Heterogeneous-Graph Reasoning and Fine-Grained Aggregation for Fact Checking

Anonymous ACL submission

Abstract

Fact checking is a challenging task that requires corresponding evidences to verify the property of a claim based on reasoning. Previous studies generally i) construct the graph by treating each evidence-claim pair as node which is a simple way that ignores to exploit their implicit interaction, or building a fully-connected graph among claim and evidences where the entailment relationship between claim and evidence would be considered equally to the semantic relationship among evidences; ii) aggregate evidences equally without considering their different stances towards the verification of fact. Towards the above issues, we propose a novel heterogeneous-graph reasoning and fine-grained aggregation model, with two following modules: 1) a heterogeneous graph attention network module to distinguish different types of relationships within the constructed graph; 2) fine-grained aggregation module which learns the implicit stance of evidences towards the prediction result in details. Extensive experiments on the benchmark dataset demonstrate that our proposed model achieves much better performance than state-of-the-art methods.

1 Introduction

Today, social media is considered as the biggest platform to share news and seek information. However, misinformation is spreading at increasing rates and may cause great impact to society. The reach of fake news was best highlighted during the critical months of the 2016 U.S. presidential election generated millions of shares and comments on Facebook (Zafarani et al., 2019). Therefore, automatic detection of fake news on social media has become a significant and beneficial problem. We pay more attention on fact checking task, which utilizes external knowledge to determine the claim veracity when given a claim.

Verifying the truthfulness of a claim with respect to evidence can be regarded as a special case of recognizing textual entailment (RTE) (Dagan et al., 2005) or natural language inference (NLI) (Bowman et al., 2015). Typically, existing approaches contain the representation learning process and evidence aggregation process. Representation process tries to enhance the semantic expression of claim and evidence via sequence structure methods (Hanselowski et al., 2018a; Soleimani et al., 2020) or graph based neural networks (Zhou et al., 2019; Liu et al., 2019) where they utilize simple combination methods such as just dealing with claim-evidence pair as graph nodes. The evidence aggregation process aims to find out the most important evidence which contributes more to claim verification with different methods like mean pooling, attention-based aggregation, etc.

However, existing approaches Liu et al. (2019) establish a semantic-based graph, which ignore the difference between relationships among nodes in reasoning graph. For example in Figure 1, given the claim "*Al Jardine is an American rhythm guitarist.*" and the retrieved evidence sentences (i.e., *E1-E5*), making the correct prediction requires model to reason that "*Al Jardine*" is the person mentioned in $E2$ and "*rhythm guitarist*" is occurred in $E1$ based on the entailment interaction of claim with the evidences. Furthermore, we also expect the semantical coherence of multiple evidences from *E1* to *E5* to automatically filter unrelated evidence such as $E3$-$E5$. We believe it's crucial for verification to mine distinct relationships within the reasoning graph.

Besides, in previous methods, stance of evidences towards claim are aggregated equally or some irrelevant evidences are prevented from predicting the veracity of claim roughly via simple attention mechanism. However, each piece of evidence has a different impact on the claim, which needs to be exploited on fine-grained perspective.

To alleviate above issues, we propose a novel **H**eterogeneous-**G**raph **R**easoning and Fine-**G**rained **A**ggregation Model (HGRGA), which not

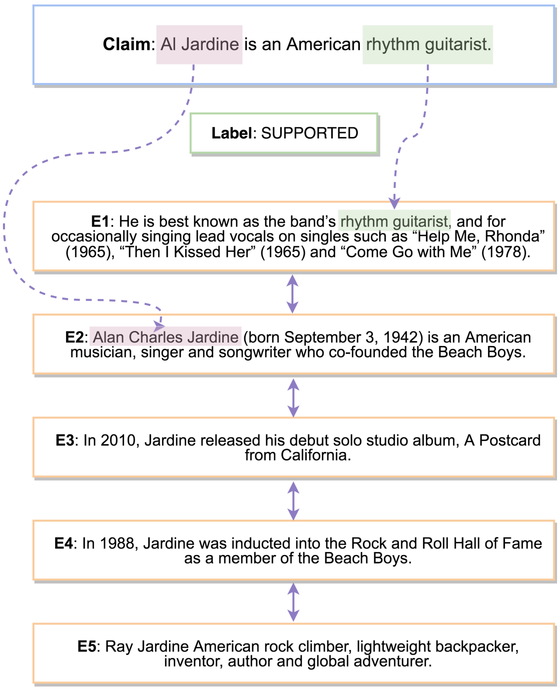


Figure 1: A motivating example for fact checking and the FEVER task. The purple solid line denotes the semantical coherence between each piece of evidence. The purple dotted line denotes entailment consistence between claim and evidences. Verifying the fact requires exploiting these two different implicit relationships during reasoning process.

only enhances the representation learning for claim and evidences by capturing different types of relationships within the constucted graph but also aggregating stances of evidences towards claim concretely. More specifically, we construct a heterogeneous evidence-evidence-claim graph based on graph attention network to enhance the representation of claim and evidences. Besides, we utilize an capsule network to further aggregate evidences with different implicit stances towards the claim, and learn the weights via dynamic routing which indicate how each of evidence attributes the veracity of claim.

We conduct experiments on the real-world benchmark dataset. Extensive experimental results demonstrate the effectiveness of our model. HGRGA boosts the performance for fact checking and the main contributions of this work are summarized as follows:

- To our best knowledge, this is the first study of representing reasoning structure as a heterogeneous graph. The graph attention based heterogeneous interaction achieves significant improvements over state-of-the-art methods.
- We incorporate the capsule network structure into our proposed model to learn implicit stances of evidences towards the claim on fine-grained perspective.
- Experimental results show that our model achieves superior performance on the large-scale benchmark dataset for fact verification.

2 Background and related work

2.1 Problem fomulation

The input of our task is a claim and a collection of Wikipedia articles D. The goal is to extract a set of evidence sentences from D and assign a veracity relation label $y \in \mathcal{Y} = \{\mathrm{S}, \mathrm{R}, \mathrm{N}\}$ to a claim with respect to the evidence set, where **S** = *SUPPORTED*, **R** = *REFUTED*, and **N** = *NOTENOUGHINFO(NEI)*.

2.2 Fact checking

The process of evidence-based fact checking involves the following three subtasks: document retrieval, evidence sentence selection and claim verification. In the document retrieval phrase, researchers use a hybrid approach that combines search results from the MediaWiki API[1] and the results on the basic of the term frequency-inverse document frequency (TF-IDF) model (Hanselowski et al., 2018b). In the evidence sentence selection phrase, Nie et al. (2019); Hanselowski et al. (2018b) use the enhanced sequential inference model (ESIM) to encode and align a claim-evidence pair. Chen et al. (2016) train a ranking model to rank evidence sentences via different kinds of loss, such as pointwise and pairwise loss. Many fact checking approaches aims to improve the performance of claim verification phrase. Previous work modified existing RTE/NLI models to deal with multiple sentences (Thorne et al., 2018a; Nie et al., 2019; Hanselowski et al., 2018b), concatenated all sentence (Stammbach and Neumann, 2019).

Recently, there are some approaches related to graph-based neural networks (Kipf and Welling, 2016). For example, Zhou et al. (2019) build a fully-connected evidence graph where each node indicates a piece of evidence while Liu et al. (2019) conduct fine-grained evidence propagation in the graph. Zhong et al. (2019) use semantic role labeling (SRL) to build a graph structure, where a node

[1] https://www.mediawiki.org/wiki/API

can be a word or a phrase depending on the SRL's outputs.

2.3 Pre-trained language models

Pre-trained language representation models such as GPT (Radford et al., 2018), BERT (Devlin et al., 2018) are proven to be effective on many NLP tasks. These models employ well-designed pre-training tasks to fuse context information and train on rich data. Each BERT layer transforms an input token sequence (one or two sentences) by using self-attention mechanism. Hence, we use BERT as the sentence encoder in our framework to encode better semantic representation.

2.4 Capsule network

A recent method called capsule network explored by Sabour et al. (2017) introduces an iterative routing process to learn a hierarchy of feature detectors which send low-level features to high-level capsules only when there is a strong agreement of their predictions to high-level capsules. Researchers recently apply capsule network into NLP task such as text classification (Zhao et al., 2018), slot filling (Zhang et al., 2018), etc.

3 Proposed method

In this section, we present an overview of the architecture of the proposed framework HGRGA for fact verification. As shown in Figure 2, given a claim and the retrieved evidence, we first utilize a sentence encoder to obtain representations for the claim and the evidences. Then we build a heterogeneous evidence-evidence-claim graph to propagate information among claim and evidence. Finally, we use the capsule network to model the implicit stances of evidences towards claim on fine-grained perspective.

3.1 Sentence Encoder

Given an input sentence, we employ BERT (Devlin et al., 2018) as our sentence encoder by extracting the final hidden state of the [CLS] token as the representation, where [CLS] is the special classification embedding in BERT.

Specifically, given a claim c and N pieces of retrieved evidence $\{e_1, e_2, \ldots, e_N\}$, we feed each sentence into BERT to obtain the claim representation $\mathbf{c}$ and the evidence representation $\mathbf{e_i}$. That is,

$$\begin{aligned} \mathbf{c} &= \mathrm{BERT}(c) \\ \mathbf{e_i} &= \mathrm{BERT}(e_i) \end{aligned} \tag{1}$$

We thus denote the utterance as a matrix, i.e., $X = [\mathbf{c}, \mathbf{e_1}, \mathbf{e_2}, ..., \mathbf{e_N}]^{\mathrm{T}}$, where $\mathbf{c}, \mathbf{e_i} \in \mathbb{R}^d$ respectively denotes the d-dimensional embedding of the claim and each relative evidence.

3.2 Graph Reasoning Network

This section describes how to incorporate the heterogeneous graph attention network into our model. Based on the observation as illustrated in Figure 1, we assume that given a claim, the evidence should be semantically coherent with each other while the claim should be entailment consistent with the relevant evidence. Therefore, we decompose the evidence-evidence-claim graph into claim-evidence subgraph and evidence-evidence subgraph.

Claim-Evidence Subgraph Considering that the neighbors of each node in subgraphs have different importance to learn node embedding for fact checking task, we use graph attention network (GAT) (Veličković et al., 2017) to generate the sentence representation of claim and the retrieved evidence.

We use $H^l_{ce} = [h^l_0, h^l_1, h^l_2, ..., h^l_N]^{\mathrm{T}}$ to represent the hidden states of nodes at layer l and initially, $H^0_{ce} = X$. In order to encode structural contexts to improve the sentence-level representation by adaptively learning different contributions of neighbors to each node, we perform self-attention mechanism on the nodes to model the interactions between each node and its neighbors. The attention coefficient can be computed as follows:

$$\begin{aligned} \alpha^l_{i,j} &= Atten(h^l_i, h^l_j) \\ &= \frac{exp(\phi(a^T[W^l h^l_i || W^l h^l_j]))}{\sum_{j \in N_i} exp(\phi(a^T[W^l h^l_i || W^l h^l_j]))} \end{aligned} \tag{2}$$

where $\alpha^l_{i,j}$ indicates the importance of node i to j at layer l, a is a weight vector, W^l is a layer-specific trainable transformation matrix, $||$ means "concatenate" operation, N_i contains node i's one-hop neighbors and node i itself, ϕ denotes the activation function, such as LeakyReLU (Girshick et al., 2014). Here, we use the adjacency matrix A^{ce} to denotes the relationship between each node, which is defined as:

$$A^{ce}_{i,j} = \begin{cases} 1 & i \in \{claim\}, \\ & j \in \{claim, e_1, ..., e_N\} \\ 0 & otherwise \end{cases} \tag{3}$$

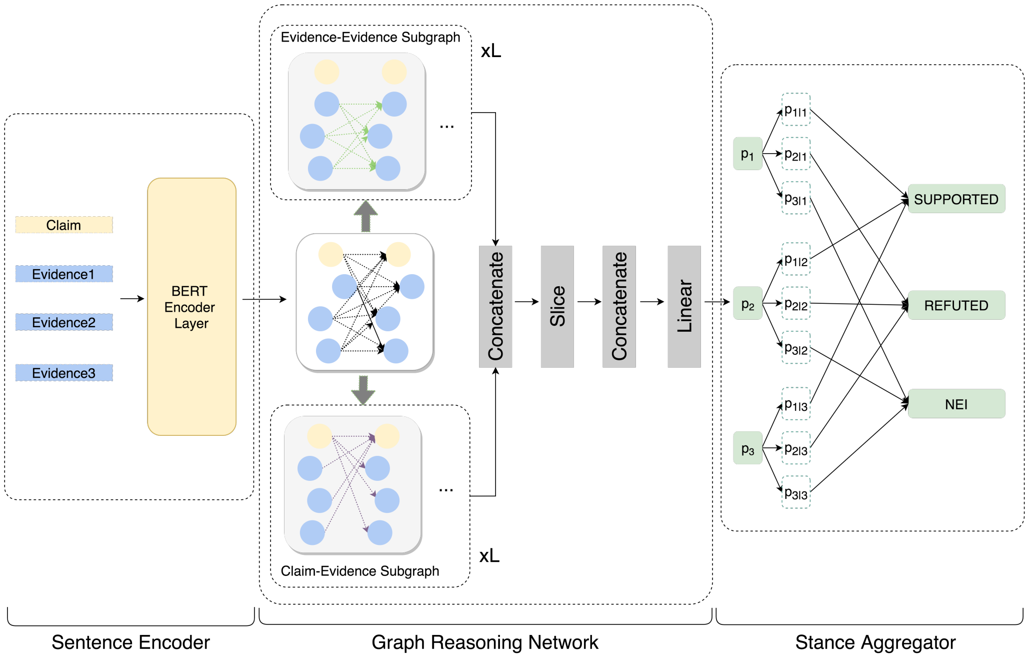


Figure 2: The pipeline of our method. The HGRGA framework is illustrated in the proposed method section.

Then the layer-wise propagation rule is defined as:

$$h_i^{l+1} = ReLU(\sum_{j \in N_i} \alpha_{i,j}^l W^l h_j^l) \tag{4}$$

After that, multi-head attention (Vaswani et al., 2017) is utilized to stabilize the learning process of self-attention and extend attention mechanism. Thus Eq. 4 would be extended to the multi-head attention process of concatenating M attention heads:

$$h_i^{l+1} = \overset{M}{\underset{m=1}{||}} ReLU(\sum_{j \in N_i} \alpha_{i,j}^{l,m} W_m^l h_j^l) \tag{5}$$

where $||$ represents concatenation, $\alpha_{i,j}^{l,m}$ is a normalized attention coefficient computed by the m-th head at the l-th layer, and W_m^l is the corresponding input linear transformation's weight matrix. By stacking L layers of GAT, the output embedding in the final layer is calculated using averaging, instead of the concatenation operation:

$$h_i^L = ReLU(\frac{1}{M} \sum_{m=1}^{M} \sum_{j \in N_i} \alpha_{i,j}^{L-1,m} W_m^{L-1} h_j^{L-1}) \tag{6}$$

Through aforementioned operations, we get the final layer of claim-evidence subgraph result $H_{ce}^L = [h_0^L, h_1^L, h_2^L, ..., h_N^L]^{\mathrm{T}}$.

Evidence-Evidence Subgraph Similarly to the claim-evidence subgraph in Section 3.2, we enhance the semantical coherence of each evidence via GAT method. More concretely, we use $H_{ee}^l = [\tilde{h}_0^l, \tilde{h}_1^l, \tilde{h}_2^l, ..., \tilde{h}_N^l]^{\mathrm{T}}$ to represent the hidden states of nodes at layer l and initially, $H_{ee}^0 = X$. Besides, the relationship between nodes within subgraph is different and we utilize the adjacency matrix A^{ee} to denotes the relationship between each node, which is defined as:

$$A_{i,j}^{ee} = \begin{cases} 1 & i \in \{e_1, ..., e_N\}, \\ & j \in \{e_1, ..., e_N\} \\ 0 & otherwise \end{cases} \tag{7}$$

Finally, the output of evidence-evidence subgraph can be updated via $H_{ee}^L = [\tilde{h}_0^L, \tilde{h}_1^L, \tilde{h}_2^L, ..., \tilde{h}_N^L]^{\mathrm{T}}$

Fusion of Subgraphs To fuse the information contained in two subgraphs, we concatenate H_{ce}^L and H_{ee}^L to form implicit representation of claim and evidences, denoted as H^L . Then, we propose a slice operation to extract claim and evidence feature separately from H^L, denoted as $s_c \in \mathbb{R}^{2d \times 1}$ and $s_e \in \mathbb{R}^{2d \times N}$. Consequently, we tile s_c N times and concatenate them with s_e to construct a new feature matrix as

$$\begin{aligned} \mathbf{s} &= concat(s_c, s_e) \\ \mathbf{p} &= tanh(W_s \mathbf{s} + b_s) \end{aligned} \tag{8}$$

where $W_s \in \mathbb{R}^{d \times 4d}$ and $b_s \in \mathbb{R}^{d \times 1}$ are the weight and bias matrix for dimensionality reduction operation. $\mathbf{p} \in \mathbb{R}^{d \times N}$ denotes the implicit stance of evidences towards final class prediction. The reason we use the concatenation operation is that we think the evidence nodes in the following aggregation process need the information from the claim to guide the routing agreement process among them.

3.3 Stance Aggregator

To model the fine-grained stances of evidences towards class prediction, we incorporate the capsule network (Sabour et al., 2017) into our model. We regard **p** as the primary capsule $p_i|_{i=1}^{N} \in \mathbb{R}^d$, Let $v_k|_{k=1}^{K} \in \mathbb{R}^{d_c}$ denote the high-level class capsules, where K denotes the number of classes and d_c means the dimension of class capsules' representation. The capsule model learns a hierarchy of feature detectors via a routing-by-agreement mechanism, which define the different contributions of stances of evidences towards prediction result.

Dynamic Routing-by-agreement We denote $p_{k|i}$ as the resulting prediction vector of the i-th stance capsule when being recognized as the k-th class:

$$p_{k|i} = \sigma(W_k p_i^T + b_k) \tag{9}$$

where $k \in \{1, 2, ..., K\}$ denotes the class type and $i \in \{1, 2, ..., N\}$. σ is the activation function such as $tanh$. $W_k \in \mathbb{R}^{d_c \times d}$ and $b_k \in \mathbb{R}^{d_c \times 1}$ are the weight and bias matrix for the k-th capsule.

The dynamic routing-by-agreement learns an agreement value $c_{k,i}$ that determines how likely the i-th stance capsule agrees to be routed to the k-th class capsule. $c_{k,i}$ is calculated by the dynamic routing-by-agreement algorithm (Sabour et al., 2017), which is briefly recalled in Algorithm 1.

The algorithm determines the agreement value $c_{k,i}$ between stance capsules and class capsules while learning the class representations v_k in an unsupervised, iterative fashion. c_i is a vector that consists of all $c_{k,i}$ where $k \in K$. $b_{k,i}$ is the logit (initialized as zero) representing the log prior probability that the i-th stance capsule agrees to be routed to the k-th class capsule. During each iteration (Line 4), each class representation v_k is calculated by aggregating all the prediction vectors, weighted by the agreement values $c_{k,i}$ obtained from $b_{k,i}$ (Line 6-7):

$$\begin{aligned} s_k &= \sum_i^N c_{k,i} p_{k|i} \\ v_k &= g(s_k) \end{aligned} \tag{10}$$

Algorithm 1 Dynamic routing-by-aggrement

1: **procedure** DYNAMIC ROUTING($p_{k|i}$, $iter$)
2: for each stance capsule i and class capsule k: $b_{k,i} \leftarrow 0$.
3:
4: **for** $iter$ iterations **do**
5: for all stance capsule i: $c_i \leftarrow softmax(b_i)$
6: for all class capsule k: $s_k \leftarrow \sum_r c_{k,i} p_{k|i}$
7: for all class capsule k: $v_k = squash(s_k)$
8: for all stance capsule i and class capsule k: $b_{k,i} \leftarrow b_{k,i} + p_{k|i} \cdot v_k$
9: **end for**
10: Return v_k
11: **end procedure**

where g is a non-linear squashing function which limits the length of v_k to [0, 1]. Once we updated the class representation v_k during iteration, the logit $b_{k,i}$ becomes larger when the dot product $p_{k|i} \cdot v_k$ is large, which means representation of stance capsule $p_{k|i}$ is more similar to class representation v_k. In our scenario, that is, stance of evidences contributes more to a certain category. Meanwhile, we can observe the fine-grained distributions towards prediction result of different stances.

Max-margin Loss for Class Detection Based on the capsule theory (Sabour et al., 2017), the orientation of the activation vector v_k represents class properties while its length indicates the activation probability. The loss function considers a max-margin loss on each labeled utterance:

$$\begin{aligned} \mathcal{L} = \sum_{k=1}^{K} \{ [\![y = v_k]\!] \cdot max(0, m^+ - ||v_k||)^2 \\ + \lambda [\![y \neq v_k]\!] \cdot max(0, ||v_k|| - m^-)^2 \} \} \end{aligned} \tag{11}$$

where $||v_k||$ is the norm of v_k and $[\![]\!]$ is an indicator function, y is the ground truth label. λ is the weighting coefficient, and m^+ and m^- are margins.

The prediction of the utterance can be easily determined by choosing the activation vector with the largest norm $\hat{y} = \underset{k \in \{1,2,...,K\}}{\arg\max} ||v_k||$

4 Experimental Setting

4.1 Dataset and Evaluation Metrics

We conduct experiments on the dataset FEVER (Thorne et al., 2018a). The dataset consists of 185,455 annotated claims with a set of 5,416,537 Wikipedia documents from the June 2017 Wikipedia dump. We follow the dataset

Split	SUPPORTED	REFUTED	NEI
Train	80,035	29,775	35,639
Dev	6,666	6,666	6,666
Test	6,666	6,666	6,666

Table 1: Statistics of FEVER dataset.

partition from the FEVER Shared Task (Thorne et al., 2018b). Table 1 shows the statistics of the dataset.

We evaluated performance by using the label accuracy (LA) and FEVER score (F-score). LA measures the 3-way classification accuracy of class prediction without considering the retrieved evidence. The F-score reflects the performance of both evidence sentence selection and veracity relation prediction, where a complete set of true evidence sentences is present in the selected sentences, and the claim is correctly labeled.

4.2 Baseline

The baselines include top models during FEVER1.0 task, BERT based models and graph-based models.

Three top models (Athene (Hanselowski et al., 2018b), UNC NLP (Nie et al., 2019), UCL MRG (Yoneda et al., 2018)) in FEVER1.0 shared task are compared in our experiment.

As BERT (Devlin et al., 2018) has achieved promising performance on several NLP tasks, we use BERT-pair, BERT-concat from previous work (Zhou et al., 2019) as our baselines.

Other baselines are following like GEAR (Zhou et al., 2019), KGAT (Liu et al., 2019) and DREAM (Zhong et al., 2019).

4.3 Implementation Details

We employ a three-step pipeline with components for document retrieval, sentence selection and claim verification to solve the task. More details can be found in Appendix A.

5 Experimental Results

In this section, we first present the overall performance of our model HGRGA compared with other approaches. Then we conduct an ablation study to explore the effectiveness of the heterogeneous graph structure and the fine-grained capsule network. Finally, we present a case study to demonstrate the effectiveness of our framework.

Models	FEVER			
	Dev		Test	
	LA	F-score	LA	F-score
UKP Athene	68.49	64.74	65.46	61.58
UCL MRG	69.66	65.41	67.62	62.52
UNC NLP	69.72	66.49	68.21	64.21
BERT(base)	73.51	71.38	70.67	68.50
BERT(large)	74.59	72.42	71.86	69.66
BERT-Pair	73.30	68.90	69.75	65.18
BERT-Concat	73.67	68.89	71.01	65.64
GEAR	74.84	70.69	71.60	67.10
KGAT(BERT base)	78.02	75.88	72.81	69.40
KGAT(BERT large)	77.91	75.86	73.61	70.24
DREAM	79.23	-	**76.85**	70.60
Our Model	**80.67**	**77.54**	74.26	**70.72**

Table 2: Overall performance on the FEVER dataset (%).

5.1 Overall Performance

Table 2 shows the performance of our proposed method versus all the compared methods on FEVER dataset, where the best result of each column is bolded to indicate the significant improvement over all baselines.

As shown in Table 2, in terms of LA, our model significantly outperforms BERT-based models with 80.67% and 74.26% on both development and test sets respectively. It is worth noting that, our approach, which exploits distinct types of relationships between nodes within reasoning graph, outperforms GEAR and KGAT, both of which regard claim-evidence pair as node and ignore different implicit interactions among them. However, in terms of LA, DREAM outperforms our approach with 76.85% on the test set. One possible reason is that DREAM incorporates graph-level semantic structure of evidence obtained by Semantic Role Labeling (SRL) which may contain more external information. Despite this, in terms of FEVER score, which is a kind of more comprehensive metrics, our method outperforms it.

5.2 Ablation Study

Effect of Heterogeneous Graph We observe how the model performs when some critical components are removed. The specific results are shown in Table 3, where H_{ce} represents the node' representation updated via claim-evidence subgraph and H_{ee} denotes the node' representation learned via evidence-evidence subgraph. Besides, $Homo$ denotes the reasoning graph is regarded as the ho-

Models		LA	F-score
Our Model		80.67	77.54
-w/o H_{ce}		75.64	70.32
-w/o H_{ee}		77.68	73.52
$Homo$		78.89	75.93
Aggregation	max	77.33	75.23
	mean	77.54	74.97
	attention	77.92	75.10

Table 3: Ablation analysis in the development set of FEVER.

mogenous graph which ignores different types of relationships between claim and evidence, evidence and evidence. As expected, with the removal of important components, the performance of model gradually decrease, especially when the reasoning graph is trained as the homogeneous structure, the LA score drops by nearly 2%, which also shows the strong effectiveness of heterogeneous graph. We will attempts to explore the effective result of heterogeneous structure in Section 5.2. Besides, it's worth noting that, when H_{ce} is removed, model still has a proper result, where it's investigated in previous study (Hansen et al., 2021) and an important problem is highlighted that whether models for automatic fact verification have the ability of reasoning.

Effect of Capsule Layer We explore the effectiveness of the capsule network aggregation by comparing it with other different aggregation methods, such as mean-aggregator, max-aggregator and attention-aggregator. The mean aggregator performs the element-wise Mean operation among stances' representation while the max aggregator performs the element-wise Max operation. The attention aggregator is followed from Zhou et al. (2019), where the dot-product attention operation is used among evidence representation. As shown in Table 3, we can find that our approach using capsule network performs better than other aggregation methods.

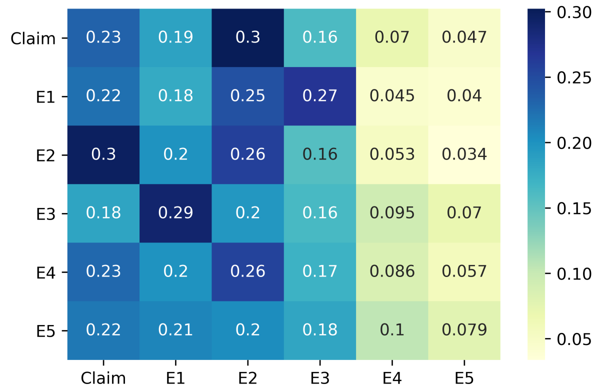
Furthermore, when capsule network is trained, we can easily observe the distribution of stance of evidences towards predicted class during iterations. We will show an example in Section 5.2.

Case Study Table 4 shows an example in our experiments which needs multiple pieces of evidence to make the right inference. There are some noisy evidences such as $E4$-$E5$, which are not semantically coherent with $E1$-$E3$, and a confusing evidence $E2$ which may introduce spurious information and mislead the model to predict the label incorrectly. In order to observe the difference between homogenous graph structure and heterogeneous graph structure, we plot the claim-evidence attention map from the model learned under these two settings.

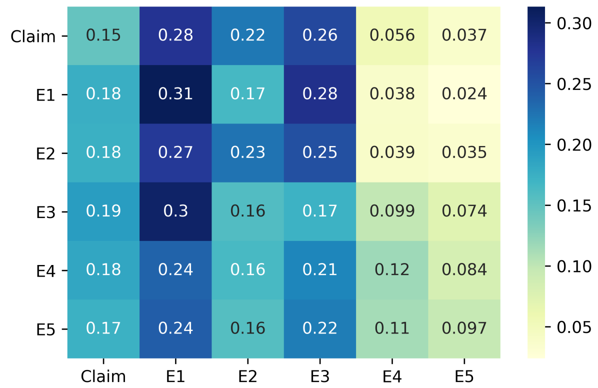
Claim: One ***host*** of ***Weekly Idol*** is ***a comedian***.
Evidence: **E1:** ***The show is hosted by comedian Jeong Hyeong-don*** and rapper Defconn. **E2:** Defconn, ***one host of Weekly Idol, is a rapper*** used to perform several songs on the show. **E3:** ***Weekly Idol is a South Korean variety show***, which airs Wednesdays, 6PM KST, on MBC Every1, MBC's cable and satellite network for comedy and variety shows. **E4:** Many comics achieve a cult following while touring famous comedy hubs such as the Just for Laughs festival in Montreal, the Edinburgh Fringe, and Melbourne Comedy Festival in Australia. **E5:** However, a comic's stand-up success does not guarantee a film's critical or box office success.
Label: SUPPORTED

Table 4: A case of the claim that requires integrating multiple evidence to verify. Facts shared across the claim and the evidences are highlighted with different colors.

As shown in Figure 3a, when the reasoning graph is constructed as homogenous structure, the model would consider the entailment relationship between claim and evidence equally to another relationship, semantic coherence among each evidence. With high similarity between claim and $E2$ on semantic perspective, the proposed method tends to attend $E2$, which leads to a prediction error. In contrast, when the inference relationship between claim and evidence is explicitly exploited, the ability of reasoning would be further enhanced. Making the correct prediction requires model to reason based on the understanding that "*comedian*" is occurred in $E1$ and "*Weekly Idol*" is a show mentioned in $E3$. Based on the observation as illustrated in Figure 3b, our approach pays more attention on $E1$ and $E3$, which provide the most useful information in this case, and the label is correctly detected as *SUPPORTED*.



(a) Homogenous graph structure. Predicted label: *REFUTED*.



(b) Heterogeneous graph structure. Predicted label: *SUPPORTED*.

Figure 3: Attention map of claim-evidence subgraph with different kinds of graph structure for the case in Table 4.

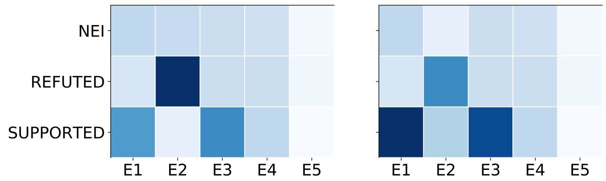


Figure 4: The learned agreement values between class capsules (y-axis) and stance capsules (x-axis) for the case in Table 4. Left: after the first iteration. Right: after the second iteration.

The dynamically learned agreement values within capsule aggregation layer naturally reflect how stance of evidences are collectively aggregated into class capsules for each input utterance. We visualize the agreement values between each stance capsule and each class capsule. The left part of Figure 4 shows that after the first iteration, since the model improperly recognize $E2$ as a whole, the *REFUTED* capsule contribute significantly to the final result. From the right part of Figure 4, we found that with the entailment relationship between claim and evidence being captured in claim-evidence subgraph, evidence $E1$ and $E3$ contribute more to the correct class capsule *SUPPORTED*, which leads to a reasonable result.

6 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 of some phrases on some complex cases. For example, the claim "*Philomena is a film nominated for **seven** awards.*" is supported by the evidence "*It was also nominated for **four** BAFTA Awards and **three** Golden Globe Awards.*" The model needs to understand that four plus three equals seven in this case. Another case is that 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 understand the relationship between *novel* and *book*. Solving this type of problem requires the incorporation of additional knowledge, such as math logic and common sense.

The second type of errors is due to the failure of retrieving relevant evidences. For example, the claim states "*Lyon is a city in Southwest France.*", and the ground-truth evidence states "*Lyon had a population of 506,615 in 2014 and is France's third-largest city after Paris and Marseille.*", which gives not enough information to help model make a true judgement.

7 Consultion

In this work, we present a novel heterogeneous-graph reasoning and fine-grained aggregation framework on the claim verification subtask of FEVER. We propose heterogeneous graph attention network to better exploit different types of relationships between nodes within reasoning graph. Furthermore, the capsule network learned through dynamic routing-by-aggrement is utilized to observe fine-grained distributions of stances towards claim from multiple pieces of evidence. The framework is proven to be effective and our final pipeline achieves significant and explainable performance. In the future, we would like to explore a fine-grained reasoning mechanism within graph and jointly learn evidence selection and claim verification.

References

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

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, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Machine Learning Challenges Workshop*, pages 177–190. Springer.

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*.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587.

Andreas Hanselowski, Avinesh PVS, Benjamin Schiller, Felix Caspelherr, Debanjan Chaudhuri, Christian M Meyer, and Iryna Gurevych. 2018a. A retrospective analysis of the fake news challenge stance detection task. *arXiv preprint arXiv:1806.05180*.

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

Casper Hansen, Christian Hansen, and Lucas Chaves Lima. 2021. Automatic fake news detection: Are models learning to reason? *arXiv preprint arXiv:2105.07698*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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

Zhenghao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. 2019. Fine-grained fact verification with kernel graph attention network. *arXiv preprint arXiv:1910.09796*.

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.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic routing between capsules. *arXiv preprint arXiv:1710.09829*.

Amir Soleimani, Christof Monz, and Marcel Worring. 2020. Bert for evidence retrieval and claim verification. *Advances in Information Retrieval*, 12036:359.

Dominik Stammbach and Guenter Neumann. 2019. Team domlin: Exploiting evidence enhancement for the fever shared task. In *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*, pages 105–109.

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*.

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.

Reza Zafarani, Xinyi Zhou, Kai Shu, and Huan Liu. 2019. Fake news research: Theories, detection strategies, and open problems. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3207–3208.

Chenwei Zhang, Yaliang Li, Nan Du, Wei Fan, and Philip S Yu. 2018. Joint slot filling and intent detection via capsule neural networks. *arXiv preprint arXiv:1812.09471*.

Chen Zhao, Chenyan Xiong, Corby Rosset, Xia Song, Paul Bennett, and Saurabh Tiwary. 2020. Transformer-xh: Multi-evidence reasoning with extra hop attention.

Wei Zhao, Jianbo Ye, Min Yang, Zeyang Lei, Suofei Zhang, and Zhou Zhao. 2018. Investigating capsule networks with dynamic routing for text classification. *arXiv preprint arXiv:1804.00538*.

Wanjun Zhong, Jingjing Xu, Duyu Tang, Zenan Xu, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. 2019. Reasoning over semantic-level graph for fact checking. *arXiv preprint arXiv:1909.03745*.

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. *arXiv preprint arXiv:1908.01843*.

A Implementation Details

In the document retrieval and sentence selection stages, we simply follow the method from Hanselowski et al. (2018b) since their method has the highest score on evidence recall in the former FEVER shared task and we focus on the claim verification task. We describe our implementation details in this section.

Document Retrieval and Sentence Selection We adopt the entity linking approach from Hanselowski et al. (2018b), which uses entities as search queries and find relevant Wikipedia pages through the online MediaWiki API[2]. Then related sentences are selected from retrieval document. We follow the previous method from Zhao et al. (2020) and use BERT as sentence retrieval model. We use the [CLS] hidden state to represent claim and evidence sentence pair. Then a rank layer is trained to rank score via pairwise loss. Sentences with top-5 relevance scores are selected to form the final evidence set in our experiments.

Claim Verification In our HGRGA, we set the batch size to 256, the number of evidences N to 5 and the dimension of features d to 768. The number of class capsules K is 3, the dimension of class capsules d_c is 10. We set the number L of the graph attention layer as 2, and the head number M as 4. The model is trained to minimize the capsule loss (Sabour et al., 2017) using the Adam optimizer (Kingma and Ba, 2014) with an initial learning rate of 3e-5. In the loss function, the down-weighting coefficient λ is 0.5, margins m^+ and m^- are set to 0.8 and 0.2. We use an early stopping strategy on the label accuracy of the validation set, with a patience of 10 epochs.

[2]https://www.mediawiki.org/wiki/API