Reasoning Over Semantic-Level Graph for Fact Checking

Wanjun Zhong[1]*, **Jingjing Xu**[3]*, **Duyu Tang**[2], **Zenan Xu**[1], **Nan Duan**[2], **Ming Zhou**[2]
Jiahai Wang[1] **and Jian Yin**[1]
[1] The School of Data and Computer Science, Sun Yat-sen University.
Guangdong Key Laboratory of Big Data Analysis and Processing, Guangzhou, P.R.China
[2] Microsoft Research [3] MOE Key Lab of Computational Linguistics, Peking University
{zhongwj25@mail2,xuzn@mail2}.sysu.edu.cn
{wangjiah@mail,issjyin@mail}.sysu.edu.cn
{dutang,nanduan,mingzhou}@microsoft.com; jingjingxu@pku.edu.cn

Abstract

Fact checking is a challenging task because verifying the truthfulness of a claim requires reasoning about multiple retrievable evidence. In this work, we present a method suitable for reasoning about the semantic-level structure of evidence. Unlike most previous works, which typically represent evidence sentences with either string concatenation or fusing the features of isolated evidence sentences, our approach operates on rich semantic structures of evidence obtained by semantic role labeling. We propose two mechanisms to exploit the structure of evidence while leveraging the advances of pre-trained models like BERT, GPT or XLNet. Specifically, using XLNet as the backbone, we first utilize the graph structure to re-define the relative distances of words, with the intuition that semantically related words should have short distances. Then, we adopt graph convolutional network and graph attention network to propagate and aggregate information from neighboring nodes on the graph. We evaluate our system on FEVER, a benchmark dataset for fact checking, and find that rich structural information is helpful and both our graph-based mechanisms improve the accuracy. Our model is the state-of-the-art system in terms of both official evaluation metrics, namely claim verification accuracy and FEVER score.

1 Introduction

Internet provides an efficient way for individuals and organizations to quickly spread information to massive audiences. However, malicious people spread false news, which may have significant influence on public opinions, stock prices, even presidential elections (Faris et al., 2017). Vosoughi et al. (2018) show that false news reaches more people than the truth. The situation is more urgent as advanced pre-trained language models (Radford et al., 2019) can produce remarkably coherent and fluent texts, which lowers the barrier for the abuse of creating deceptive content. In this paper, we study fact checking with the goal of automatically assessing the truthfulness of a textual claim by looking for textual evidence.

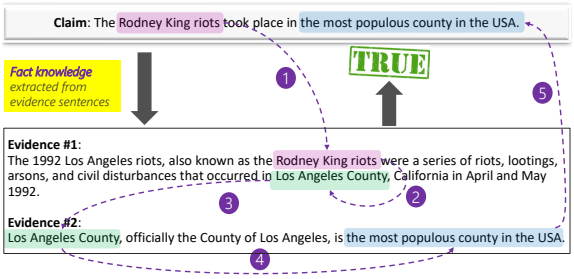

Figure 1: A motivating example for fact checking and the FEVER task. Verifying the claim requires understanding the semantic structure of multiple evidence sentences and the reasoning process over the structure.

Previous works are dominated by natural language inference models (Dagan et al., 2013; Angeli and Manning, 2014) because the task requires reasoning of the claim and retrieved evidence sentences. They typically either concatenate evidence sentences into a single string, which is used in top systems in the FEVER challenge (Thorne et al., 2018b), or use feature fusion to aggregate the features of isolated evidence sentences (Zhou et al., 2019). However, both methods fail to capture rich semantic-level structures among multiple evidence, which also prevents the use of deeper reasoning model for fact checking. In Figure 1, we give a motivating example. Making the correct prediction requires a model to reason based on the understanding that "*Rodney King riots*" is occurred in "*Los Angeles County*" from the first evidence, and that "*Los Angeles County*" is "*the most populous county*

* Work done while this author was an intern at Microsoft Research.

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 6170–6180
July 5 - 10, 2020. ©2020 Association for Computational Linguistics

in the USA" from the second evidence. It is therefore desirable to mine the semantic structure of evidence and leverage it to verify the truthfulness of the claim.

Under the aforementioned consideration, we present a graph-based reasoning approach for fact checking. With a given claim, we represent the retrieved evidence sentences as a graph, and then use the graph structure to guide the reasoning process. Specifically, we apply semantic role labeling (SRL) to parse each evidence sentence, and establish links between arguments to construct the graph. When developing the reasoning approach, we intend to simultaneously leverage rich semantic structures of evidence embodied in the graph and powerful contextual semantics learnt in pre-trained models like BERT (Devlin et al., 2018), GPT (Radford et al., 2019) and XLNet (Yang et al., 2019). To achieve this, we first re-define the distance between words based on the graph structure when producing contextual representations of words. Furthermore, we adopt graph convolutional network and graph attention network to propagate and aggregate information over the graph structure. In this way, the reasoning process employs semantic representations at both word/sub-word level and graph level.

We conduct experiments on FEVER (Thorne et al., 2018a), which is one of the most influential benchmark datasets for fact checking. FEVER consists of 185,445 verified claims, and evidence sentences for each claim are natural language sentences from Wikipedia. We follow the official evaluation protocol of FEVER, and demonstrate that our approach achieves state-of-the-art performance in terms of both claim classification accuracy and FEVER score. Ablation study shows that the integration of graph-driven representation learning mechanisms improves the performance. We briefly summarize our contributions as follows.

- We propose a graph-based reasoning approach for fact checking. Our system apply Semantic Role Labeling (SRL) to construct graphs and present two graph-driven representation learning mechanisms.

- Results verify that both graph-based mechanisms improve the accuracy, and our final system achieves state-of-the-art performance on the FEVER dataset.

2 Task Definition and Pipeline

With a textual claim given as the input, the problem of fact checking is to find supporting evidence sentences to verify the truthfulness of the claim.

We conduct our research on FEVER (Thorne et al., 2018a), short for Fact Extraction and VERification, a benchmark dataset for fact checking. Systems are required to retrieve evidence sentences from Wikipedia, and predict the claim as "*SUPPORTED*", "*REFUTED*" or "*NOT ENOUGH INFO (NEI)*", standing for that the claim is supported by the evidence, refuted by the evidence, and is not verifiable, respectively. There are two official evaluation metrics in FEVER. The first is the accuracy for three-way classification. The second is FEVER score, which further measures the percentage of correct retrieved evidence for "*SUPPORTED*" and "*REFUTED*" categories. Both the statistic of FEVER dataset and the equation for calculating FEVER score are given in Appendix B.

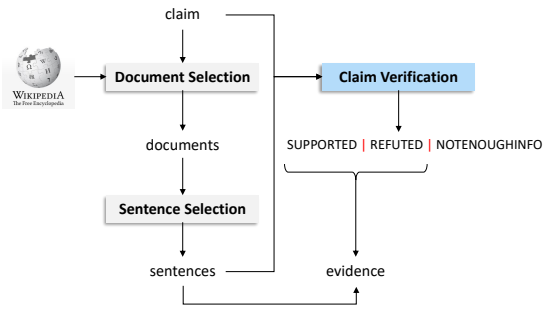

Figure 2: Our pipeline for fact checking on FEVER. The main contribution of this work is a graph-based reasoning model for claim verification.

Here, we present an overview of our pipeline for FEVER, which follows the majority of previous studies. Our pipeline consists of three main components: a document retrieval model, a sentence-level evidence selection model, and a claim verification model. Figure 2 gives an overview of the pipeline. With a given claim, the document retrieval model retrieves the most related documents from a given collection of Wikipedia documents. With retrieved documents, the evidence selection model selects top-k related sentences as the evidence. Finally, the claim verification model takes the claim and evidence sentences as the input and outputs the veracity of the claim.

The main contribution of this work is the graph-based reasoning approach for claim verification, which is explained detailedly in Section 3. Our

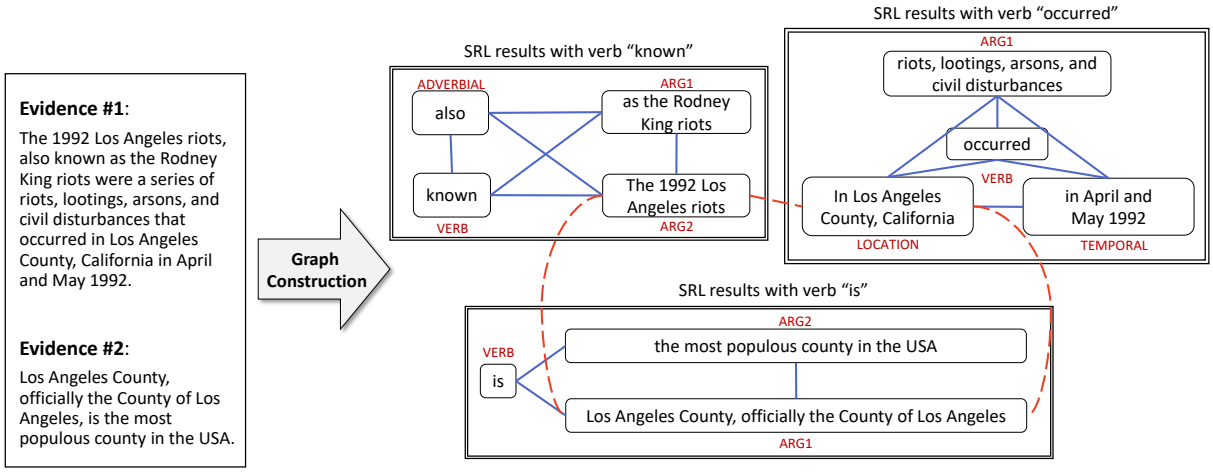

Figure 3: The constructed graph for the motivating example with two evidence sentences. Each box describes a "tuple" which is extracted by SRL triggered by a verb. Blue solid lines indicate edges that connect arguments within a tuple and red dotted lines indicate edges that connect argument across different tuples.

strategies for document selection and evidence selection are described in Section 4.

3 Graph-Based Reasoning Approach

In this section, we introduce our graph-based reasoning approach for claim verification, which is the main contribution of this paper. Taking a claim and retrieved evidence sentences[1] as the input, our approach predicts the truthfulness of the claim. For FEVER, it is a three-way classification problem, which predicts the claim as "*SUPPORTED*", "*REFUTED*" or "*NOT ENOUGH INFO (NEI)*".

The basic idea of our approach is to employ the intrinsic structure of evidence to assess the truthfulness of the claim. As shown in the motivating example in Figure 1, making the correct prediction needs good understanding of the semantic-level structure of evidence and the reasoning process based on that structure. In this section, we first describe our graph construction module (§3.1). Then, we present how to apply graph structure for fact checking, including a contextual representation learning mechanism with graph-based distance calculation (§3.2), and graph convolutional network and graph attention network to propagate and aggregate information over the graph (§3.3 and §3.4).

3.1 Graph Construction

Taking evidence sentences as the input, we would like to build a graph to reveal the intrinsic structure of these evidence. There might be many different ways to construct the graph, such as open information extraction (Banko et al., 2007), named entity recognition plus relation classification, sequence-to-sequence generation which is trained to produce structured tuples (Goodrich et al., 2019), etc. In this work, we adopt a practical and flexible way based on semantic role labeling (Carreras and Màrquez, 2004). Specifically, with the given evidence sentences, our graph construction operates in the following steps.

- For each sentence, we parse it to tuples[2] with an off-the-shelf SRL toolkit developed by AllenNLP[3], which is a re-implementation of a BERT-based model (Shi and Lin, 2019).
- For each tuple, we regard its elements with certain types as the nodes of the graph. We heuristically set those types as verb, argument, location and temporal, which can also be easily extended to include more types. We create edges for every two nodes within a tuple.
- We create edges for nodes across different tuples to capture the structure information among multiple evidence sentences. Our idea is to create edges for nodes that are literally similar with each other. Assuming entity A and entity B come from different tuples, we add one edge if one of the following conditions is satisfied: (1) A equals B; (2) A contains B; (3) the number of overlapped words

[1] Details about how to retrieve evidence for a claim are described in Section 4.

[2] A sentence could be parsed as multiple tuples.

[3] https://demo.allennlp.org/semantic-role-labeling

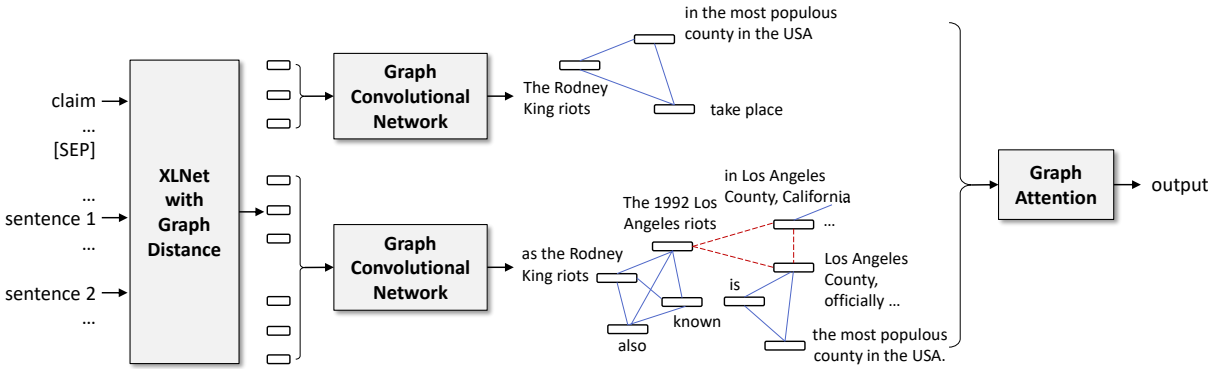

Figure 4: An overview of our graph-based reasoning approach for claim verification. Taking a claim and evidence sentences as the input, we first calculate contextual word representations with graph-based distance (§3.2). After that, we use graph convolutional network to propagate information over the graph (§3.3), and use graph attention network to aggregate information (§3.4) before making the final prediction.

between A and B is larger than the half of the minimum number of words in A and B.

Figure 3 shows the constructed graph of the evidence in the motivating example. In order to obtain the structure information of the claim, we use the same pipeline to represent a claim as a graph.

Our graph construction module offers an approach on modeling structure of multiple evidence, which could be further developed in the future.

3.2 Contextual Word Representations with Graph Distance

We describe the use of graph for learning graph-enhanced contextual representations of words[4].

Our basic idea is to shorten the distance between two semantically related words on the graph, which helps to enhance their relationship when we calculate contextual word representations with a Transformer-based (Vaswani et al., 2017) pre-trained model like BERT and XLNet. Supposing we have five evidence sentences $\{s_1, s_2, ... s_5\}$ and the word w_{1i} from s_1 and the word w_{5j} from s_5 are connected on the graph, simply concatenating evidence sentences as a single string fails to capture their semantic-level structure, and would give a large distance to w_{1i} and w_{5j}, which is the number of words between them across other three sentences (i.e., s_2, s_3, and s_4). An intuitive way to achieve our goal is to define an $N \times N$ matrix of distances of words along the graph, where N is the total number of words in the evidence. However, this is unacceptable in practice because the representation learning procedure will take huge memory space, which is also observed by Shaw et al. (2018).

In this work, we adopt pre-trained model XLNet (Yang et al., 2019) as the backbone of our approach because it naturally involves the concept of relative position[5]. Pre-trained models capture rich contextual representations of words, which is helpful for our task which requires sentence-level reasoning. Considering the aforementioned issues, we implement an approximate solution to trade off between the efficiency of implementation and the informativeness of the graph. Specifically, we reorder evidence sentences with a topology sort algorithm with the intuition that closely linked nodes should exist in neighboring sentences. This would prefer that neighboring sentences contain either parent nodes or sibling nodes, so as to better capture the semantic relatedness between different evidence sentences. We present our implementation in Appendix A. The algorithm begins from nodes without incident relations. For each node without incident relations, we recursively visit its child nodes in a depth-first searching way.

After obtaining graph-based relative position of words, we feed the sorted sequence into XLNet to obtain the contextual representations. Meanwhile, we obtain the representation $h([CLS])$ for a special token $[CLS]$, which stands for the joint representation of the claim and the evidence in Transformer-based architecture.

[4]In Transformer-based representation learning pipeline, the basic computational unit can also be word-piece. For simplicity, we use the term "word" in this paper.

[5]Our approach can also be easily adapted to BERT by adding relative position like Shaw et al. (2018).

3.3 Graph Convolutional Network

We have injected the graph information in Transformer and obtained $h([CLS])$, which captures the semantic interaction between the claim and the evidence at word level [6]. As shown in our motivating example in Figure 1 and the constructed graph in Figure 3, the reasoning process needs to operate on span/argument-level, where the basic computational unit typically consists of multiple words like "*Rodney King riots*" and "*the most popular county in the USA*".

To further exploit graph information beyond word level, we first calculate the representation of a node, which is a word span in the graph, by averaging the contextual representations of words contained in the node. After that, we employ multi-layer graph convolutional network (GCNs) (Kipf and Welling, 2016) to update the node representation by aggregating representations from their neighbors on the graph. Formally, we denote G as the graph constructed by the previous graph construction method and make $H \in \boldsymbol{R}^{N^v \times d}$ a matrix containing representation of all nodes, where N^v and d denote the number of nodes and the dimension of node representations, respectively. Each row $H_i \in \boldsymbol{R}^d$ is the representation of node i. We introduce an adjacency matrix A of graph G and its degree matrix D, where we add self-loops to matrix A and $D_{ii} = \sum_j A_{ij}$. One-layer GCNs will aggregate information through one-hop edges, which is calculated as follows:

$$H_i^{(1)} = \rho(\widetilde{A} H_i W_0), \tag{1}$$

where $H_i^{(1)} \in \boldsymbol{R}^d$ is the new d-dimension representation of node i, $\widetilde{A} = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$ is the normalized symmetric adjacency matrix, W_0 is a weight matrix, and ρ is an activation function. To exploit information from the multi-hop neighboring nodes, we stack multiple GCNs layers:

$$H_i^{(j+1)} = \rho(\widetilde{A} H_i^{(j)} W_j), \tag{2}$$

where j denotes the layer number and H_i^0 is the initial representation of node i initialized from the contextual representation. We simplify $H^{(k)}$ as $\boldsymbol{H}$ for later use, where $\boldsymbol{H}$ indicates the representation of all nodes updated by k-layer GCNs.

[6] By "word" in "word-level", we mean the basic computational unit in XLNet, and thus $h([CLS])$ capture the sophisticated interaction between words via multi-layer multi-head attention operations.

The graph learning mechanism will be performed separately for claim-based and evidence-based graph. Therefore, we denote $\boldsymbol{H}_c$ and $\boldsymbol{H}_e$ as the representations of all nodes in claim-based graph and evidence-based graphs, respectively. Afterwards, we utilize the graph attention network to align the graph-level node representation learned for two graphs before making the final prediction.

3.4 Graph Attention Network

We explore the related information between two graphs and make semantic alignment for final prediction. Let $\boldsymbol{H}_e \in \boldsymbol{R}^{N_e^v \times d}$ and $\boldsymbol{H}_c \in \boldsymbol{R}^{N_c^v \times d}$ denote matrices containing representations of all nodes in evidence-based and claim-based graph respectively, where N_e^v and N_c^v denote number of nodes in the corresponding graph.

We first employ a graph attention mechanism (Veličković et al., 2017) to generate a claim-specific evidence representation for each node in claim-based graph. Specifically, we first take each $h_c^i \in \boldsymbol{H_c}$ as query, and take all node representations $h_e^j \in \boldsymbol{H_e}$ as keys. We then perform graph attention on the nodes, an attention mechanism $a : \boldsymbol{R}^d \times \boldsymbol{R}^d \rightarrow \boldsymbol{R}$ to compute attention coefficient as follows:

$$e_{ij} = a(\boldsymbol{W_c} h_c^i, \boldsymbol{W_e} h_e^j) \tag{3}$$

which means the importance of evidence node j to the claim node i. $W_c \in \boldsymbol{R}^{F \times d}$ and $W_e \in \boldsymbol{R}^{F \times d}$ is the weight matrix and F is the dimension of attention feature. We use the dot-product function as a here. We then normalize e_{ij} using the softmax function:

$$\alpha_{ij} = softmax_j(e_{ij}) = \frac{exp(e_{ij})}{\sum_{k \in N_e^v} exp(e_{ik})} \tag{4}$$

After that, we calculate a claim-centric evidence representation $\boldsymbol{X} = [x_1, \ldots, x_{N_c^v}]$ using the weighted sum over $\boldsymbol{H_e}$:

$$x_i = \sum_{j \in N_e^v} \alpha_{ij} h_e^j \tag{5}$$

We then perform node-to-node alignment and calculate aligned vectors $A = [a_1, \ldots, a_{N_c^v}]$ by the claim node representation $\boldsymbol{H^c}$ and the claim-centric evidence representation $\boldsymbol{X}$,

$$a_i = f_{align}(h_c^i, x^i), \tag{6}$$

where $f_{align}()$ denotes the alignment function. Inspired by Shen et al. (2018), we design our alignment function as:

$$f_{align}(x, y) = W_a[x, y, x - y, x \odot y], \tag{7}$$

where $W_a \in \boldsymbol{R}^{d \times 4*d}$ is a weight matrix and $\odot$ is element-wise Hadamard product. The final output g is obtained by the mean pooling over A. We then feed the concatenated vector of g and the final hidden vector $h([CLS])$ from XLNet through a MLP layer for the final prediction.

4 Document Retrieval and Evidence Selection

In this section, we briefly describe our document retrieval and evidence selection components to make the paper self contained.

4.1 Document Retrieval

The document retrieval model takes a claim and a collection of Wikipedia documents as the input, and returns m most relevant documents.

We mainly follow Nie et al. (2019), the top-performing system on the FEVER shared task (Thorne et al., 2018b). The document retrieval model first uses keyword matching to filter candidate documents from the massive Wikipedia documents. Then, NSMN (Nie et al., 2019) is applied to handle the documents with disambiguation titles, which are 10% of the whole documents. Documents without disambiguation title are assigned with higher scores in the resulting list. The input to the NSMN model includes the claim and candidate documents with disambiguation title. At a high level, NSMN model has encoding, alignment, matching and output layers. Readers who are interested are recommended to refer to the original paper for more details.

Finally, we select top-10 documents from the resulting list.

4.2 Sentence-Level Evidence Selection

Taking a claim and all the sentences from retrieved documents as the input, evidence selection model returns the top-k most relevant sentences.

We regard evidence selection as a semantic matching problem, and leverage rich contextual representations embodied in pre-trained models like XLNet (Yang et al., 2019) and RoBERTa (Liu et al., 2019a) to measure the relevance of a claim to every evidence candidate. Let's take XLNet as an example. The input of the sentence selector is

$$ce_i = [Claim, SEP, Evidence_i, SEP, CLS]$$

where $Claim$ and $Evidence_i$ indicate tokenized word-pieces of original claim and i^{th} evidence candidate, d denotes the dimension of hidden vector, and SEP and CLS are symbols indicating ending of a sentence and ending of a whole input, respectively. The final representation $h_{\boldsymbol{ce_i}} \in \boldsymbol{R}^d$ is obtained via extracting the hidden vector of the CLS token.

After that, we employ an MLP layer and a softmax layer to compute score $s^+_{ce_i}$ for each evidence candidate. Then, we rank all the evidence sentences by score $s^+_{ce_i}$. The model is trained on the training data with a standard cross-entropy loss. Following the official setting in FEVER, we select top-5 evidence sentences. The performance of our evidence selection model is shown in Appendix C.

5 Experiments

We evaluate on FEVER (Thorne et al., 2018a), a benchmark dataset for fact extraction and verification. Each instance in FEVER dataset consists of a claim, groups of ground-truth evidence from Wikipedia and a label (i.e., "*SUPPORTED*", "*REFUTED*" or "*NOT ENOUGH INFO (NEI)*"), indicating its veracity. FEVER includes a dump of Wikipedia, which contains 5,416,537 pre-processed documents. The two official evaluation metrics of FEVER are label accuracy and FEVER score, as described in Section 2. Label accuracy is the primary evaluation metric we apply for our experiments because it directly measures the performance of the claim verification model. We also report FEVER score for comparison, which measures whether both the predicted label and the retrieved evidence are correct. No evidence is required if the predicted label is *NEI*.

5.1 Baselines

We compare our system to the following baselines, including three top-performing systems on FEVER shared task, a recent work GEAR (Zhou et al., 2019), and a concurrent work by Liu et al. (2019b).

- Nie et al. (2019) employ a semantic matching neural network for both evidence selection and claim verification.
- Yoneda et al. (2018) infer the veracity of each claim-evidence pair and make final prediction by aggregating multiple predicted labels.
- Hanselowski et al. (2018) encode each claim-evidence pair separately, and use a pooling function to aggregate features for prediction.

Method	Label Acc (%)	FEVER Score (%)
Hanselowski et al. (2018)	65.46	61.58
Yoneda et al. (2018)	67.62	62.52
Nie et al. (2019)	68.21	64.21
GEAR (Zhou et al., 2019)	71.60	67.10
KGAT (Liu et al., 2019b)	72.81	69.40
DREAM (our approach)	**76.85**	**70.60**

Table 1: Performance on the blind test set on FEVER. Our approach is abbreviated as DREAM.

- GEAR (Zhou et al., 2019) uses BERT to obtain claim-specific representation for each evidence sentence, and applies graph network by regarding each evidence sentence as a node in the graph.

- KGAT (Liu et al., 2019b) is concurrent with our work, which regards sentences as the nodes of a graph and uses Kernel Graph Attention Network to aggregate information.

5.2 Model Comparison

Table 1 reports the performance of our model and baselines on the blind test set with the score showed on the public leaderboard[7]. As shown in Table 1, in terms of label accuracy, our model significantly outperforms previous systems with 76.85% on the test set. It is worth noting that, our approach, which exploits explicit graph-level semantic structure of evidence obtained by SRL, outperforms GEAR and KGAT, both of which regard sentences as the nodes and use model to learn the implicit structure of evidence [8]. By the time our paper is submitted, our system achieves state-of-the-art performance in terms of both evaluation metrics on the leaderboard.

5.3 Ablation Study

Table 2 presents the label accuracy on the development set after eliminating different components (including the graph-based relative distance (§3.2) and graph convolutional network and graph attention network (§3.3 and §3.4) separately in our model.

[7]The public leaderboard for perpetual evaluation of FEVER is https://competitions.codalab.org/competitions/18814#results. DREAM is our user name on the leaderboard.

[8]We don't overclaim that the superiority of our system to GEAR and KGAT only comes from the explicit graph structure, because we have differences in other components like sentence selection and the pre-trained model.

Model	Label Accuracy
DREAM	79.16
-w/o Relative Distance	78.35
-w/o GCN&GAN	77.12
-w/o both above modules	75.40

Table 2: Ablation study on develop set.

The last row in Table 2 corresponds to the baseline where all the evidence sentences are simply concatenated as a single string, where no explicit graph structure is used at all for fact verification.

As shown in Table 2, compared to the XLNet baseline, incorporating both graph-based modules brings 3.76% improvement on label accuracy. Removing the graph-based distance drops 0.81% in terms of label accuracy. The graph-based distance mechanism can shorten the distance of two closely-linked nodes and help the model to learn their dependency. Removing the graph-based reasoning module drops 2.04% because graph reasoning module captures the structural information and performs deep reasoning about that. Figure 5 gives a case study of our approach.

5.4 Error Analysis

We randomly select 200 incorrectly predicted instances and summarize the primary types of errors.

The first type of errors is caused by failing to match the semantic meaning between phrases that describe the same event. For example, the claim states *"Winter's Tale is a book"*, while the evidence states *"Winter 's Tale is a 1983 novel by Mark Helprin"*. The model fails to realize that "novel" belongs to "book" and states that the claim is refuted. Solving this type of errors needs to involve external knowledge (e.g. ConceptNet (Speer et al., 2017)) that can indicate logical relationships between different events.

The misleading information in the retrieved evidence causes the second type of errors. For example, the claim states "*The Gifted is a movie*", and the ground-truth evidence states "*The Gifted is an upcoming American television series*". However, the retrieved evidence also contains "*The Gifted is a 2014 Filipino dark comedy-drama movie*", which misleads the model to make the wrong judgment.

6 Related Work

In general, fact checking involves assessing the truthfulness of a claim. In literature, a claim can be

Claim	**Text**: Congressional Space Medal of Honor is the highest award given only to astronauts by NASA. **Tuples:** ('Congressional Space Medal of Honor', 'is', 'the highest award given only to astronauts by NASA') ('the highest award', 'given','only', 'to astronauts', 'by NASA')
Evidence #1	**Text**: The highest award given by NASA , Congressional Space Medal of Honor is awarded by the President of the United States in Congress 's name on recommendations from the Administrator of the National Aeronautics and Space Administration . **Tuples**: ('The highest award','given','by NASA') ('Congressional Space Medal of Honor','awarded','by the President of the United States')
Evidence #2	**Text**: To be awarded the Congressional Space Medal of Honor , an astronaut must perform feats of extraordinary accomplishment while participating in space flight under the authority of NASA . **Tuples**: ('awarded', 'the Congressional Space Medal of Honor') ('To be awarded the Congressional Space Medal of Honor','an astronaut','perform','feats of extraordinary accomplishment') ('an astronaut', 'participating','in space flight','under the authority of NASA')

Figure 5: A case study of our approach. Facts shared across the claim and the evidence are highlighted with different colors.

a text or a subject-predicate-object triple (Nakashole and Mitchell, 2014). In this work, we only consider textual claims. Existing datasets differ from data source and the type of supporting evidence for verifying the claim. An early work by Vlachos and Riedel (2014) constructs 221 labeled claims in the political domain from POLITIFACT.COM and CHANNEL4.COM, giving meta-data of the speaker as the evidence. POLIFACT is further investigated by following works, including Ferreira and Vlachos (2016) who build Emergent with 300 labeled rumors and about 2.6K news articles, Wang (2017) who builds LIAR with 12.8K annotated short statements and six fine-grained labels, and Rashkin et al. (2017) who collect claims without meta-data while providing 74K news articles. We study FEVER (Thorne et al., 2018a), which requires aggregating information from multiple pieces of evidence from Wikipedia for making the conclusion. FEVER contains 185,445 annotated instances, which to the best of our knowledge is the largest benchmark dataset in this area.

The majority of participating teams in the FEVER challenge (Thorne et al., 2018b) use the same pipeline consisting of three components, namely document selection, evidence sentence selection, and claim verification. In document selection phase, participants typically extract named entities from a claim as the query and use Wikipedia search API. In the evidence selection phase, participants measure the similarity between the claim and an evidence sentence candidate by training a classification model like Enhanced LSTM (Chen et al., 2016) in a supervised setting or using string similarity function like TFIDF without trainable parameters. Padia et al. (2018) utilizes semantic frames for evidence selection. In this work, our focus is the claim classification phase. Top-ranked three systems aggregate pieces of evidence through concatenating evidence sentences into a single string (Nie et al., 2019), classifying each evidence-claim pair separately, merging the results (Yoneda et al., 2018), and encoding each evidence-claim pair followed by pooling operation (Hanselowski et al., 2018). Zhou et al. (2019) are the first to use BERT to calculate claim-specific evidence sentence representations, and then develop a graph network to aggregate the information on top of BERT, regarding each evidence as a node in the graph. Our work differs from Zhou et al. (2019) in that (1) the construction of our graph requires understanding the syntax of each sentence, which could be viewed as a more fine-grained graph, and (2) both the contextual representation learning module and the reasoning module have model innovations of taking the graph information into consideration. Instead of training each component separately, Yin and Roth (2018) show that joint learning could improve both claim verification and evidence selection.

7 Conclusion

In this work, we present a graph-based approach for fact checking. When assessing the veracity of a claim giving multiple evidence sentences, our approach is built upon an automatically constructed graph, which is derived based on semantic role labeling. To better exploit the graph information, we propose two graph-based modules, one for calculating contextual word embeddings using graph-based distance in XLNet, and the other for learning representations of graph components and reasoning over the graph. Experiments show that both graph-based modules bring improvements and our final system is the state-of-the-art on the public leaderboard by the time our paper is submitted.

Evidence selection is an important component of fact checking as finding irrelevant evidence may lead to different predictions. A potential solution

is to jointly learn evidence selection and claim verification model, which we leave as a future work.

Acknowledgement

Wanjun Zhong, Zenan Xu, Jiahai Wang and Jian Yin are supported by the National Natural Science Foundation of China (U1711262, U1611264,U1711261,U1811261,U1811264, U1911203), National Key R&D Program of China (2018YFB1004404), Guangdong Basic and Applied Basic Research Foundation (2019B1515130001), Key R&D Program of Guangdong Province (2018B010107005). The corresponding author is Jian Yin.

References

Gabor Angeli and Christopher D Manning. 2014. Naturalli: Natural logic inference for common sense reasoning. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 534–545.

Michele Banko, Michael J Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *Ijcai*, volume 7, pages 2670–2676.

Xavier Carreras and Lluís Màrquez. 2004. Introduction to the conll-2004 shared task: Semantic role labeling. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004*, pages 89–97.

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2016. Enhanced lstm for natural language inference. *arXiv preprint arXiv:1609.06038*.

Ido Dagan, Dan Roth, Mark Sammons, and Fabio Massimo Zanzotto. 2013. Recognizing textual entailment: Models and applications. *Synthesis Lectures on Human Language Technologies*, 6(4):1–220.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Robert Faris, Hal Roberts, Bruce Etling, Nikki Bourassa, Ethan Zuckerman, and Yochai Benkler. 2017. Partisanship, propaganda, and disinformation: Online media and the 2016 us presidential election.

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.

Ben Goodrich, Vinay Rao, Peter J Liu, and Mohammad Saleh. 2019. Assessing the factual accuracy of generated text. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 166–175. ACM.

Andreas Hanselowski, Hao Zhang, Zile Li, Daniil Sorokin, Benjamin Schiller, Claudia Schulz, and Iryna Gurevych. 2018. Ukp-athene: Multi-sentence textual entailment for claim verification. *arXiv preprint arXiv:1809.01479*.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019a. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Zhenghao Liu, Chenyan Xiong, and Maosong Sun. 2019b. Kernel graph attention network for fact verification. *arXiv preprint arXiv:1910.09796*.

Ndapandula Nakashole and Tom M Mitchell. 2014. Language-aware truth assessment of fact candidates. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1009–1019.

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.

Ankur Padia, Francis Ferraro, and Tim Finin. 2018. Team UMBC-FEVER : Claim verification using semantic lexical resources. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 161–165, Brussels, Belgium. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).

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.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*.

Dinghan Shen, Xinyuan Zhang, Ricardo Henao, and Lawrence Carin. 2018. Improved semantic-aware network embedding with fine-grained word alignment. *arXiv preprint arXiv:1808.09633*.

Peng Shi and Jimmy Lin. 2019. Simple bert models for relation extraction and semantic role labeling. *arXiv preprint arXiv:1904.05255*.

Robert Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-First AAAI Conference on Artificial Intelligence*.

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

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018b. The fact extraction and verification (fever) shared task. *arXiv preprint arXiv:1811.10971*.

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.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

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.

Soroush Vosoughi, Deb Roy, and Sinan Aral. 2018. The spread of true and false news online. *Science*, 359(6380):1146–1151.

William Yang Wang. 2017. " liar, liar pants on fire": A new benchmark dataset for fake news detection. *arXiv preprint arXiv:1705.00648*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

Wenpeng Yin and Dan Roth. 2018. Twowingos: A two-wing optimization strategy for evidential claim verification. *arXiv preprint arXiv:1808.03465*.

Takuma Yoneda, Jeff Mitchell, Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Ucl machine reading group: Four factor framework for fact finding (hexaf). In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 97–102.

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. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 892–901, Florence, Italy. Association for Computational Linguistics.

A Typology Sort Algorithm

Algorithm 1 Graph-based Distance Calculation Algorithm.

Require: A sequence of nodes $S = \{s_i, s_2, \cdots, s_n\}$; A set of relations $R = \{r_1, r_2, \cdots, r_m\}$

function DFS(node, visited, sorted_sequence)
  for each child s_c in node's children **do**
    if s_c has no incident edges and visited[s_c]==0 **then**
      visited[s_c]=1
      DFS(s_c, visited)
    end if
  end for
  sorted_sequence.append(0, node)
end function
sorted_sequence = []
visited = [0 for i in range(n)]
S,R = changed_to_acyclic_graph(S,R)
for each node s_i in S **do**
  if s_i has no incident edges and visited[i] == 0 **then**
    visited[i] = 1
    for each child s_c in s_i's children **do**
      DFS(s_c, visited, sorted_sequence)
    end for
    sorted_sequence.append(0,s_i)
  end if
end for
return sorted_sequence

B FEVER

The statistic of FEVER is shown in Table 3.

Split	SUPPORTED	REFUTED	NEI
Training	80,035	29,775	35,659
Dev	6,666	6,666	6,666
Test	6,666	6,666	6,666

Table 3: Split size of SUPPORTED, REFUTED and NOT ENOUGH INFO (NEI) classes in FEVER.

FEVER score is calculated with equation 8, where y is the ground truth label, $\hat{y}$ is the predicted label, $\boldsymbol{E} = [E_1, \cdots, E_k]$ is a set of ground-truth evidence, and $\hat{\boldsymbol{E}} = [\hat{E}_1, \cdots, \hat{E}_5]$ is a set of predicted evidence.

$$Instance\_Correct(y, \hat{y}, \boldsymbol{E}, \hat{\boldsymbol{E}}) \stackrel{def}{=} y = \hat{y} \wedge (y = NEI \vee Evidence\_Correct(\boldsymbol{E}, \hat{\boldsymbol{E}})) \quad (8)$$

C Evidence Selection Results

In this part, we present the performance of the sentence-level evidence selection module that we develop with different backbone. We take the concatenation of claim and each evidence as input, and take the last hidden vector to calculate the score for evidence ranking. In our experiments, we try both

RoBERTa and XLNet. From Table 4, we can see that RoBERTa performs slightly better than XLNet here. When we submit our system on the leaderboard, we use RoBERTa as the evidence selection model.

Model	Dev. Set			Test Set		
	Acc.	Rec.	F1	Acc.	Rec.	F1
XLNet	26.60	87.33	40.79	25.55	85.34	39.33
RoBERTa	26.67	87.64	40.90	25.63	85.57	39.45

Table 4: Results of evidence selection models.

D Training Details

In this part, we describe the training details of our experiments. We employ cross-entropy loss as the loss function. We apply AdamW as the optimizer for model training. For evidence selection model, we set learning rate as 1e-5, batch size as 8 and maximum sequence length as 128.

In claim verification model, the XLNet network and graph-based reasoning network are trained separately. We first train XLNet and then freeze the parameters of XLNet and train the graph-based reasoning network. We set learning rate as 2e-6, batch size as 6 and set maximum sequence length as 256. We set the dimension of node representation as 100.