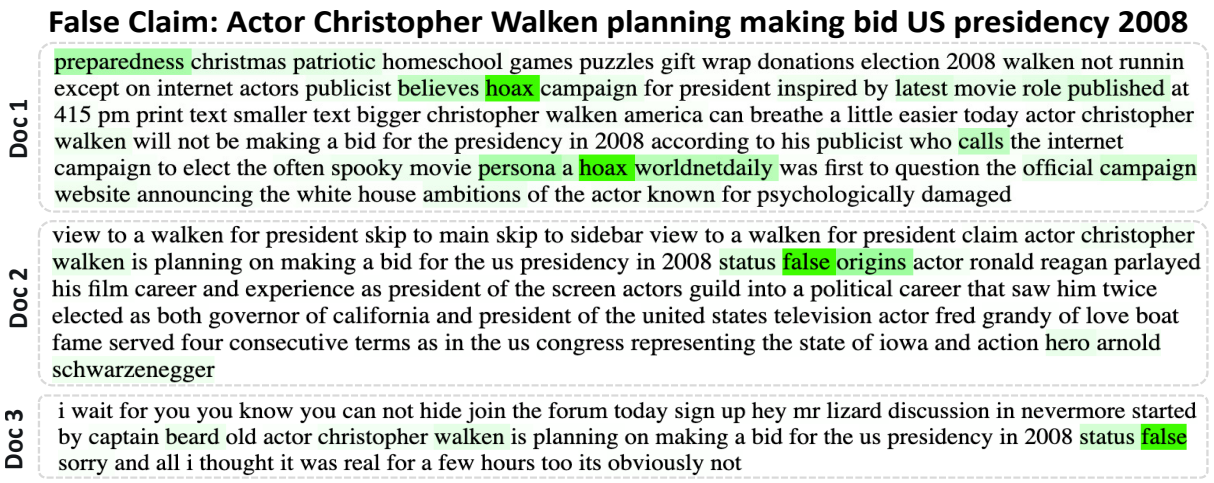


Figure 3: Visualization of attention weights of *the first attention head* on three documents relevant to a false claim in word-level attention layer

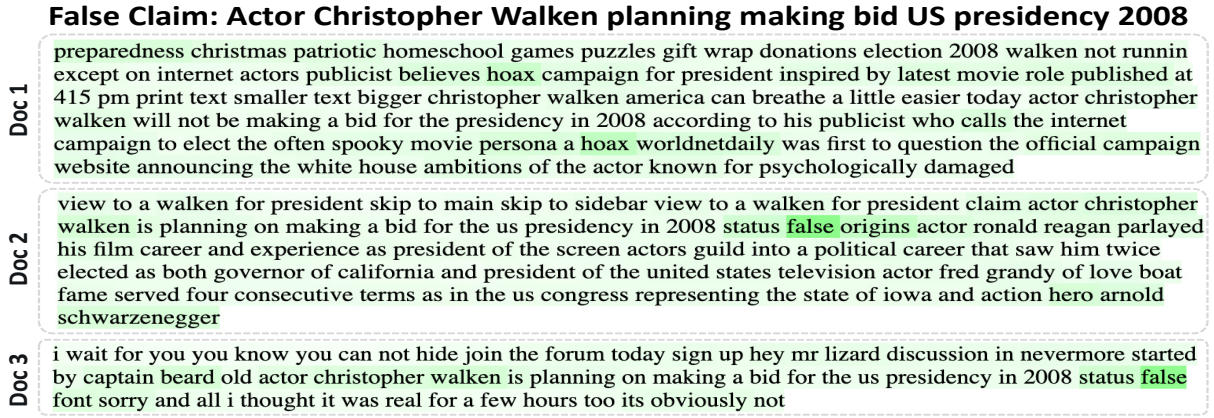


Figure 4: Visualization of attention weights of *the second attention head* on three documents relevant to a false claim in word-level attention layer

crucial to determine verdict of the claims. Finally, using all information (Text + Pubs + Spkrs) helps us achieve the best result in PolitiFact. In Snopes, we omit results of Text + Speakers and Text + Pubs + Spkrs because the dataset does not contain speakers' information. In particular, model *Text + Pubs + Spkrs* outperformed methods *Text Only* and *Text + Publishers* significantly (p-value< 0.05). Based on these results, we conclude that integrating information of speakers and publishers is useful for detecting misinformation.

5.6 Impact of the Number of Attention Heads

In this section, we examine sensitivity of MAC with respect to the number of heads h_1 in word attention layer and the number of heads h_2 in document attention layer. We vary h_1 and h_2 in $\{1, 2, 3, 4, 5\}$. Since AUC is less sensitive to any threshold, we report AUC of MAC on two datasets in Fig. 2(a) and 2(b). A common pattern we can observe in the two figures is that performance of MAC tends to be better when we increase the number heads h_1 in word attention layer while performance of MAC tends to decrease when increasing h_2. This phenomenon indicates that word attention is more important than evidence attention. In Snopes, MAC has the best AUC when $h_1 = 5, h_2 = 2$. In PolitiFact, MAC reaches the peak when $h_1 = 3, h_2 = 1$.

5.7 Case Study

To understand how multi-head attention mechanism works, from the testing set, we visualize attention weights on three documents of a false claim *Actor Christopher Walken planning making bid US presidency 2008*. Note, our MAC correctly classifies the claim as fake news. In Fig. 3 and Fig. 4, we show the claim and visualization of two different heads in word attention layer. Note that Popat et al.

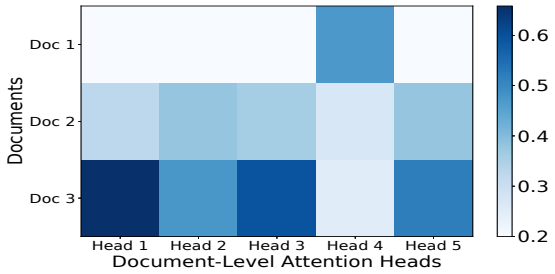


Figure 5: Visualization of five attention heads in document-level attention layer for three documents

(2018), who released the datasets, already lowercased and removed punctuations. To conduct fair comparison, we directly used the datasets without any additional preprocessing. In Fig. 3, attention weights are sparse, indicating that the first attention head focuses on the most important words which determine credibility of the claim (e.g. *hoax*, *false*). Differently, in Fig. 4, the second attention head has more diffused attention weights to capture more useful phrases from documents (e.g. *walken not running*, *its obviously not*). Moving on to attention heads in evidence attention layer in Fig. 5, we show a heat map where the x-axis is the five heads extracted from evidence attention layer and the y-axis is three documents relevant to the same claim in Fig. 3 and 4. As we can see in Fig. 5, *Head 1*, *Head 3* and *Head 5* emphasize on *Doc 3* which contains refuting phrases (e.g. *its obviously not*), while *Head 4* focuses on *Doc 1* which has negating information such as *walken not running*. Both *Doc 1* and *Doc 3* have crucial signals to fact-check the claim. From these analyses, we conclude that heads in word attention layer capture different semantic contributions of words and different heads in document attention layer captures important documents.

6 Conclusions

In this paper, we propose a novel evidence-aware model to fact-check textual claims. Our MAC is designed by hierarchically stacking two attention layers. The first one is a word attention layer and the second one is a document attention layer. In both layers, we propose multi-head attention mechanisms to capture different semantic contributions of words and documents. Our MAC outperforms the baselines significantly with an average increase of 6% to 9% over the best results from baselines with a maximum improvements of 18%. We conduct ablation studies to understand the performance of MAC and provide a case study to show the effectiveness of the attention mechanisms. In future work, we will further examine other data types such as images to improve the performance of our model.

Acknowledgment

This work was supported in part by NSF grant CNS-1755536, AWS Cloud Credits for Research, and Google Cloud. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsors.

References

Hunt Allcott and Matthew Gentzkow. 2017. Social media and fake news in the 2016 election. *Journal of economic perspectives*, 31(2):211–36.

Aparna Alluri. 2019. Whatsapp: The 'black hole' of fake news in india's election. https://www.bbc.com/news/world-asia-india-47797151.

Ashoka. 2020. Misinformation spreads faster than coronavirus: How a social organization in turkey is fighting fake news. https://bit.ly/36qqmmH.

Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. 2011. Information credibility on twitter. In *Proceedings of the 20th international conference on World wide web*, pages 675–684. ACM.

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

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2020. Tabfact: A large-scale dataset for table-based fact verification. In *International Conference on Learning Representations*.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *Neural Information Processing Systems*.

CNN. 2020. How facebook is combating spread of covid-19 misinformation. https://cnn.it/3gjtBkg.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. 2005. Bidirectional lstm networks for improved phoneme classification and recognition. In *International Conference on Artificial Neural Networks*, pages 799–804. Springer.

Aditi Gupta, Hemank Lamba, Ponnurangam Kumaraguru, and Anupam Joshi. 2013. Faking sandy: characterizing and identifying fake images on twitter during hurricane sandy. In *Proceedings of the 22nd international conference on World Wide Web*, pages 729–736.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

Shimon Kogan, Tobias J Moskowitz, and Marina Niessner. 2019. Fake news: Evidence from financial markets. *Available at SSRN 3237763*.

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. In *The 5th International Conference on Learning Representations*.

Xiaomo Liu, Armineh Nourbakhsh, Quanzhi Li, Rui Fang, and Sameena Shah. 2015. Real-time rumor debunking on twitter. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1867–1870. ACM.

Yang Liu and Yi-Fang Brook Wu. 2018. Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *Empirical Methods in Natural Language Processing*.

Jing Ma, Wei Gao, Shafiq Joty, and Kam-Fai Wong. 2019. Sentence-level evidence embedding for claim verification with hierarchical attention networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2561–2571.

Rahul Mishra and Vinay Setty. 2019. Sadhan: Hierarchical attention networks to learn latent aspect embeddings for fake news detection. In *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 197–204.

Yixin Nie, Haonan Chen, and Mohit Bansal. 2019. Combining fact extraction and verification with neural semantic matching networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6859–6866.

Jeffrey Pennington, Richard Socher, and Christopher D 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.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pages 2227–2237.

Kashyap Popat, Subhabrata Mukherjee, Andrew Yates, and Gerhard Weikum. 2018. Declare: Debunking fake news and false claims using evidence-aware deep learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 22–32.

Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. 2017. 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*, pages 2931–2937.

Kai Shu, Limeng Cui, Suhang Wang, Dongwon Lee, and Huan Liu. 2019. defend: Explainable fake news detection. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 395–405.

Kai Shu, Deepak Mahudeswaran, Suhang Wang, and Huan Liu. 2020. Hierarchical propagation networks for fake news detection: Investigation and exploitation. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 14, pages 626–637.

Amir Soleimani, Christof Monz, and Marcel Worring. 2020. Bert for evidence retrieval and claim verification. In *European Conference on Information Retrieval*, pages 359–366. Springer.

Mark Stencel. 2019. Number of fact-checking outlets surges to 188 in more than 60 countries. https://bit.ly/36y3S3l.

James Thorne and Andreas Vlachos. 2017. An extensible framework for verification of numerical claims. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 37–40.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. 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)*, pages 809–819.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Andreas Vlachos and Sebastian Riedel. 2015. Identification and verification of simple claims about statistical properties. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2596–2601. Association for Computational Linguistics.

Nguyen Vo and Kyumin Lee. 2018. The rise of guardians: Fact-checking url recommendation to combat fake news. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 275–284.

Nguyen Vo and Kyumin Lee. 2019. Learning from fact-checkers: Analysis and generation of fact-checking language. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 335–344.

Nguyen Vo and Kyumin Lee. 2020a. Standing on the shoulders of guardians: Novel methodologies to combat fake news. In *Disinformation, Misinformation, and Fake News in Social Media*, pages 183–210. Springer.

Nguyen Vo and Kyumin Lee. 2020b. Where are the facts? searching for fact-checked information to alleviate the spread of fake news. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7717–7731.

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)*, pages 422–426.

Xuezhi Wang, Cong Yu, Simon Baumgartner, and Flip Korn. 2018. Relevant document discovery for fact-checking articles. In *Companion Proceedings of the The Web Conference 2018*, pages 525–533.

Liang Wu and Huan Liu. 2018. Tracing fake-news footprints: Characterizing social media messages by how they propagate. In *Proceedings of the eleventh ACM international conference on Web Search and Data Mining*, pages 637–645.

Lianwei Wu, Yuan Rao, Xiong Yang, Wanzhen Wang, and Ambreen Nazir. 2020. Evidence-aware hierarchical interactive attention networks for explainable claim verification. In *International Joint Conferences on Artificial Intelligence*.

Di You, Nguyen Vo, Kyumin Lee, and Qiang Liu. 2019. Attributed multi-relational attention network for fact-checking url recommendation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1471–1480.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. In *Advances in Neural Information Processing Systems*, pages 9051–9062.

Zhe Zhao, Paul Resnick, and Qiaozhu Mei. 2015. Enquiring minds: Early detection of rumors in social media from enquiry posts. In *Proceedings of the 24th international conference on world wide web*, pages 1395–1405.

Jie Zhou, Xu Han, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2019. Gear: Graph-based evidence aggregating and reasoning for fact verification. page Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics.