mathematics

MDPI

Article

Knowledge-Enhanced Graph Attention Network for Fact Verification

Chonghao Chen, Jianming Zheng * and Honghui Chen

Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha 410073, China; chenchonghao@nudt.edu.cn (C.C.); chenhonghui@nudt.edu.cn (H.C.)
* Correspondence: zhengjianming12@nudt.edu.cn

Abstract: Fact verification aims to evaluate the authenticity of a given claim based on the evidence sentences retrieved from Wikipedia articles. Existing works mainly leverage the natural language inference methods to model the semantic interaction of claim and evidence, or further employ the graph structure to capture the relation features between multiple evidences. However, previous methods have limited representation ability in encoding complicated units of claim and evidences, and thus cannot support sophisticated reasoning. In addition, a limited amount of supervisory signals lead to the graph encoder could not distinguish the distinctions of different graph structures and weaken the encoding ability. To address the above issues, we propose a Knowledge-Enhanced Graph Attention network (KEGA) for fact verification, which introduces a knowledge integration module to enhance the representation of claims and evidences by incorporating external knowledge. Moreover, KEGA leverages an auxiliary loss based on contrastive learning to fine-tune the graph attention encoder and learn the discriminative features for the evidence graph. Comprehensive experiments conducted on FEVER, a large-scale benchmark dataset for fact verification, demonstrate the superiority of our proposal in both the multi-evidences and single-evidence scenarios. In addition, our findings show that the background knowledge for words can effectively improve the model performance.

Keywords: fact verification; external knowledge; graph attention network; contrastive learning

Citation: Chen, C.; Zheng, J.; Chen, H. Knowledge-Enhanced Graph Attention Network for Fact Verification. *Mathematics* **2021**, *9*, 1949. https://doi.org/10.3390/math9161949

Academic Editor: Mikhail Goubko

Received: 13 July 2021
Accepted: 12 August 2021
Published: 15 August 2021

1. Introduction

The rapid development of social media allows more individuals to share their opinions and findings. However, it also leads to the fast spread of rumors and fake news, which may further cause public security problems. Hence, automatically identifying the misleading claim becomes a hot issue and attracts a lot of attention in the natural language processing (NLP) community. To achieve this goal, the fact verification task [1,2] is proposed to evaluate the authenticity of a given claim based on the evidence sentences retrieved from external knowledge sources, e.g., Wikipedia. In detail, the authenticity is judged by the labels of "SUPPORT", "REFUTE" or "NOT ENOUGH INFO", which demonstrates the evidences can support/refute the claim, or indicates the claim is not verifiable, respectively.

Intuitively, fact verification can be considered as a variant of a natural language inference (NLI) task [3], which motivates the researchers to employ NLI methods to deal with it. For instance, previous works typically view all evidence sentences as an ensemble and concatenate them with the claim to obtain the overall similarity score [1,4], or compute the individual similarity for each claim–evidence pair and then aggregate them as the final result [5,6]. However, such traditional NLI methods cannot deal with the claims that need multiple evidences to verify, since they fail to model the semantic relations of evidences. For capturing the relation features, some studies [7,8] attempt to employ a graph structure to model the evidence relations, which extract related semantic units of evidences as nodes and build special edges for them.

Although such graph-based models can capture the evidence relation features, they have limited representation ability in encoding complicated units of claim and evidences, and thus cannot support sophisticated reasoning. As shown in Figure 1, to verify the given claim "The Stanford prison experiment was funded by an organization that coordinates, executes, and promotes the science and technology programs of the U.S. Army and Marine Corps", the model not only need to capture the semantic relations of "Stanford prison experiment" and "Office of Naval Research", but also understand that highlighted words "office" and "army" in evidences are synonymous with "organization" and "navy" in the claim, respectively. This suggests that integrating extra background knowledge of semantic units from knowledge bases (KBs), e.g., WordNet, can reduce the difficulty of the model in inferring the relation of claim and evidences. Therefore, we argue that the embeddings of semantic units in claim and evidences should be both context-aware and knowledge-aware. In addition, previous graph-neural-network methods usually leverage a limited amount of labeled samples to learn the representation of evidence graph [9–11], which could not distinguish the distinctions of different graph structures and easily leads to the over-fitting problem of graph encoder.

Claim: The Stanford prison experiment was funded by an **organization** that coordinates, executes, and promotes the science and technology programs of the U.S. **Army** and Marine Corps.

Label: SUPPORT

Evidence: [Stanford prison experiment, 2, It was funded by the U.S. **Office** of Naval Research as an investigation into the causes of difficulties between guards and prisoners in the United States Navy and United States Marine Corps.]

[Office of Naval Research, 0, The Office of Naval Research is within the United States Department of the Navy that coordinates , executes , and promotes the science and technology programs of the U.S. **Navy** and Marine Corps…]

Background knowledge from WordNet:

Office: An administrative unit of government.
Organization: The persons who make up a body for the purpose of administering something.
Army: A military unit that is part of an army.
Navy: An organization of military vessels belonging to a country and available for sea warfare.

Figure 1. An example from FEVER. The format of evidence is [Document name (Wikipedia), line number, evidence content], which denotes the evidence is extracted from the "line number" of article "Document name". Words highlighted in the claim and evidences refer to those that are not easily understood by the model. The underlined words in the evidences and background knowledge are crucial information to verify the claim.

To deal with the above issues, we propose a Knowledge-Enhanced Graph Attention network (KEGA) for fact verification, which incorporates external background knowledge for words, and introduces an auxiliary loss based on contrastive learning to help the graph encoder learn discriminative representations for evidence graphs. In detail, given the claim and retrieved evidences, we first employ the language model BERT [12] to obtain the context-aware embeddings of related tokens. Then, we design a knowledge integration module to retrieve relevant KB concepts from WordNet, which stores the lexical relations between words, and use the pre-trained KB embeddings of these concepts to generate the knowledge-aware token representations. After that, to achieve the goal of evidence reasoning, we construct the relation graph for evidences, which considers the entities as nodes and leverages their co-occurrence relation to build edges. On this basis, we use the graph attention mechanism to encode the relation feature of semantic units in evidences and adopt a mixture aggregator to obtain the graph representation. Finally, we concatenate

the graph representation and context representation of claim–evidence pairs to predict the label distributions of the claim. In particular, besides using the traditional cross-entropy loss for model training, we apply a contrastive loss function based on the self-supervised objective to fine-tune the graph encoder, which adopts a simple dropout mask strategy to generate the positive sample.

We evaluate our proposal on a large-scale benchmark dataset for fact verification, i.e., FEVER [1]. Generally, the experimental results show that KEGA outperforms the competitive state-of-the-art baselines in terms of the label accuracy and FEVER Score on both the multi-evidence and single evidence scenarios. Further, the results of ablation study validate the effectiveness of our designed modules. In addition, our findings show that external knowledge and fine-grained entity information are essential for model performance.

In summary, the contributions of our work can be summarized as:

(1) We design a knowledge-enhanced module to incorporate the background knowledge for words, which enhances the representation of claim and evidences, as well as helps the model understand potential semantics of evidence.

(2) We propose to leverage a graph contrastive loss to fine-tune the graph encoder, which can explore potential supervisory signals between the samples and alleviate the over-fitting problem of graph encoder.

(3) We conduct extensive experiments to validate the effectiveness of KEGA by comparing it with state-of-the-art baselines. Experimental results show our proposal can beat the baselines in terms of label accuracy and FEVER Score.

We summarize related works in Section 2. Furthermore, the technical details of our proposal are introduced in Section 3. We describe the dataset, baselines, and experimental setups in Section 4. Finally, we analyze the experimental results in Section 5 and give the conclusion in Section 6.

2. Related Work

In this section, we give a summarization of the literature closely related to ours. We first describe the general approaches devoted to the fact verification task in Section 2.1. Then, we introduce the knowledge-aware methods and contrastive learning methods for NLP tasks, as well as their relations to our work in Section 2.2.

2.1. Approaches for Fact Verification

As a fiercely-discussed topic in the NLP community, fact verification is a task to evaluate the authenticity of a given claim based on retrieved evidences from trustworthy corpora, e.g., Wikipedia, which can be widely used in downstream knowledge-based applications [13,14]. Most of the existing works for fact verification are devoted to the FEVER dataset [1,2], which consists of 145,449 human-annotated claims. In particular, this task can be divided into two subtasks, i.e., evidence retrieval and claim verification. The former requires the system to identify correct evidence sentences for the claim from Wikipedia articles. Targeted to this goal, existing works usually train a ranking model or classification model by computing the similarity between claim and each sentence in articles [4,5,7,8,15], which presents high recall and F_1 scores.

Due to the great performance of these methods for evidence retrieval, most of the researches recently are focused on the latter task, i.e., claim verification. In general, these approaches can be roughly categorized into the traditional NLI-based methods and graph-based NLI methods. For instance, Thorne et al. [1] introduce the decomposable attention model (DA) [16] to compute the soft-alignment scores between claim and evidences. Hanselowski et al. [5], Yoneda et al. [6] adopt the classical NLI module called enhanced sequence inference model (ESIM) [17] to compute the similarity features for the claim with each evidence sentence, and employ an attention mechanism to aggregate them for prediction. Similarly, as a modified version of ESIM, neural semantic matching network (NSMN) is proposed by [4], which combines extra token-level features to improve the verification accuracy. In addition, Soleimani et al. [15] leverage the pre-trained language

model BERT [12] to obtain the representations of each claim–evidence pair, and then use an multi-layer perception layer to generate the result.

Although these traditional NLI-based methods have greatly promoted the accuracy of claim verification, they could not deal well with the claims that need multiple evidences to verify. To address this problem, Zhou et al. [7] introduce the graph structure to represent the relation of evidence. On this basis, Liu et al. [8] propose to modify the propagation strategy of the evidence graph, which aggregates fine-grained word features from neighbor sentence nodes. Ref. Zhong et al. [9], Chen et al. [10,11,18] design special graph construction strategies for the evidence, which aims to capture the potential relation features. Different from the above works, Hidey and Diab [19], Nie et al. [20] propose to train the evidence retrieval and claim verification stage by an end-to-end architecture. Similarly, Yin and Roth [21] argue that train the components in a multi-task fashion can improve model performance.

Borrowing the merits of previous graph-based methods, we extract the noun phrases from evidences to build the entity graph for reasoning. On this basis, we innovatively incorporate extra background knowledge for the words, which can help improve the accuracy of evidence representation. In addition, we introduce a contrastive loss function to alleviate the over-fitting problem of the graph attention encoder.

2.2. Knowledge-Aware Methods vs. Contrastive Learning Methods for NLP

Recently, many researchers have explored the potential to empower the NLP model by introducing external knowledge. These approaches can be divided into the structural knowledge-aware methods (e.g., knowledge graph) and unstructured knowledge-aware methods (e.g., textual knowledge). In detail, the former extract the nodes [22,23], triples [24,25] or subgraphs [26,27] from existing large-scale knowledge graphs and use them to enhance the text representation. For instance, Feng et al. [27] apply the ConceptNet in the question answer (QA) system and construct graph relevant to the context from KGs, which can hep produce interpretable predictions. Zhong et al. [28] propose to employ pre-trained knowledge graph embeddings to solve the multi-choice QA task. In addition, the success of pre-trained language models [12,29,30] demonstrates the necessity of unstructured knowledge, which directly use large-scale unlabeled text (e.g., Wikipedia) to help the model learn universal representations for words.

As a variant of self-supervised representation learning, contrastive learning aims to explore the potential supervisory signals from the samples for model training, which is widely used in recent NLP tasks for representation learning [31–38]. In detail, the objective of the contrastive loss function is to pull neighbors together and push non-neighbors apart [39,40]. For instance, the representation of a sentence is intuitively close to its corresponding context paragraph in the embedding space, which motivates the proposal of IS-BERT [33]. In addition, Yan et al. [41] compare the common data augmentation ways to generate the negative samples of a given sentence and employ a contrastive loss layer on top of the BERT encoder to obtain high-quality sentence representation. Similarly, Gao et al. [36] propose to use a simple dropout mask to generate the positive pairs, which obviously improves the accuracy of textual similarity tasks compared to other unsupervised methods.

Inspired by the above methods, in our paper, we adopt the pre-trained language model BERT to generate contextual representations of the claim and evidences. Then, we retrieve external knowledge from WordNet and use the pre-trained KB embeddings to produce knowledge-aware token representations. In addition, we use the dropout strategy to generate the positive sample of the original graph structure and introduce an auxiliary loss based on contrastive learning, which can help the graph encoder to learn the discriminative features of different graph structures.

3. Approach

In this section, we describe the technical details of our knowledge-enhanced graph attention network. The overall architecture of KEGA is shown in Figure 2, which mainly

contains five modules: embedding layer, graph construction, knowledge integration, graph attention network, and output layer.

In the following sections, we first describe the problem formulation and notations used in this paper (Section 3.1). Then we introduce the embedding layer and construction way of the entity graph (Section 3.2). After that, we show how to integrate external background knowledge from KBs (Section 3.3). Moreover, we illustrate the applying process of graph attention networks for evidence reasoning (Section 3.4). Finally, we describe the design of the output layer and contrastive loss function (Section 3.5).

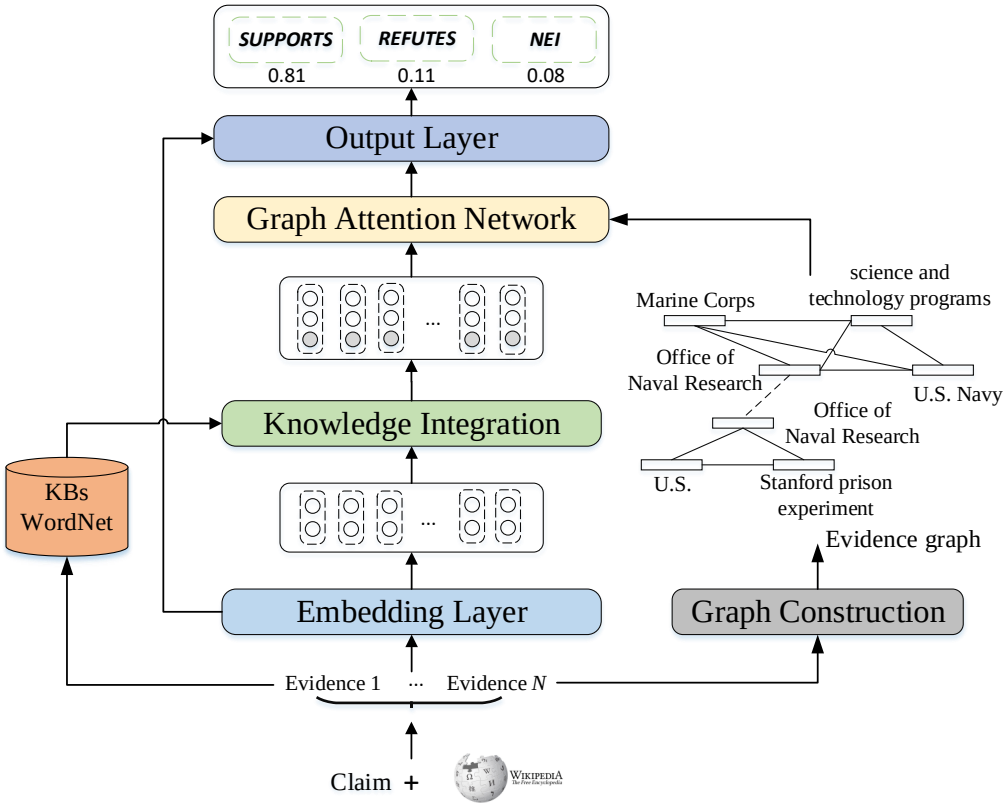


Figure 2. Structure of the KEGA model.

3.1. Problem Definition and Notation

Given a single sentence of unknown authenticity named as claim o and a set of processed Wikipedia articles $A = \{a_1, \ldots, a_{|A|}\}$, fact verification is defined as a multistage task which first identifies suitable sentence-level evidences $S = \{s_1, \ldots, s_N\}$ from Wikipedia and then bases on the retrieved evidences to predict the claim label y, i.e.,

$$\begin{cases} \mathcal{F}_{retrieval}(o, A) \rightarrow S \\ \mathcal{F}_{prediction}(o, S) \rightarrow y \end{cases}, \tag{1}$$

where $y \in \{SUPPORT, REFUTE, NEI\}$.

In this paper, we mainly borrow the merits of previous works for evidence retrieval [5,7,8] and focus on the latter subtask, i.e., claim verification, with a aim to improve the sophisticated reasoning ability of model based on existing evidences. For clarity, we summarize the major notations used in our paper in Table 1.

Table 1. Major notations used in this paper.

Variable	Description
$\mathbf{x}_i$	The representation of the i-th token produced by BERT in the claim–evidence pair x.
$\mathbf{x}_c$	The overall representation of the claim–evidence pair x.
$\mathbf{A}$	An adjacency matrix storing the edge information of entities
$\mathbf{c}_j$	The representation of knowledge concept c_j from WordNet.
$\mathbf{W}_c$,$\mathbf{W}_x$,$\mathbf{W}_k$	Trainable parameter matrics in the knowledge integration module.
$\mathbf{k}_i$	The knowledge state embedding for the token x_i in the claim–evidence pair.
$\mathbf{s}_b$, $\mathbf{r}$, $\mathbf{o}_b$	The representation of subject s_b, relation r and object o_b.
$\mathbf{u}_i$	The knowledge-aware embedding of the token x_i
$\mathbf{u}_c$	The knowledge-aware embedding matrix of the claim tokens.
$\mathbf{u}_e$	The knowledge-aware embedding matrix of the evidence tokens.
$\mathbf{W}_g$, $\mathbf{W}_a$	Trainable parameter matrics in the graph attention network.
$\mathbf{M}_e$	The binary matrix denoting the position relation of entities and tokens.
$\mathbf{e}_i$	The representation of the i-th entity in the entity set E.
$\mathbf{E}_a$	The attention-based pooling result of the updated entity embeddings.
$\mathbf{h}_i$	The hidden output of entity i produced by Bi-LSTM.
$\mathbf{E}_s$	The sequence representation of the entity graph.
$\mathbf{E}_{i,g}^{z_i}$	The graph representations of the i-th sample produced by the dropout mask z.
$\mathbf{W}_0$,$\mathbf{W}_1$,$\mathbf{W}_f$,$\mathbf{b}_f$	Trainable parameter matrics in the output layer.
g_i	The weight to highlight the entity relevant to the local claim.
$P(y)$, $Q(y)$	The distribution of prediction label and ground-truth label.

3.2. Embedding Layer and Graph Construction

In this section, we describe the process of text encoding and entity graph construction. Before getting to the point, we briefly introduce how to obtain relevant evidences for the claim. Following [5,7,8], we first identify the entity mentions from the claim o as search queries and store the top-ranking documents of Wikipedia. Then, we train a ranking model by a modified hinge loss function, which extracts positive and negative pairs from the claim and sentences, respectively. Finally, we employ the trained model to find the top-N ranking sentences from the documents and generate the evidence set as $S = \{s_1, \ldots, s_N\}$.

For text encoding, we use BERT [12] as the backbone network of the embedding layer, which can generate the embedding of each token and the overall representation for the input sentence. In detail, given the claim o and evidence set $S = \{s_1, \ldots, s_N\}$, we first concatenate all the sentences of evidence set as s'. Then we feed the claim and concatenated evidences into BERT and obtain token embeddings of the claim–evidence pair denoted as $\{\mathbf{x}_i\}_{i=1}^{l_1+l_2} \in \mathbb{R}^{d_1}$:

$$\{\mathbf{x}_i\}_{i=1}^{l_1+l_2} = \mathcal{F}_{BERT}([[CLS]; o; [SEP]; s'; [SEP]]), \tag{2}$$

where $[CLS]$ and $[SEP]$ are the identifiers for input sequence, d_1 is the dimension of BERT hidden state, l_1 and l_2 are the length of claim and concatenated-evidence, respectively. In addition, as the byproduct of BERT encoder, the representation of claim–evidence pair can be produced by the $[CLS]$ token and is denoted as $\mathbf{x}_c \in \mathbb{R}^{d_1}$.

To capture the semantic relations of multiple evidences, inspired by [10,42], we construct the evidence graph which takes the entities as nodes and bases on their co-occurrence relations to build edges. In detail, we first extract the entities from evidences by the named entity recognition tool [42] and generate the entity set as $E = \{e_1, \ldots, e_n\}$, which contains main types of noun phrases, e.g., locations, organization, person. To alleviate the computing overhead and improve the transparency of evidence reasoning, we design three types of rules for building edges: (i) Entity nodes from the same evidence sentence should have a connection. (ii) Entity nodes from different sentences referring to the same entity should have a connection. (iii) The central entity node (one node exists in the article title) should have connections with other nodes in the same sentence. In particular, we use an adjacency matrix $\mathbf{A}$ to store the edge information, $\mathbf{A}_{ij} = 1$ denotes there exists an edge from node i to j and otherwise $\mathbf{A}_{ij} = 0$.

3.3. Knowledge Integration

This module takes the claim–evidence pair and related token representations as the input and returns the knowledge-aware token embeddings by incorporating external knowledge, i.e., WordNet. The detailed architecture is shown in Figure 3.

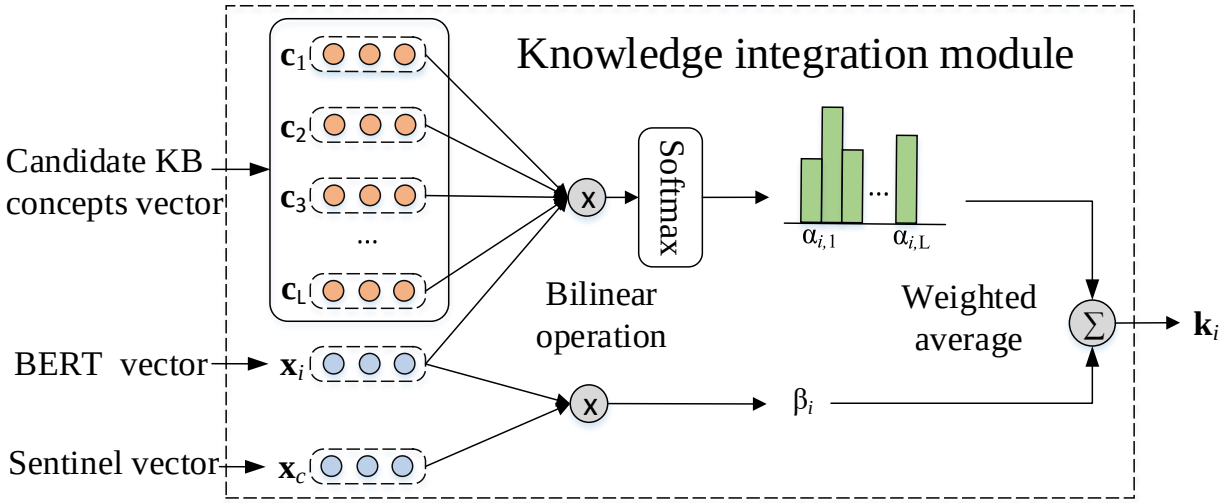


Figure 3. Detailed structure of the knowledge integration module.

In detail, we first train a KB encoder to learn the embeddings of concepts in WordNet. In particular, WordNet stores the lexical relations between word synsets in the form of triples, i.e., $(subject, relation, object)$, we choose to encode them in a continuous vector space instead of directly encoding them as symbolic facts. Inspired by [43], we use a margin-based ranking function to pre-train the KB encoder, where the positive samples are those triples (s_b, r, o_b) already in the WordNet and the negative samples (s'_b, r, o'_b) are those obtained by corrupting either one of the relation arguments. In addition, we leverage a bilinear function to compute the score of samples, and the loss function can be formulated as:

$$\begin{cases} \mathcal{L}(\Omega) = \sum_{(s_b,r,o_b)\in T} \sum_{(s'_b,r,o'_b))\in T'} \max\left\{ Score_{(s_b,r,o_b)} - Score_{(s'_b,r,o'_b)} + 1, 0 \right\} \\ Score_{(s_b,r,o_b)} = \mathbf{s}_b^\top diag(\mathbf{r}) \mathbf{o}_b \end{cases}, \quad (3)$$

where T and T' are the sets of positive and negative samples, $\mathbf{s}_b, \mathbf{r}, \mathbf{o}_b \in \mathbb{R}^{d_2}$ are vectors of subject, relation and object, respectively, d_2 is the size of KB embedding. Furthermore, the $diag(r)$ denotes the diagonal matrix with the main diagonal given by $\mathbf{r}$. In the test phase, we can obtain the representation for each knowledge concept c_j in WordNet denoted as $\mathbf{c}_j \in \mathbb{R}^{d_2}$.

Then, we identify a set of synsets for given tokens x_i as candidate knowledge concepts denoted as $V(x_i)$, and select the most relevant ones adaptively by an attention mechanism. In detail, we use a bilinear operation to compute the attention weight α_{ij} between concept c_j and token x_i as:

$$\alpha_{ij} \propto \exp\left(\mathbf{c}_j^\top \mathbf{W}_c \mathbf{x}_i\right), \quad (4)$$

where $\mathbf{W}_c \in \mathbb{R}^{d_2 \times d_1}$ is a trainable matrix. To avoid the misleading knowledge brought by irrelevant concepts (KBs may store the fact irrelevant to the context), we further introduce a knowledge sentinel which records the context information of the local token. In detail,

we use the representation of claim–evidence pair $\mathbf{x}_c$ as the sentinel vector and compute the attention weight on the local context as :

$$\beta_i \propto \exp\left(\mathbf{x}_c^\top \mathbf{W}_x \mathbf{x}_i\right), \tag{5}$$

where $\mathbf{W}_x \in \mathbb{R}^{d_1 \times d_1}$ is a parameter matrix to be learn. Finally, we can obtain the knowledge state vector $\mathbf{k}_i$ which contains the external KB information relevant to the token x_i as well as local context by:

$$\mathbf{k}_i = \sum_j \alpha_{ij} \mathbf{c}_j + \beta_i \mathbf{W}_k \mathbf{x}_c. \tag{6}$$

Here $\sum_j \alpha_{ij} + \beta_i = 1$ and we set $\mathbf{k}_i = 0$ if there not exists knowledge concepts for token x_i, i.e., $V(x_i) = \emptyset$, $\mathbf{W}_k \in \mathbb{R}^{d_2 \times d_1}$ is the trainable parameter matrix. We concatenate the knowledge vector and original token vector to produce the knowledge-aware embedding of tokens denoted as $\{\mathbf{u}_i\}_{i=1}^{l_1+l_2} \in \mathbb{R}^{d_1+d_2}$, which can be divided into the embedding matrix of claim tokens $\mathbf{u}_c$ and evidence tokens $\mathbf{u}_e$, respectively.

3.4. Graph Attention Networks

In this section, we illustrate the applying process of graph attention networks for the evidence reasoning. Before getting to the point, we need to calculate the initial entity representations based on the given knowledge-aware token embeddings of claim–evidence pair $\{\mathbf{u}_i\}_{i=1}^{l_1+l_2}$. In detail, we first build a binary matrix $\mathbf{M}_e$ to record the relations between tokens and entities, where $\mathbf{M}_e(i,j) = 1$ denotes the j-th token is in the text span of the i-th entity and otherwise $\mathbf{M}_e(i,j) = 0$. Then we retain the entity-related rows of the evidence token embedding matrix by multiply $\mathbf{u}_e$ with $\mathbf{M}_e(i,j)$:

$$\mathbf{u}_e^m = \mathbf{M}_e \odot \mathbf{u}_e, \tag{7}$$

where $\odot$ denotes the element-wise multiplication. After that, we adopt two different pooling layers, i.e., max-pooling and mean-pooling, to fuse the related token representations, respectively. Finally, we concatenate the pooling results and feed them into a one-layer perceptron to generate the entity representations denoted as $\mathbf{E} = \{\mathbf{e}_1, \ldots, \mathbf{e}_n\} \in \mathbb{R}^{d_1 \times n}$:

$$\mathbf{E} = \mathcal{F}_{\mathcal{MLP}}([Maxpool(\mathbf{u}_e^m), Meanpool(\mathbf{u}_e^m)]). \tag{8}$$

After obtaining the initial entity representations, inspired by [10,42], we leverage a selection gate $\{g_i\}_{i=1}^n$ to highlight the entities relevant to the local claim.

$$\begin{cases} g_i = \text{softmax}(\gamma_i) = \frac{\exp(\gamma_j)}{\sum_{k=1}^n \exp(\gamma_k)} \\ \gamma_i = \sigma(\frac{\overline{\mathbf{u}}_c^\top \mathbf{W}_g \mathbf{e}_i}{\sqrt{d_1}}) \end{cases}, \tag{9}$$

where $\overline{\mathbf{u}}_c$ denotes the average embeddings of the claim tokens, $\mathbf{W}_g \in \mathbb{R}^{d_1 \times d_1}$ is the parameter matrix, and σ is the sigmoid activation function. Then we multiply the selection gate with the initial entity representation to obtain 0-th layer entity representations $\mathbf{E}^{(0)}$ by:

$$\mathbf{E}^{(0)} = [g_1 \mathbf{e}_1, \ldots, g_n \mathbf{e}_n]. \tag{10}$$

For evidence graph reasoning, we use the graph attention network to propagate the entity features and model the relations between entities. In detail, we first calculate the attention coefficients $\xi_{i,j}$ between entity i and entity j by:

$$\xi_{i,j}^{(0)} = \text{softmax}(\text{ReLU}(\mathbf{W}_a[\mathcal{F}_{\mathcal{MLP}}(\mathbf{e}_i^{(0)}), \mathcal{F}_{\mathcal{MLP}}(\mathbf{e}_j^{(0)})])), \tag{11}$$

where $\mathbf{W}_a$ is the parameter matrix to be learned. Then, we aggregate the neighbor node embeddings of entity i as its updated representation, which can be formulated as:

$$\mathbf{e}_i^{(1)} = \text{ReLU}\left(\sum_{j \in B_i} \xi_{i,j}^{(0)} \mathbf{e}_j^{(0)}\right), \tag{12}$$

where B_i is the neighbor nodes of entity i. Similarly, we can obtain the entity embeddings after t-layer graph encoding denoted as $\mathbf{E}^{(t)} = \{\mathbf{e}_1^{(t)}, \ldots, \mathbf{e}_n^{(t)}\}$.

3.5. Output Layer

In this section, we introduce the design of the output layer, which takes the representations of updated entities and claim–evidence pair as input and returns the distribution of the prediction label. Given the entity representations after t-layer graph propagation $\mathbf{E}^{(t)} = \{\mathbf{e}_1^{(t)}, \ldots, \mathbf{e}_n^{(t)}\}$, we apply a mixture aggregator to allow for better graph feature extraction. In detail, we first adopt an attention mechanism to extract relevant entity features for the local claim, which can be formulated as:

$$\begin{cases} \mathbf{E}_a = \sum_{i=1}^{n} \delta_i \mathbf{e}_i^{(t)} \\ \delta_i = \mathbf{W}_1\left(\text{ReLU}\left(\mathbf{W}_0\left[\overline{\mathbf{u}}_c, \mathbf{e}_i^{(t)}\right]\right)\right) \end{cases}, \tag{13}$$

where $\mathbf{W}_1$ and $\mathbf{W}_0$ are the parameter matrics to be learned, δ_i denotes the attention weight. Then, to obtain the sequence features of entities, we apply a bi-direction long short-term memory network (Bi-LSTM) [44] to process the entity sequence as follows:

$$\begin{array}{ll} \overrightarrow{h_i^{(t)}} = \overrightarrow{\text{LSTM}}\left(\mathbf{e}_i^{(t)}, \overrightarrow{h_{i-1}^{(t)}}\right) & \left(\overrightarrow{h_i^{(t)}} \in \mathbb{R}^{d_1}\right) \\ \overleftarrow{h_i^{(t)}} = \overleftarrow{\text{LSTM}}\left(\mathbf{e}_i^{(t)}, \overleftarrow{h_{i+1}^{(t)}}\right) & \left(\overleftarrow{h_i^{(t)}} \in \mathbb{R}^{d_1}\right), \end{array} \tag{14}$$

where $i = 1, 2, \ldots, n$, $\overrightarrow{h_i^{(t)}}$ and $\overleftarrow{h_i^{(t)}}$ are the outputs of forward and backward LSTM, respectively. We concatenate them to obtain the hidden output of entity i:

$$\mathbf{h}_i^{(t)} = \left(\overrightarrow{h_i^{(t)}}; \overleftarrow{h_i^{(t)}}\right), \quad \left(\mathbf{h}_i^{(t)} \in \mathbb{R}^{2d_1}, i = 1, 2, \ldots, n\right). \tag{15}$$

After that, we feed the last entity of the sequence into a fully-connected layer to produce the representation of entity sequence $\mathbf{E}_s \in \mathbb{R}^{d_1}$:

$$\mathbf{E}_s = \mathcal{F}_{\mathcal{MLP}}(\mathbf{h}_n^{(t)}). \tag{16}$$

Finally, we concatenate the sequence representation and attention-based pooling result of entity graph to produce the final graph representation $\mathbf{E}_g \in \mathbb{R}^{2d_1}$. Then we concatenate it with the mean-pooling result of knowledge-enhanced token embeddings $\overline{\mathbf{u}}$ and employ a fully connected layer to get the distribution of prediction label denoted as $P(y)$:

$$P(y) = \text{softmax}\left(\text{ReLU}\left(\mathbf{W}_f[\overline{\mathbf{u}}, \mathbf{E}_g] + \mathbf{b}_f\right)\right), \tag{17}$$

where $\mathbf{W}_f \in \mathbb{R}^{p \times (3d_1 + d_2)}$, $\mathbf{b}_f \in \mathbb{R}^{p \times 1}$, p is the number of potential labels. In addition, following [7,8], we adopt the cross-entropy loss function to train our model:

$$\mathcal{L}_{en} = \text{CrossEntropy}(Q(y), P(y)), \tag{18}$$

where $Q(y)$ is the distribution of ground-truth labels.

Based on the above graph representation learning, we further describe another important component of our model, i.e., the applying process of auxiliary loss based on the contrastive learning. In the previous evidence reasoning stage, we use the graph attention network to propagate the entity features, which is usually trained by the given label signals. However, limited label information could not help the graph encoder to capture the distinctions of different graph structures and easily lead to the over-fitting problem.

To address it, inspired by the idea of contrastive learning [36,41], we attempt to introduce external supervisory signals from samples for the graph training, which aims to maximize the agreement of positive graph sample pair and keep the distance for the negative graph pair in the same batch. In detail, given a batch of M samples, we feed the entity set of each sample into the graph encoder twice by applying different dropout masks, i.e., z and z'. We denote the graph representations produced by different dropout masks as $\mathbf{E}_{i,g}^{z_i}$ and $\mathbf{E}_{i,g}^{z_i'}$ and consider the latter as the positive sample for former. In addition, other samples $\{\mathbf{E}_{j,g}^{z_j'}\}_{j=1,j\neq i}^{M}$ in the same batch are regarded as negative ones. In particular, we directly use the dropout mask placed on the fully-connected layer of the graph attention network instead of adding additional dropout. To pull the positive pairs together and push negative samples apart in the embedding space, the training objective becomes:

$$\ell_i = -\log \frac{e^{\text{sim}\left(\mathbf{E}_{i,g}^{z_i}, \mathbf{E}_{i,g}^{z_i'}\right)/\tau}}{\sum_{j=1}^{M} e^{\text{sim}\left(\mathbf{E}_{i,g}^{z_i}, \mathbf{E}_{j,g}^{z_j'}\right)/\tau}}, \tag{19}$$

where $\text{sim}(\cdot)$ indicates the cosine similarity function, τ is the temperature hyperparameter. Then we average the losses of M samples to obtain the contrastive loss $\mathcal{L}_{con}$.

We add up all the losses to jointly train our model, which can be formulated as:

$$\mathcal{L} = \mathcal{L}_{en} + \theta \mathcal{L}_{con}, \tag{20}$$

where θ is the hyperparameter to control the proportion of the contrastive loss.

4. Experimental Setup

To investigate the effectiveness of our proposal, we compare the performance of our KEGA with several baseline models on the public benchmark dataset for fact verification. In this section, we introduce the dataset and evaluation mertics, baselines, research questions as well as experimental setups in detail.

4.1. Dataset and Evaluation Metrics

In this paper, we evaluate our proposal and baselines on FEVER, a large-scale benchmark dataset for fact verification [1]. In detail, it contains 185,445 human-annotated claims labeled as "SUPPORTED", "REFUTED" or "NOT ENOUGH INFO". In particular, each "SUPPORTED" or "REFUTED" claim has a ground-truth evidence set that can be used to evaluate its veracity. The dataset is separated into the training set, development set, as well as blind test set, and the test score can be obtained from the official evaluation system. Table 2 shows the detailed dataset split size for different classes.

For the evaluation metrics, we use the scoring metric in [1] to evaluate the model performance, i.e., Label Accuracy (LA) and FEVER Score. The label accuracy only evaluates the correctness of prediction label, while FEVER Score considers a claim is correctly classified should meet the following conditions: (i) the predicted label is correct and (ii) the retrieved evidences should have at least one complete set of ground-truth evidence sentence.

To further explore the performance of our model in dealing with the claims that need multiple evidences to verify, we filter out the samples of "NOT ENOUGH INFO" label, and then split the development set into an easy dev set and a difficult dev set according to the number of evidences. In detail, the easy and difficult dev set contain 9682 and 3650

samples, respectively. In addition, to investigate the effect brought by the entities and knowledge concepts, we divide the original dev set according to the number of extracted entities and retrieved concepts in a sample, respectively.

Table 2. Statistics of FEVER.

Split	SUPPORTED	REFUTED	NEI
Training	80,035	29,775	35,659
Dev	6666	6666	6666
Test	6666	6666	6666

4.2. Model Summary

We compare the performance of our proposed KEGA with ten comparable baselines, including traditional NLI methods, i.e., Athene, UNC NLP and UCL MRG, BERT-based NLI approaches, i.e., BERT Concat, BERT Pair and SRS, and graph-based models, i.e., GEAR, RoEG, CosG, and HHGN. We list them as follows:

Athene [5]: It proposes to use an ESIM model to retrieve evidences as well as compute the similarity features between claim and each evidence sentence, and leverage an attention mechanism to aggregate features;

UNC NLP [4]: It proposes an NSMN model to retrieve evidences and extract the features of concatenated evidences and claim;

UCL MRG [6]: It is an ESIM-based model which predicts the label distribution for each claim–evidence pair and aggregates their results by an MLP layer;

BERT Concat [7]: It proposes a BERT-based sequence classification model with an ESIM evidence retrieval component, which uses the concatenated evidences and claim as input;

BERT Pair [7]: It is a BERT-based sequence classification model with an ESIM evidence retrieval component, which applies each claim–evidence pair as input;

SR-MRS [45]: It is a BERT-based sequence classification model with a hierarchical retrieval component, which applies the concatenated evidences and claim as input;

GEAR [7]: It is a fully-connected graph-based method with an ESIM retrieval module, which leverages the attention mechanism to aggregate sentence node features.

RoEG [10]: It proposes an entity graph-based evidence representation learning method, which adopts a BERT-based retrieval component to retrieve evidence.

CosG [18]: It is an entity graph-based method with a BERT retrieval component, which introduces contrastive learning tasks to learn the evidence representations.

HHGN [11]: It proposes a heterogeneous graph-based model with a BERT retrieval component, which designs a hierarchical-structure-based evidence reasoning strategy to learn the evidence embeddings.

We also investigate the performance of our proposal with different-layer graph attention network, i.e., KEGA-1-layer, KEGA-2-layer, and KEGA-3-layer.

4.3. Research Question

We focus on the following research questions to guide our experiments:

RQ1 Does KEGA improve the overall performance compared to the baselines for fact verification?

RQ2 The knowledge integration layer, graph attention network, output layer, and applying of contrastive loss, do they really improve the performance of our model?

RQ3 How does the KEGA perform compared to other methods in the single evidence and multi-evidences scenarios?

RQ4 What is the impact on the performance of the number of knowledge concept and entity?

4.4. Experimental Settings

Following [7], we set the size of evidence set as 5, which consists of the top-5 ranking sentences from the Wikipedia article. In the graph construction component, we set the maximal number of extracted entities as 40 [10]. The maximal length of the input sequence for BERT is set as 256. In addition, the dimension of BERT hidden state and the size of KB embedding are set as 768, 100, respectively. The ratio of dropout in the graph attention network is set to 0.2 and the temperature hyperparameter is set to 0.05. The hyperparameter θ is set as 0.25. We use the Adaptive Moment [46] as the optimization and set the initial learning rate as 5×10^{-5}, and the batch size is set as 8.

5. Result and Discussion

5.1. Overall Evaluation

To answer **RQ1**, we investigate the performance of our proposal and baselines, as well as present the results in Table 3. First, we focus on the performance of the baseline methods, as shown in Table 3, the BERT-based models present noticeable improvements over the traditional methods, i.e., Athene, UCL MRG, UNC NLP, in terms of both metrics. This phenomenon demonstrates the superiority of BERT in representation learning compared to the traditional embedding methods. As for the BERT-based method, the graph-based models generally outperform the non-graph models. It can be explained by the fact that the graph model can help to extract the potential relation features between evidences, which can effectively deal with the claims that need multiple evidences to verify. In addition, we can see that the heterogeneous-graph-based model HHGN shows 1–2.21% and 0.66–4.09% improvements against other graph models, i.e., GEAR, RoEG, and CosG, in terms of label accuracy and FEVER Score on the development set. This indicates that the complicated graph structure and fine-grained semantic representations are better at modeling the semantic relation of evidence. However, such complicated graphs also bring high computing overhead, which motivates us to use the entity graph structure similar to RoEG and CosG for the evidence representation learning.

Table 3. Performance (%) of discussed models on the development (dev) set and the bind test set. The results produced by the best baseline and the best performer in each column are underlined and boldfaced, respectively.

Model	Dev		Test	
	LA	**FEVER Score**	**LA**	**FEVER Score**
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 Concat	73.67	68.89	71.01	65.64
BERT Pair	73.30	68.90	69.75	65.18
SR-MRS	75.12	70.18	72.56	67.26
GEAR	74.84	70.69	71.60	67.10
RoEG	75.43	73.24	71.47	67.51
CosG	76.95	74.12	72.37	68.32
HHGN	<u>77.05</u>	<u>74.78</u>	<u>72.62</u>	<u>68.81</u>
KEGA-1-layer	77.10	75.07	72.65	68.92
KEGA-2-layer	**77.32**	**75.15**	**72.71**	**69.01**
KEGA-3-layer	76.91	74.75	72.56	68.54

Next, comparing the three variants of our proposal, i.e., KEGA, we can find that KEGA shows better performance when the layer number of graph attention network is set as 2. This indicates that multi-layer feature propagation can boost the representation learning

of evidence graphs. However, it also can lead to the over-smoothing problem of node features when the number of layers is set too large, i.e., 3-layer. Finally, we concentrate on the performance of our proposal against the baselines. Clearly, the KEGA-2-layer is the best performer among all discussed models. Compared with the best baseline HHGN, which uses a complicated heterogeneous graph for evidence representation, the KEGA-2-layer can achieve 0.27% and 0.37% improvements on the development set in terms of label accuracy and FEVER Score. This demonstrates the superiority of our model, which can leverage a simple entity graph structure to obtain comparable performance. In particular, compared to the models also employing the entity graph, we can find that our KEGA-2-layer shows obvious improvement compared to RoEG and CosG. For instance, the KEGA-2-layer beats RoEG by 1.89% and 1.91% in terms of both metrics on the development set, which can be explained by the fact that our knowledge integration module can introduce accurate background information for the text representation and further support sophisticated evidence reasoning.

5.2. Ablation Study

To answer **RQ2**, we examine the label accuracy and FEVER Score of our proposal on the development set after removing or replacing some fundamental modules of KEGA separately, e.g., knowledge integration, graph attention network, output layer, and applying of the contrastive loss. The ablation results are shown in Table 4. It is worth noting that here we use the KEGA-2-layer.

Table 4. Results in terms of label accuracy and FEVER Score of KEGA without different modules on the development set (%).

Model	LA	FEVER Score
KEGA	77.32	75.15
w/o knowledge integration	75.51 (−1.81%)	73.39 (−1.76%)
w/o graph attention network	76.15 (−1.17%)	74.04 (1.11%)
w/o output layer	76.28 (−1.04%)	74.13 (−1.02%)
w/o contrastive loss	76.72 (−0.60%)	74.58 (−0.57%)

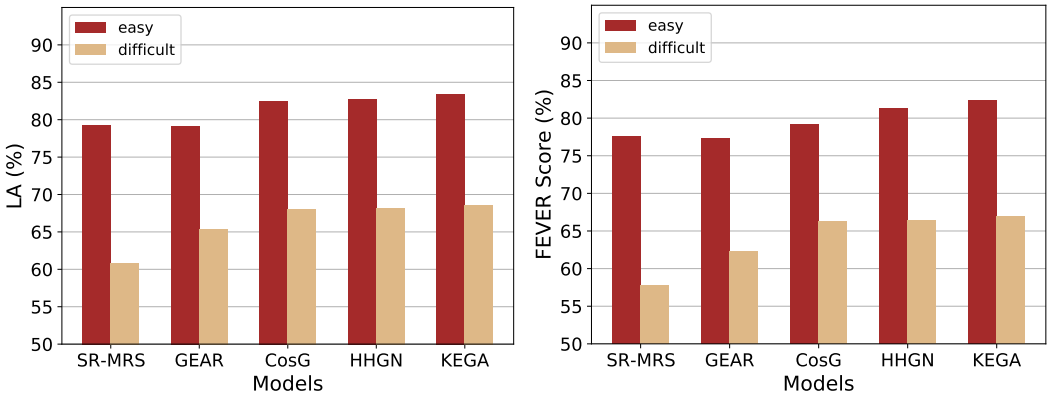
In general, the performances of KEGA have noticeable decreases in terms of label accuracy and FEVER Score when removing or replacing a certain module, which can validate the effectiveness of our designed modules. In detail, we can find that the biggest drop of model performance is brought by removing the knowledge integration module, which leads to 1.81% and 1.76% decreases in terms of label accuracy and FEVER Score. This can be explained by the fact that the external background knowledge for the tokens of the claim and evidences can effectively enrich the text representation, thus helping the model to understand the potential semantic relations and make inferences. In addition, when removing the graph attention network and using the initial entity representation for label prediction, it also presents obvious decreases by 1.17% and 1.11% in terms of both metrics. This further validates the effectiveness of graph structure for evidence representation learning.

Further, we replace the mixture aggregator of the output layer with a simple mean-pooling operation, the performances of KEGA drop by 1.04% and 1.22%, which demonstrates that the mixture aggregator can effectively extract the graph sequence features and relevant entity information for label prediction. To investigate the effectiveness of the contrastive loss function, we remove it and only use the cross-entropy loss for model training, which leads to 0.60% and 0.57% decreases in terms of label accuracy and FEVER Score. This can be explained by the fact that the contrastive loss function can help the

graph encoder to purposefully generate discriminative graph representation and identify different graph structures.

5.3. Model Comparison on Multiple and Single Evidence Scenario

To answer **RQ3**, we compare KEGA with four comparable baseline models, i.e., SR-MRS, GEAR, CosG, and HHGN, on the easy dev set and difficult dev set, respectively. The results are plotted in Figure 4. Generally, as shown in Figure 4, the performances of all models on the easy dev set are obviously higher than the difficult dev set in terms of both metrics. This demonstrates that the claims that need multiple evidences to verify are still the major challenge for the fact verification models. In particular, in terms of label accuracy, as shown in Figure 4a, we can find that the graph-based models present nearly 4.59–7.78% improvements compared to the non-graph model, i.e., SR-MRS, on the difficult dev set. It indicates that the graph model is better at dealing with the multi-evidence scenario. As for the graph-based models, our KEGA beats the best baseline model HHGN by 0.56% and 0.34% on the easy and difficult dev set, respectively. It demonstrates that the knowledge-enhanced token representations can effectively help the model to understand the semantic of evidence and conduct sophisticated evidence reasoning.



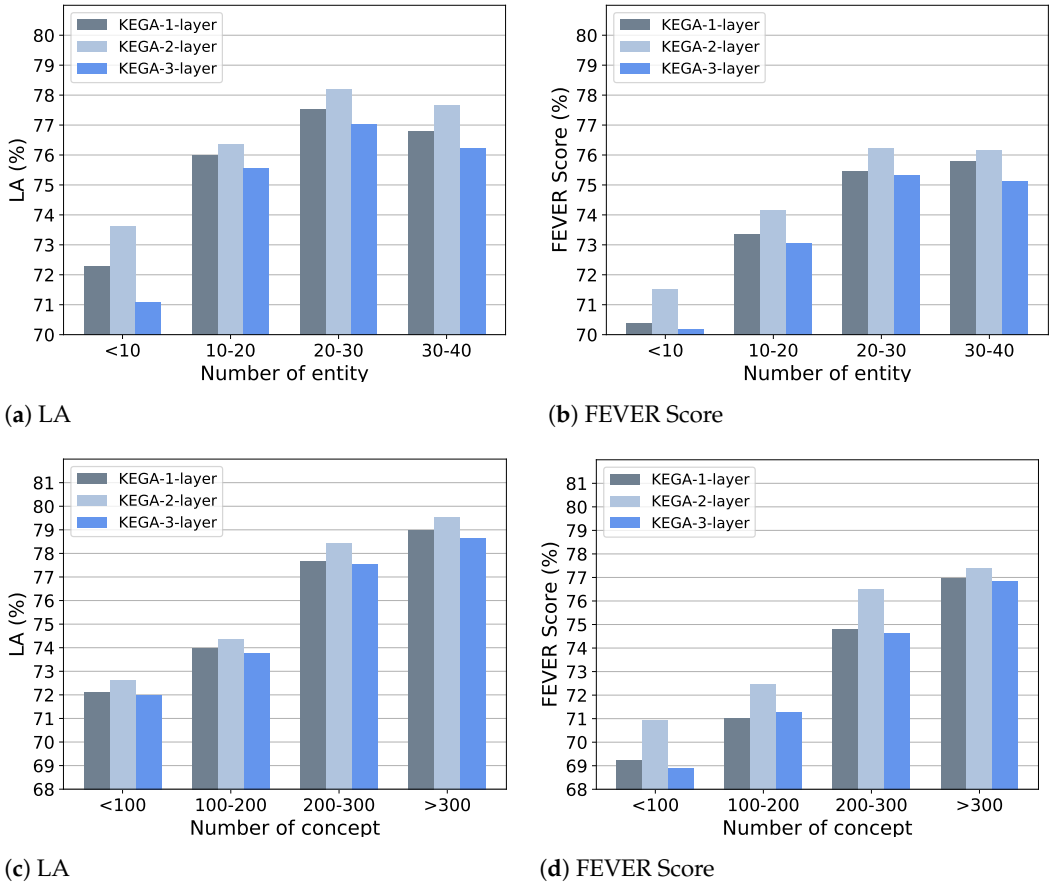
(**a**) LA (**b**) FEVER Score

Figure 4. Performance on easy and difficult dev sets (%).

Similar results can be found in Figure 4b. Our KEGA shows obvious improvements compared to other baselines models in terms of FEVER Score. For instance, KEGA beats the best baseline model HHGN by 1.09% and 0.59% on the easy and difficult set, respectively. This further demonstrates the superiority of our knowledge-enhanced graph attention network method in both multi-evidences and single-evidence scenarios.

5.4. Impact of the Number of Knowledge Concept and Entity

To answer **RQ4**, we investigate the performance of KEGA with different numbers of layers on various groups of samples in the development set, where the groups are divided by the number of entities and knowledge concepts, respectively. The results are plotted in Figure 5. In general, we can find that the KEGA-2-layer outperforms other variants on all groups in terms of label accuracy and FEVER Score, which are consistent with the results shown in Table 3. Next, we focus on the effect of the entity number. As shown in Figure 5a, in terms of label accuracy, we can find that with the increases of the entity number, the performances of all models improve. This indicates that the fine-grained semantic unit, i.e., entity, plays an important role in evidence reasoning. In addition, the performances of all models have slight drops by 0.56–0.79% when the number of entities exceeds 30, which demonstrates that the entities may bring noise when the number is too large. Similar results can be found in Figure 5b.



(**a**) LA (**b**) FEVER Score

(**c**) LA (**d**) FEVER Score

Figure 5. Effect on performance of KEGA variants in terms of label accuracy and FEVER Score (%) with different number of entities and concepts.

For the effect of knowledge concept, as shown in Figure 5c, we can see that the performances of all models increase when the number of concepts increases. In particular, the performances have noticeable improvements when the number of concepts exceeds 200. This further demonstrates the effectiveness of background knowledge for fact verification. In detail, lexical relations for the tokens in claim and evidence can help the encoder to generate more accurate representations for tokens, and thus improve the reasoning ability of the model. Similar results can be found in Figure 5d.

6. Conclusions and Future Work

In this paper, we propose a knowledge-enhanced graph attention network (KEGA) for the task of fact verification, which introduces external background knowledge from KBs to enhance the representations of claim and evidence, as well as leverage an auxiliary loss based on contrastive learning to help the model capture the graph structure features. In detail, using BERT as the backbone network, we first introduce a knowledge integration module to incorporate relevant knowledge concepts from WordNet and use pre-trained KB embeddings to enrich the representation of tokens. Then, we leverage the knowledge-aware token embeddings to generate the initial representation of entities and employ the graph attention network to conduct the graph reasoning. On this basis, we adopt a contrastive loss function to force the graph attention encoder to distinguish different graph structures and alleviate the over-fitting problem. Finally, we aggregate the graph features and context representation for label prediction. Experimental results demonstrate the superiority of our proposal in terms of label accuracy and FEVER Score. In addition, our findings show that entity and knowledge concept are essential for model performance.

As to the limitations of this work, our proposal only works on the specific language environment due to the lack of multilingual knowledge base. In addition, similar to

previous works, our model could not address the claims with contradictory evidence sentences. Thus, we plan to explore the robustness of our model in the multilingual environment by related datasets and appropriate knowledge bases as well as investigate how to improve the accuracy of evidence retrieval in the future.

As for future work, on the one hand, we would like to explore how to improve the explainability and causability of our model based on the graph neural network, e.g., introducing the knowledge graph to represent the relation features of entities, which can provide more transparent evidence reasoning paths for claim verification and be applied to other similar tasks, e.g., automated medical decision [47]. On the other hand, we plan to extend KEGA by mining more potential supervisory signals from the samples and train the model in an unsupervised setting.

Author Contributions: Conceptualization, C.C. and J.Z.; methodology, C.C.; validation, C.C.; data curation, J.Z.; writing—original draft preparation, J.Z.; writing—review and editing, C.C., J.Z., H.C.; visualization, J.Z., H.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Postgraduate Scientific Research Innovation Project of Hunan Province under No. CX20200056.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Thorne, J.; Vlachos, A.; Christodoulopoulos, C. 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, New Orleans, LO, USA, 1–6 June 2018; pp. 809–819.
2. Thorne, J.; Vlachos, A. The Fact Extraction and VERification (FEVER) Shared Task. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language, Brussels, Belgium, 31 October–4 November 2018; pp. 1–9.
3. Bowman, S.R.; Angeli, G.; Potts, C.; Manning, C.D. A large annotated corpus for learning natural language inference. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 632–642.
4. Nie, Y.; Chen, H.; Bansal, M. Combining Fact Extraction and Verification with Neural Semantic Matching Networks. In Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January 27–1 February 2019; pp. 6859–6866.
5. Hanselowski, A.; Zhang, H.; Li, Z. UKP-Athene: Multi-Sentence Textual Entailment for Claim Verification. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 103–108.
6. Yoneda, T.; Mitchell, J.; Welbl, J. UCL Machine Reading Group: Four Factor Framework For Fact Finding (HexaF). In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 97–102.
7. Zhou, J.; Han, X.; Yang, C. GEAR: Graph-based Evidence Aggregating and Reasoning for Fact Verification. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 892–901.
8. Liu, Z.; Xiong, C.; Sun, M.; Liu, Z. Fine-grained Fact Verification with Kernel Graph Attention Network. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 7342–7351.
9. Zhong, W.; Xu, J.; Tang, D. Reasoning Over Semantic-Level Graph for Fact Checking. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 6170–6180.
10. Chen, C.; Cai, F.; Hu, X.; Zheng, J. An entity-graph based reasoning method for fact verification. *Inform. Process. Manag.* **2021**, *58*, 102472. [CrossRef]
11. Chen, C.; Cai, F.; Hu, X.; Chen, W. HHGN: A Hierarchical Reasoning-based Heterogeneous Graph Neural Network for fact verification. *Inf. Process. Manag.* **2021**, *58*, 102659. [CrossRef]
12. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186.
13. Pan, Z.; Chen, W.; Chen, H. Dynamic Graph Learning for Session-Based Recommendation. *Mathematics* **2021**, *9*, 1420. [CrossRef]

14. Chen, W.; Chen, H. Collaborative Co-Attention Network for Session-Based Recommendation. *Mathematics* **2021**, *9*, 1392. [CrossRef]
15. Soleimani, A.; Monz, C.; Worring, M. BERT for Evidence Retrieval and Claim Verification. In Proceedings of the 42nd European Conference on Information Retrieval, Lisbon, Portugal, 14–17 April 2020; pp. 359–366.
16. Parikh, A.; Tackstrom, O.; Das, D.; Uszkoreit, J. A Decomposable Attention Model for Natural Language Inference. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–4 November 2016; pp. 2249–2255.
17. Chen, Q.; Zhu, X.; Ling, Z.H. Enhanced LSTM for Natural Language Inference. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, BC, Canada, 30 July–4 August 2017; pp. 1657–1668.
18. Chen, C.; Zheng, J.; Chen, H. CosG: A Graph-Based Contrastive Learning Method for Fact Verification. *Sensors* **2021**, *21*, 3471. [CrossRef] [PubMed]
19. Hidey, C.; Diab, M. Team SWEEPer: Joint Sentence Extraction and Fact Checking with Pointer Networks. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 150–155.
20. Nie, Y.; Bauer, L.; Bansal, M. Simple Compounded-Label Training for Fact Extraction and Verification. In Proceedings of the Third Workshop on Fact Extraction and VERification (FEVER), Seattle, DC, USA, 9 July 2020; pp. 1–7.
21. Yin, W.; Roth, D. TwoWingOS: A Two-Wing Optimization Strategy for Evidential Claim Verification. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 105–114.
22. Yang, B.; Mitchell, T.M. Leveraging Knowledge Bases in LSTMs for Improving Machine Reading. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, BC, Canada, 30 July–4 August 2017; pp. 1436–1446.
23. Wang, X.; Kapanipathi, P. Improving Natural Language Inference Using External Knowledge in the Science Questions Domain. In Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; pp. 7208–7215.
24. Mihaylov, T.; Frank, A. Knowledgeable Reader: Enhancing Cloze-Style Reading Comprehension with External Commonsense Knowledge. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 15–20 July 2018; pp. 821–832.
25. Kundu, S.; Khot, T.; Sabharwal, A.; Clark, P. Exploiting Explicit Paths for Multi-hop Reading Comprehension. In Proceedings of the 57th Conference of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 2737–2747.
26. Lin, B.Y.; Chen, X.; Chen, J.; Ren, X. KagNet: Knowledge-Aware Graph Networks for Commonsense Reasoning. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, Hong Kong, China, 3–7 November 2019; pp. 2829–2839.
27. Feng, Y.; Chen, X.; Lin, B.Y. Scalable Multi-Hop Relational Reasoning for Knowledge-Aware Question Answering. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, Online, 16–20 November 2020; pp. 1295–1309.
28. Zhong, W.; Tang, D.; Duan, N. Improving Question Answering by Commonsense-Based Pre-training. In *Natural Language Processing and Chinese Computing*; NLPCC 2019; Springer: Dunhuang, China, 2019; Volume 11838, pp. 16–28.
29. Yang, Z.; Dai, Z.; Yang, Y. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In Proceedings of the Annual Conference on Neural Information Processing Systems 2019, Vancouver, BC, Canada, 8–14 December 2019; pp. 5754–5764.
30. Clark, K.; Luong, M.; Le, Q.V.; Manning, C.D. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In Proceedings of 8th International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020.
31. Klein, T.; Nabi, M. Contrastive Self-Supervised Learning for Commonsense Reasoning. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, 5–10 July 2020; pp. 7517–7523.
32. Sun, S.; Gan, Z.; Fang, Y. Contrastive Distillation on Intermediate Representations for Language Model Compression. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, Online, 16–20 November 2020; pp. 498–508.
33. Zhang, Y.; He, R.; Liu, Z. An Unsupervised Sentence Embedding Method by Mutual Information Maximization. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, Online, 16–20 November 2020; pp. 1601–1610.
34. Fang, H.; Xie, P. CERT: Contrastive Self-supervised Learning for Language Understanding. *arXiv* **2020**, arXiv:2005.12766.
35. Carlsson, F.; Gyllensten, A.C.; Gogoulou, E. Semantic Re-tuning with Contrastive Tension. In Proceedings of the 9th International Conference on Learning Representations, Vienna, Austria, 4–8 May 2021.
36. Gao, T.; Yao, X.; Chen, D. SimCSE: Simple Contrastive Learning of Sentence Embeddings. *arXiv* **2021**, arXiv:2104.08821.
37. Wu, Z.; Wang, S.; Gu, J.; Khabsa, M.; Sun, F.; Ma, H. CLEAR: Contrastive Learning for Sentence Representation. *arXiv* **2020**, arXiv:2012.15466.
38. Cui, W.; Zheng, G.; Wang, W. Unsupervised Natural Language Inference via Decoupled Multimodal Contrastive Learning. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, Online, 16–20 November 2020; pp. 5511–5520.
39. Wu, Z.; Xiong, Y.; Yu, S.X.; Lin, D. Unsupervised Feature Learning via Non-Parametric Instance Discrimination. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, Salt Lake City, UT, USA, 18–23 June 2018; pp. 3733–3742.
40. Hjelm, R.D.; Fedorov, A.; Lavoie-Marchildon, S. Learning deep representations by mutual information estimation and maximization. In Proceedings of the 7th International Conference on Learning Representations, Addis Ababa, Ethiopia, 25 September 2019.

41. Yan, Y.; Li, R.; Wang, S.; Zhang, F. ConSERT: A Contrastive Framework for Self-Supervised Sentence Representation Transfer. *arXiv* **2021**, arXiv:2105.11741.
42. Qiu, L.; Xiao, Y.; Qu, Y. Dynamically Fused Graph Network for Multi-hop Reasoning. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 6140–6150.
43. Yang, B.; Yih, W.; He, X. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In Proceedings of the 3rd International Conference on Learning Representations, San Diego, CA, USA, 14 December 2015.
44. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
45. Nie, Y.; Wang, S.; Bansal, M. Revealing the Importance of Semantic Retrieval for Machine Reading at Scale. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, Hong Kong, China, 3–7 November 2019; pp. 2553–2566.
46. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the 3rd International Conference on Learning Representations, San Diego, CA, USA, 14 December 2015.
47. Holzinger, A.; Malle, B.; Saranti, A.; Pfeifer, B. Towards multi-modal causability with Graph Neural Networks enabling information fusion for explainable AI. *Inf. Fusion* **2021**, *71*, 28–37. [CrossRef]