GLAF: Global-to-Local Aggregation and Fission Network for Semantic-Level Fact Verification

Zhiyuan Ma[1], **Jianjun Li**[1]*, **Guohui Li**[1], **Yongjing Cheng**[2]
[1] Huazhong University of Science and Technology (HUST), China
[2] National University of Defense Technology (NUDT), China
{zhiyuanma,jianjunli,guohuili}@hust.edu.cn
davidcheng1001@163.com

Abstract

Accurate fact verification depends on performing fine-grained reasoning over crucial entities by capturing their latent logical relations hidden in multiple evidence clues, which is generally lacking in existing fact verification models. In this work, we propose a novel Global-to-Local Aggregation and Fission network (GLAF) to fill this gap. Instead of treating entire sentences or all semantic elements within them as nodes to construct a coarse-grained or unstructured evidence graph as in previous methods, GLAF constructs a fine-grained and structured evidence graph by parsing the rambling sentences into structural triple-level reasoning clues and regarding them as graph nodes to achieve fine-grained and interpretable evidence graph reasoning. Specifically, to capture latent logical relations between the clues, GLAF first employs a local fission reasoning layer to conduct fine-grained multi-hop reasoning, and then uses a global evidence aggregation layer to achieve information sharing and the interchange of evidence clues for final claim label prediction. Experimental results on the FEVER dataset demonstrate the effectiveness of GLAF, showing that it achieves the state-of-the-art performance by obtaining a 77.62% FEVER score.

1 Introduction

The classic fact verification (FV) task is defined as retrieving relevant sentences as evidence and conducting joint reasoning over these evidence sentences to verify the correctness of a claim, and finally returning a result such as "SUPPORTS", "REFUTES", or "NOT ENOUGH INFO". With the increasingly frequent internet fraud, political rumors, fake news and other false information online, fact verification is becoming more and more important. How to automatically verify the fake claims and prevent their spread is a vital problem.

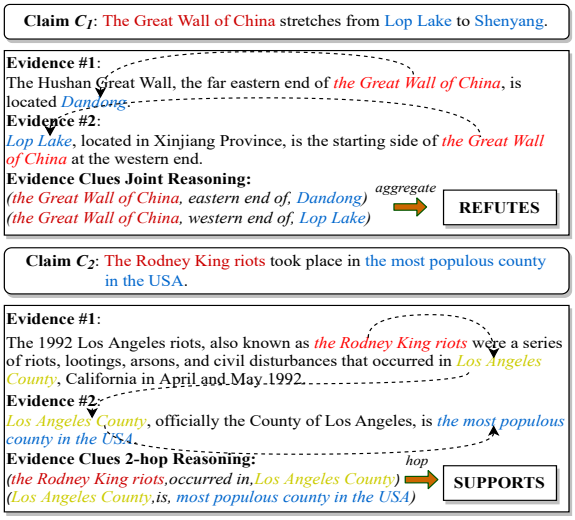


Figure 1: Two motivating examples from FEVER, which requires multi-evidence joint reasoning or multi-hop reasoning to achieve accurate claim label prediction.

In recent years, natural language inference models actually have dominated the study of fact verification (Si et al., 2021; Zhu et al., 2021; Thorne et al., 2018a; Luken et al., 2018; Yin and Roth, 2018; Ye et al., 2020), and many graph augmented neural inference models have been proposed (Zhou et al., 2019; Zhong et al., 2020; Liu et al., 2020; Chen et al., 2021a,b), which first integrate multi-evidence reasoning into fact verification with evidence graph, and then output the claim label prediction result.

Though achieving remarkable progress, existing neural inference FV models still suffer from the following three limitations. **Firstly**, they generally lack the capability of fine-grained evidence clue representing and semantic-level entity reasoning. Most of them either concatenate evidence sentences into a single string (Thorne et al., 2018b), or just treat each evidence-claim pair as a sentence-level node (Zhou et al., 2019; Liu et al., 2020). Since these methods represent and aggregate the evidence at sentence-level, they have difficulty in achieving fine-grained reasoning. Take claim C_1

*Corresponding author.

Proceedings of the 29th International Conference on Computational Linguistics, pages 1801–1812
October 12–17, 2022.

in Figure 1 as an example, the claim states "*The Great Wall of China stretches from Lop Lake to Shenyang*", while the evidence states that the great wall stretches to "*Dandong*" instead of "*Shenyang*". Hence, it requires the FV model to carefully distinguish the subtle differences between truth and false statements. However, existing sentence-level FV models are hard to make such a meticulous discrimination over these crucial entities (e.g., "*Dandong*" and "*Shenyang*"). **Secondly**, prior models generally lack the capability of latent logical relation mining and interpretable claim verifying. As the false claims are often deliberately fabricated, they may be semantically reasonable but logically are not supported. Hence, it requires sufficient logical relation capturing and hop-based reasoning over these clues to guide an interpretable claim judgment. For example, claim C_2 in Figure 1 states "*The Rodney King riots took place in the most populous county in the USA*", while the evidence clues present that *(The Rodney King riots, occurred in, Los Angeles County)* and *(Los Angeles County, is, the most populous county in the USA)*, it requires the FV model to mine the pivot "*Los Angeles County*" and capture the latent relation between "*The Rodney King riots*" and "*the most populous county in the USA*" to make an accurate and convincing judgment by performing multi-hop reasoning. However, existing unstructured FV methods in general cannot support such a triple-level multi-hop reasoning. **Thirdly**, previous models generally lack the noise evidence filtering mechanism. Since the evidence sentences are retrieved from complex background corpora, they will inevitably introduce noises. Even worse, these noises may be magnified in subsequent neural computations, which seriously deteriorates the FV performance.

To tackle these problems, we propose a novel Global-to-Local Aggregation and Fission network (GLAF), which is a graph attention augmented neural inference model for FV. Specifically, to address the first limitation, we first parse the sentences into fine-grained and structural relation triples, each denoted as (s, r, o), and then feed them into the BERT (Devlin et al., 2019) to obtain a set of global evidence clue representations. Next, we introduce a fresh perspective to exploit these structural evidence clue triples. That is, we model each triple (s, r, o) as a map function $f_{\text{clue }r}(s) \rightarrow o$, and use it to conduct entity-level multi-hop reasoning for final claim verification. To address the second limitation, we employ two neural inference layers: local fission reasoning layer and global evidence aggregation layer, to iteratively conduct 2-hop object reasoning and evidence joint reasoning through a triple-level attention mechanism. The two neural layers are utilized to guide the interpretable reasoning process and improve the accuracy of fact verification. Finally, to address the third limitation, we use a graph pooling layer to iteratively select hidden evidence nodes as crucial evidence clues and filter out disruptive noise data, so as to improve the robustness of our FV model.

We conduct experiments on FEVER (Thorne et al., 2018a), which is one of the most influential benchmark datasets for fact verification. We follow the official evaluation protocol of FEVER and demonstrate that GLAF outperforms the recent state-of-the-art baseline systems. Ablation study also shows the effectiveness of each component in improving the performance of fact verification, and a further case study reveals that our model can effectively perform fine-grained multi-hop reasoning over these evidence clues and reach an interpretable conclusion for fact verification.

2 Related Work

2.1 Traditional Fact Verification Models

Many traditional fact verification (FV) systems utilize Natural Language Inference (NLI) techniques (Parikh et al., 2016; Peters et al., 2018; Soleimani et al., 2020) to mine the relationship between evidence and claim to make a final judgment. One of the representative work is the FEVER shared task (Thorne et al., 2018b), which aims to develop an automatic FV system to check the veracity of human-generated claims. Traditional FV models usually employ FEVER's official baseline (Thorne et al., 2018a) with a three-step pipeline: document retrieval, sentence retrieval and claim verification. Among these models, many mainly focus on the last step. For example, Nie et al. (2019) concatenate all evidences together to verify the claim. Yoneda et al. (2018) infer the veracity of each claim-evidence pair and make final prediction by aggregating multiple predicted labels. Hanselowski et al. (2018) encode each claim-evidence pair separately, and use a pooling function to aggregate features for prediction. One of the most widely used models in FEVER is Enhanced Sequential Inference Model (ESIM) proposed by Chen et al. (2017), which has been adopted to select

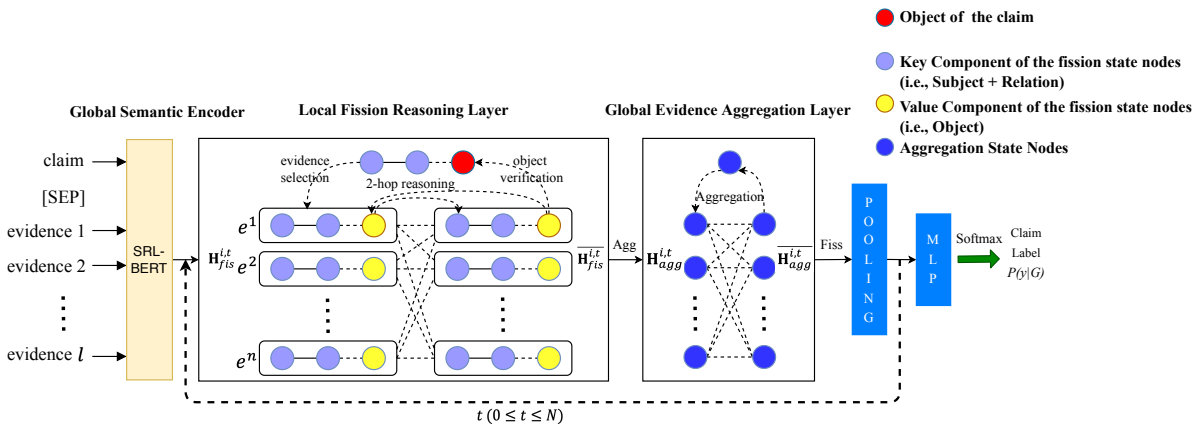


Figure 2: Architecture of GLAF.

relevant sentences in the sentence retrieval phase. Note most of the above mentioned FV methods employ simple models to extract information from evidence, but without letting evidence communicate with each other, which limits their performance. To address this problem, recently there are some good attempts. For example, Si et al. (2021) leverage the LDA model to conduct topic-aware evidence reasoning and stance-aware information aggregation for FV. Wan et al. (2021) employ DQN to find a minimal set of evidence and conduct sentence-level information aggregation for FV. Jiang et al. (2021) improve on previous pointwise aggregation manner by taking advantage of T5 model and explore a listwise-based evidence aggregation method.

2.2 Graph Augmented Fact Verification Models

Though achieving remarkable progress, the above methods in general are difficult to perform global evidence aggregation, since they are not based on graph and hence cannot take the advantage of graph propagation. Recently, by integrating multi-evidence reasoning and global information propagation into fact verification based on a constructed evidence graph, many graph augmented FV models (Zhou et al., 2019; Zhong et al., 2020; Liu et al., 2020; Chen et al., 2020) have been proposed and achieve state-of-the-art results. Among them, **GEAR** (Zhou et al., 2019) is the first to use BERT (Devlin et al., 2019) to encode evidence, and designs a graph network to aggregate information on an evidence graph constructed by treating each evidence as a node. **DREAM** (Zhong et al., 2020) further employs XLNet (Yang et al., 2019) to establish a semantic-level graph for evidence aggregation by using GCN (Kipf and Welling, 2017) and GAT (Velickovic et al., 2018). **KGAT** (Liu et al., 2020) innovatively adopts a kernel graph attention network to aggregate information by unifying the edge kernel mechanism and node kernel mechanism over the evidence graph. However, existing graph augmented FV models either treat entire sentences as sentence-level nodes, e.g., GEAR and KGAT, or extract all semantic elements within them as semantic-level nodes, e.g., DREAM, which makes them only focus on global evidence aggregation, while ignoring reasoning over local semantic clues in triple-level. Different from them, our proposed GLAF is based on a well-structured triple-level evidence graph, which facilitate fine-grained multi-hop reasoning over the crucial entities (i.e., *subject* or *object*) to capture explicit reasoning chains for interpretable claim verification.

3 Model Description

Given a claim c and l evidence sentences, the fact verification task aims to check the veracity of the claim and return a prediction label y, where $y \in \{$"SUPPORTS", "REFUTES", "NOT ENOUGH INFO"$\}$. Instead of treating entire evidence sentences or all semantic elements within them as nodes to construct a coarse-grained or unstructured evidence graph as in previous methods, GLAF constructs a fine-grained and structured evidence graph G by using an off-the-shelf semantic role labeling (SRL) toolkit[1] to parse the l sentences into n structural relation triples[2] and regarding them as graph nodes, denoted by $E = \{e^1, \ldots, e^i, \ldots, e^n\}$. For each evidence node e^i, we use $e^i = (s^i, r^i, o^i)$ to represent a relation triple (*subject*, *relation*, *object*). Then, all

[1] A re-implementation of a BERT-based model by AllenNLP.

[2] Note each sentence could be parsed as multiple triples.

these nodes are connected with edges to obtain a fully-connected evidence graph G with n nodes. Based on G, GLAF produces a prediction probability $P(y|G)$ by reasoning over these evidence triples (nodes) to predict the claim label y. Similar to KGAT, we follow the standard graph label prediction setting in graph neural network (Velickovic et al., 2018) and split the prediction into two components: 1) the evidence selection probability $P\left(e^i|G\right)$; 2) the fine-grained label prediction probability $P\left(y|e^i,G\right)$:

$$P(y|G) = \sum_{i=1}^{n} P\left(e^i|G\right) P\left(y|e^i,G\right) \quad (1)$$

As shown in Figure 2, GLAF mainly includes four modules: Global Semantic Encoder (GSE), Local Fission Reasoning (LFR) Layer, Global Evidence Aggregation (GEA) Layer, and Graph Pooling and MLP Classification Layer. Specifically, GSE is used to obtain initial representations for the claim c and all the nodes in G; LFR, with the initial node representations as inputs, is responsible for conducting 2-hop object reasoning for fine-grained claim verification. LFR outputs updated node representations, which later will be aggregated to serve as the inputs of the GEA layer; GEA is utilized to achieve information sharing by performing 1-order neighborhood information integration; Then, Graph pooling is utilized to filter noise data to select valuable nodes by calculating the evidence selection probability $P\left(e^i|G\right)$; Finally, an MLP layer is used to calculate the fine-grained label prediction probability $P\left(y|e^i,G\right)$.

Note that LFR and GEA each can perform 2-hop reasoning and 1-order neighborhood aggregation, respectively. By iteratively execute LFR and GEA, we can implement more hop reasoning and higher order aggregation to capture sufficient logical relation for more accurate verification. Such an iterative execution process is illustrated in Algorithm 1 (Lines 21-32). We use t to denote the iteration index and assume that $0 \leq t \leq N$, where N represents the total number of iterations and is a model parameter. For convenience, in the following discussion, we use the superscript t to denote the representations or values at the t-th iteration. Next, we detail the separate modules.

3.1 Global Semantic Encoder

Following KGAT, GLAF employs the pretrained language model BERT as the contextual semantic encoder to initialize the global node representations. It is worth noting that since we construct the evidence graph based on triple-level nodes, we first concatenate each triple as `[CLS]` *subject* `[SEP]` *relation* `[SEP]` *object* `[SEP]` and then feed them to BERT to obtain the initial hidden state representation of the node. Specifically, for node e^i, the evidence clue triple is initialized as $\mathbf{H}_{fis}^{i,0}$, where,

$$\mathbf{H}_{fis}^{i,0} = [\mathbf{h}_s^{i,0}; \mathbf{h}_r^{i,0}; \mathbf{h}_o^{i,0}] = \text{BERT}\left([s^i; r^i; o^i]\right) \quad (2)$$

Similarly, the claim node representation is initialized as,

$$[\mathbf{h}_s^c; \mathbf{h}_r^c; \mathbf{h}_o^c] = \text{BERT}\left([s^c; r^c; o^c]\right) \quad (3)$$

3.2 Local Fission Reasoning Layer

In order to encourage the triple nodes to update the object information for fine-grained claim verification, GLAF employs a local fission reasoning layer to conduct 2-hop entity-level reasoning between the target node and its connected nodes, and finally obtains an updated object vector. Different from the previous models that use the whole evidence sentence as a node (Zhou et al., 2019; Liu et al., 2020), we propose to perform entity-level object reasoning. Specifically, given a target node e^i, we update its object vector representation as follows:

- Calculates the cosine similarity between the object (tail-entity) of the target node e^i and the subject (head-entity) of the connected node e^j,

$$M_{fis}^{j\to i,t} = \cos\left(\mathbf{h}_o^{i,t}, \mathbf{h}_s^{j,t}\right) \quad (4)$$

- Obtains the attentive weights by softmax function,

$$\alpha_{fis}^{j\to i,t} = \frac{\exp\left(M_{fis}^{j\to i,t}\right)}{\sum_{k=1}^{n} \exp\left(M_{fis}^{k\to i,t}\right)} \quad (5)$$

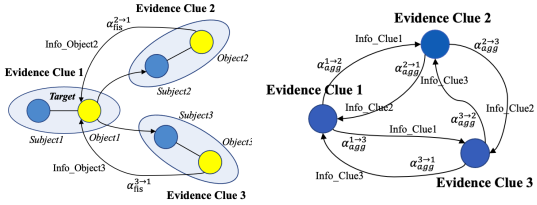
- Calculates the attentive vector corresponding to the specific object after 2-hop reasoning,

$$\mathbf{a}_o^{i,t} = \sum_{j=1}^{n} \alpha_{fis}^{j\to i,t} \mathbf{h}_o^{j,t} \quad (6)$$

- Adaptively updates the object vector representation,

$$\overline{\mathbf{h}_o^{i,t}} = \text{LeakyReLU}\left(\omega_1 \mathbf{h}_o^{i,t} + \omega_2 \mathbf{a}_o^{i,t}\right) \quad (7)$$

where ω_1 and ω_2 are two trainable linear weights, and $\overline{\mathbf{h}_o^{i,t}}$ denotes the updated object vector representation of e^i.



(a) Local fission reasoning (b) Global evidence aggregation

Figure 3: Two kinds of attention-based neural layer.

Consequently, the updated representation of e^i after a 2-hop reasoning can be obtained by,

$$\overline{\mathbf{H}_{fis}^{i,t}} = [\mathbf{h}_s^{i,t}; \mathbf{h}_r^{i,t}; \overline{\mathbf{h}_o^{i,t}}] \tag{8}$$

Function LFR$(\cdot)$ in Algorithm 1 presents the detailed implementation of the LFR layer. Finally, to facilitate the subsequent global information sharing, for each node e^i, we conduct an aggregation operation to convert the fission representation $\overline{\mathbf{H}_{fis}^{i,t}}$ to the aggregation representation $\mathbf{H}_{agg}^{i,t}$ by,

$$\mathbf{H}_{agg}^{i,t} = \mathbf{W}_{agg}^{i,t} \cdot \overline{\mathbf{H}_{fis}^{i,t}} \tag{9}$$

where $\mathbf{W}_{agg}^{i,t}$ is a trainable matrix.

3.3 Global Evidence Aggregation Layer

To achieve information sharing and multi-evidence joint reasoning, GLAF utilizes a global evidence aggregation layer to perform 1-order neighborhood aggregation, which mainly includes three steps:

- Calculates the cosine similarity between the aggregated nodes,

$$M_{agg}^{j \to i,t} = \cos\left(\mathbf{H}_{agg}^{i,t}, \mathbf{H}_{agg}^{j,t}\right) \tag{10}$$

- Obtains the attentive weights by softmax function,

$$\alpha_{agg}^{j \to i,t} = \frac{\exp\left(M_{agg}^{j \to i,t}\right)}{\sum_{k=1}^{n} \exp\left(M_{agg}^{k \to i,t}\right)} \tag{11}$$

- Calculates the attention vector as updated aggregated node representation after integrating the information of the surrounding n nodes (including itself),

$$\overline{\mathbf{H}_{agg}^{i,t}} = \sum_{j=1}^{n} \alpha_{agg}^{j \to i,t} \mathbf{H}_{agg}^{j,t} \tag{12}$$

Function GEA$(\cdot)$ in Algorithm 1 presents the detailed implementation of this layer. Similarly, to facilitate the execution of the LFR layer in the next iteration (if exists), we conduct a fission operation to convert the aggregation representation $\overline{\mathbf{H}_{agg}^{i,t}}$ to the fission representation $\mathbf{H}_{fis}^{i,t}$ by,

$$\mathbf{H}_{fis}^{i,t} = \mathbf{W}_{fis}^{i,t} \cdot \overline{\mathbf{H}_{agg}^{i,t}} \tag{13}$$

where $\mathbf{W}_{fis}^{i,t}$ is a trainable matrix.

3.4 Graph Pooling and MLP Classification Layer

GLAF employs a graph pooling layer to conduct node selection and noise filtering. Specifically, at the t-th iteration, after GEA is executed, the pooling layer discards nodes with few evidence clues and only selects the k (k is a model parameter) most valuable nodes from all of the aggregated

Algorithm 1 GLAF graph learning algorithm

Input: Evidence node set $E = \{e^1, \ldots, e^i, \ldots, e^n\}$ and claim c
Parameter: Number of iterations N and pooling parameter k
Output: Label y

Initialize: global node representations and all the parameters
function LFR(fiss_node_set, t)
  for each fiss_node e^i **do**
    for each fiss_node e^j **do**
      Similarity$(\mathbf{h}_o^{i,t}, \mathbf{h}_s^{j,t}) \to \alpha_{fis}^{j \to i,t}$ (Eqs.4-5).
    end for
    Sum$(\alpha_{fis}^{j \to i,t} \cdot \mathbf{h}_o^{j,t}) \to \mathbf{a}_o^{i,t}$ (Eq.6)
    Weigh $(\mathbf{a}_o^{i,t}, \mathbf{h}_o^{i,t})$ to update $\mathbf{h}_o^{i,t} \to \overline{\mathbf{h}_o^{i,t}}$ (Eq.7)
    Concatenate $(\mathbf{h}_s^{i,t}, \mathbf{h}_r^{i,t}, \overline{\mathbf{h}_o^{i,t}}) \to \overline{\mathbf{H}_{fis}^{i,t}}$ (Eq.8)
  end for
end function
function GEA(agg_node_set, t)
  for each agg_node e^i **do**
    for each agg_node e^j **do**
      Similarity$(\mathbf{H}_{agg}^{i,t}, \mathbf{H}_{agg}^{j,t}) \to \alpha_{agg}^{j \to i,t}$ (Eqs.10-11).
    end for
    Sum$(\alpha_{agg}^{j \to i,t} \cdot \mathbf{H}_{agg}^{j,t}) \to \overline{\mathbf{H}_{agg}^{i,t}}$ (Eq.12)
  end for
end function
Let $t = 0$, $E^0 = E$
while $t \leq N$ **do**
  LFR(E^t,t)
  for each fiss_node e^i in E **do**
    $\overline{\mathbf{H}_{fis}^{i,t}} \xrightarrow{Agg} \mathbf{H}_{agg}^{i,t}$ (Eq.9)
  end for
  GEA(E^t,t)
  for each agg_node e^i in E **do**
    $\overline{\mathbf{H}_{agg}^{i,t}} \xrightarrow{Fiss} \mathbf{H}_{fis}^{i,t}$ (Eq.13)
  end for
  $E^{t+1} = \text{Pooling}\left(E^t, c, k\right)$
  $t = t + 1$
end while
Calculate $P(y|G)$ by Eqs.16-18 for each label
$y = \text{argmax} P(y|G)$
return y

nodes to serve as the readout. Formally,

$$E^{t+1} = \text{Pooling}\left(E^t, c, k\right) \tag{14}$$

where E^{t+1} denotes the evidence node set after pooling on E^t, with $E^0 = E$. Moreover, the value of a node e^i is defined as the semantic similarity between the key vector $[\mathbf{h}_s^c; \mathbf{h}_r^c] \in \mathbb{R}^{1\times 2F}$ (F is the feature dimension) of the claim c and e^i's updated representation $\overline{\mathbf{H}_{agg}^{i,t}}$,

$$M_{\text{pool}}^{i\to c,t} = \cos\left([\mathbf{h}_s^c; \mathbf{h}_r^c] \cdot \mathbf{W}, \overline{\mathbf{H}_{agg}^{i,t}}\right) \tag{15}$$

where $\mathbf{W} \in \mathbb{R}^{2F\times F}$ is a trainable matrix, which is used to align the dimensions of the two vectors. Then, GLAF obtains the evidence selection probability $P\left(e^i|G\right)$ by,

$$P\left(e^i|G\right) = \frac{\exp\left(M_{\text{pool}}^{i\to c,t}\right)}{\sum_{j=1}^{n}\exp\left(M_{\text{pool}}^{j\to c,t}\right)} \tag{16}$$

The k nodes with the highest probability $P\left(e^i|G\right)$ will be selected as the readout.

After N iterations, the representations of the remaining activated nodes are fed into an MLP to conduct claim object verification and generate the fine-grained label prediction probability,

$$P\left(y|e^i, G\right) = \text{softmax}_i(\text{MLP}(\overline{\mathbf{H}_{agg}^{i,t}}, \mathbf{h}_o^c)) \tag{17}$$

Finally, we can obtain the final prediction probability by,

$$P(y|G) = \sum_{i=1}^{k} P\left(e^i|G\right) P\left(y|e^i, G\right) \tag{18}$$

The whole GLAF model is trained end-to-end by minimizing the cross-entropy loss,

$$L = \text{CrossEntropy}\left(y^*, P(y|G)\right) \tag{19}$$

using the ground truth verification label y^*.

4 Experimental Setup

4.1 Datasets and Metrics

We conduct all our experiments on the large-scale dataset FEVER (Thorne et al., 2018a), which consists of 185,455 annotated claims with a set of 5,416,537 Wikipedia documents from the June 2017 Wikipedia dump. We keep the dataset partition the same as the FEVER Shared Task (Thorne et al., 2018a) and TWOWINGOS (Yin and Roth, 2018). Table 1 shows the statistics of the dataset after partition.

Split	Supported	Refuted	Not Enough Info
Train	80,035	29,775	35,639
Dev	3,333	3,333	3,333
Test	3,333	3,333	3,333

Table 1: Statistics of the FEVER dataset.

Following several previous work (Zhou et al., 2019; Zhong et al., 2020; Liu et al., 2020), we use the official evaluation metrics[3] to evaluate the performance of our model on fact verification, which includes Label Accuracy (LA) and FEVER score (FEVER). LA is a general evaluation metric, which calculates claim prediction accuracy rate without considering retrieved evidence. The FEVER score considers whether all evidence included in a golden evidence set are mined, and hence better reflects the reasoning ability.

4.2 Baselines

We compare our model GLAF with the following state-of-the-art baselines.

- **UNC-NLP** (Nie et al., 2019)[4] proposes a neural semantic matching network for claim verification to jointly solve three subtasks by incorporating additional information, such as pageview frequency and WordNet features, for information aggregation.
- **BERT Fine-tuning Systems** (Zhou et al., 2019) includes BERT-Concat and BERT-Pair. The BERT-Concat system concatenates all evidence into a single string while the BERT-Pair system encodes each claim-evidence pair independently and then aggregates the results. For these two BERT fine-tuning systems, we use the source code from (Zhou et al., 2019) and keep the settings unchanged.
- **GEAR** (Zhou et al., 2019)[5] is a graph-based evidence aggregating and reasoning framework by employing an evidence aggregator to aggregate information and conduct evidence reasoning over the evidence graph.
- **DREAM** (Zhong et al., 2020)[6] is built on top of XLNet (Yang et al., 2019) and models evidence graph at a semantic-level by retrieving

[3]https://github.com/sheffieldnlp/fever-scorer
[4]https://github.com/easonnie/combine-FEVER-NSMN
[5]https://github.com/thunlp/GEAR
[6]We reproduce DREAM and try to keep the same settings as the original paper as no open-source code is available.

	Model	Precision	Recall	F1	FEVER
Dev	ESIM	24.08	86.72	37.69	71.70
	BERT	**27.29**	**94.37**	**42.34**	**75.88**
Test	ESIM	23.51	84.66	36.80	68.16
	BERT	**25.21**	**87.47**	**39.14**	**69.40**

Table 2: Results of evidence selection models.

all semantic elements as graph nodes. This model employs a GCN and a GAT network to conduct information aggregation.

- **KGAT** (Liu et al., 2020)[7] models claim-evidence pairs into nodes and adopts a kernel-based graph attention network to conduct evidence aggregating and reasoning.

4.3 Implementation Details

Evidence sentence retrieval We adopt a two-stage scheme to retrieve evidence sentences, which includes document retrieval stage and sentence selection stage. The document retrieval stage retrieves related Wikipedia pages and is kept the same with previous work (Zhou et al., 2019; Liu et al., 2020). At first, it extract all potentially entities included claim as key phrases by using the constituency parser developed by AllenNLP. Then, it regards theses key phrases as queries to search relevant Wikipedia pages through the online MediaWiki API[8], until it searching out convinced article. The sentence selection stage selects relevant sentences from retrieved Wikipedia pages. In our experiments, we try both ESIM-based retrieval model and BERT-based retrieval model. From Table 2, we can see that BERT performs better than ESIM. So, we adopt BERT-based model to retrieve evidence sentences. Specifically, following previous work (Zhou et al., 2019; Liu et al., 2020), we first feed these evidence sentences to a BERT-based ranking model. Then, we use the "`[CLS]`" hidden state to represent claim-evidence pair. Finally, we adopt a pairwise loss to optimize the ranking model for obtaining an optimal evidence retrieval result.

Triple-level clue representation. Similar to previous work (Zhou et al., 2019; Liu et al., 2020), we adopt an identical two-stage scheme to retrieve evidence sentences from background corpus. But different from them, we subsequently adopt a semantic role labeling toolkit to parse each evidence sentence into triple format. Specifically, we built the triples by using the results of the SRL toolkit[9], which includes *(subject, predicate, object)* and *(subject, attributes, value)*. Note that these extracted attributes include time, place, purpose, reason, and other crucial elements that can be mined by SRL. We process all these triples as evidence clues and feed them into BERT to obtain a set of triple-level evidence clue representations. Then, we built an evidence graph by using these triple representations as initial nodes, as described in Section 3.

Model	Dev		Test	
	LA	FEVER	LA	FEVER
UNC-NLP	0.7034	0.6716	0.6858	0.6472
BERT-Concat	0.7399	0.6987	0.7185	0.6718
BERT-Pair	0.7463	0.7008	0.7179	0.6752
KGAT (ESIM)	0.7551	0.7269	0.7348	0.7050
GLAF (ESIM)	**0.7586**	**0.7370**	**0.7441**	**0.7236**
GEAR	0.7601	0.7133	0.7304	0.6815
DREAM ($\text{XLNet}_{\text{Large}}$)	0.7792	0.7235	0.7698	0.7140
KGAT ($\text{BERT}_{\text{Base}}$)	0.7787	0.7575	0.7593	0.7419
GLAF ($\text{BERT}_{\text{Base}}$)	**0.7804**	**0.7635**	**0.7703**	**0.7494**
KGAT ($\text{CorefBERT}_{\text{Base}}$)	0.7798	0.7608	0.7635	0.7441
GLAF ($\text{CorefBERT}_{\text{Base}}$)	**0.7835**	**0.7658**	**0.7760**	**0.7522**
GLAF ($\text{BERT}_{\text{Large}}$)	0.7829	0.7662	0.7784	0.7565
GLAF ($\text{RoBERTa}_{\text{Large}}$)	0.7852	0.7641	0.7905	0.7620
GLAF ($\text{CorefRoBERTa}_{\text{Large}}$)	**0.7941**	**0.7840**	**0.8012**	**0.7762**

Table 3: Overall performance. Note the FEVER score on the blind test set is the main evaluation metric made by FEVER organizers, and all results are statistically significant with $p < 0.05$ under t-test.

Model training details. In our experiments, the batch size is set to 8, learning rate is set to $2e^{-5}$ and warmup proportion is set to 0.1. The max length is set to 140, and the max number of training epochs is set to 6. The maximum number of iterations N is set to 2. BERT and CorefBERT respectively inherit huggingface's implementation[10] and THUNLP's repository[11]. The same as previous work (Zhou et al., 2019; Liu et al., 2020), Adam optimizer is used to optimize all models. All experiments are conducted with PyTorch, and all the source code will be made publicly available upon acceptance. More details about hyper-parameter settings can be found in the Appendix.

5 Evaluation Results

5.1 Overall Performance

The overall performance is shown in Table 3, where the best performance in each scenario is in boldface. It can be observed that, compared with other baselines, GLAF exhibits the best performance

[7] https://github.com/thunlp/KernelGAT
[8] https://www.mediawiki.org/wiki/API:Main_page
[9] https://demo.allennlp.org/semantic-role-labeling
[10] https://github.com/huggingface/pytorch-transformers
[11] https://github.com/thunlp/CorefBERT

Model	FEVER(%)	
	Test	Δ
Complete model	**77.62**	-
w/o SRL retrieval & LFR Layer	72.55	5.07
w/o GEA Layer	74.20	3.42
w/o Graph Pooling Layer	76.35	1.27

Table 4: Ablation study on FEVER test set.

on all testing scenarios. With ESIM sentence retrieval, GLAF outperforms the classic top system UNC-NLP and current best model KGAT on both development and testing sets. With BERT-based sentence retrieval, GLAF outperforms GEAR by almost 10%, DREAM by almost 5% and KGAT by almost 1% test FEVER score. This illustrates the consistent effectiveness of GLAF among graph augmented reasoning models with different sentence retrieval methods. Furthermore, when using $\text{CorefBERT}_{\text{Base}}$, $\text{BERT}_{\text{Large}}$, $\text{RoBERTa}_{\text{Large}}$ and $\text{CorefRoBERTa}_{\text{Large}}$ as the encoder, GLAF achieves even better performance, especially for $\text{CorefRoBERTa}_{\text{Large}}$, it outperforms the current best model KGAT by almost 4.9% in LA metric and 4.3% in FEVER metric on blind test set and achieves the state-of-the-art performance.

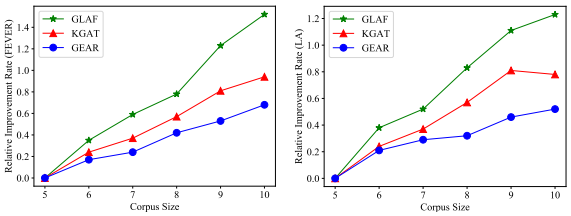
5.2 Ablation Study

In this part, we perform ablation experiments to evaluate the effectiveness of each module and set them accordingly. 1) w/o SRL retrieval & LFR Layer[12] denotes that we remove semantic triple retrieval and local fission reasoning, and just adopt the global aggregation layer to aggregate information; 2) w/o GEA Layer denotes that we remove the global evidence aggregation layer and connect the local fission reasoning layer to the pooling layer directly; 3) w/o Graph Pooling Layer denotes that we remove the graph pooling layer and connect the GEA layer to the MLP layer directly. From the results in Table 4, we can observe that removing each module will result in a performance degradation. In particular, w/o SRL retrieval & LFR Layer and w/o GEA Layer causes 5.07 and 3.42 absolute drops in test FEVER score, respectively, which further verifies the effectiveness of our model.

5.3 Effectiveness Evaluation and analysis

Assessment of evidence mining capability. This experiment evaluates the capability of our model to effectively mine evidence when incremental corpus size is given. Specifically, more

[12]LFR cannot be decoupled with SRL retrieval, since it relies on the triples parsed by SRL. Therefore, we consider them together.



(a) FEVER improvement (b) LA improvement

Figure 4: The assessment of evidence mining capability.

Corpus Size	Number of iterations N			
	0	1	2	3
5	**0.7494**	0.7488	0.7443	-
6	0.7508	**0.7522**	0.7510	-
7	0.7522	**0.7545**	0.7518	-
8	0.7531	**0.7550**	0.7522	-
9	0.7540	**0.7579**	0.7534	-
10	0.7528	**0.7611**	0.7581	-
15	0.7568	0.7608	**0.7620**	0.7588

Table 5: Study of hyperparameter N under different corpus sizes. Best performance under each corpus size is in boldface.

evidence sentences will bring more noise elements, which requires the model to have stronger evidence selection and reasoning ability to carefully distinguish these evidence clues. As shown in Figure 4, we set 5 pieces of evidence as the basic scenario, and vary the corpus size in $[6, 7, 8, 9, 10]$ to test the improvement effect on FEVER and LA. As can be observed, compared with KGAT and GEAR, our GLAF model consistently achieves the best performance on FEVER and LA metrics. We conjecture the reason might be that the pooling layer in GLAF effectively improves the reasoning model by filtering out noise clues.

Study on the number of iterations N. We conduct this experiment to explore the optimal number of iterations N under different corpus sizes. From Table 5, we can observe that with the increase of the corpus size, more iterations are needed to dig out the potential logical relationships hidden among evidence nodes. Specifically, when corpus size $= 5$, the optimal number of iterations is $N = 0$; When corpus size varies in the range of $[6, 7, 8, 9, 10]$, the optimal number of iterations is $N = 1$; When corpus size reaches 15, the optimal number of iterations is $N = 2$. This experiment reveals the relationship between model depth and its performance. Specifically, deeper model may cause the overfitting problem, while shallower models may have difficulty in mining potential advanced features. Therefore, it is important to select the proper

number of iterations.

Corpus Size	With Pooling		Without Pooling	
	LA	FEVER	LA	FEVER
5	**0.7703**	**0.7494**	0.7680	0.7492
7	**0.7761**	**0.7540**	0.7743	0.7526
10	**0.7814**	**0.7609**	0.7752	0.7536
15	**0.8012**	**0.7762**	0.7905	0.7635

Table 6: Effectiveness evaluation of pooling layer.

More evaluation on the pooling layer. We conduct this experiment to further evaluate the effectiveness of the pooling layer. The result is shown in Table 6. In this experiment, we set up a group of comparison models with and without a pooling layer, and set the corpus size within $[5, 7, 10, 15]$. From Table 6, we can observe that the model with the pooling layer achieves better performance than the one without, which demonstrates the effectiveness of the pooling layer in improving the model's reasoning ability over evidence clues through the noise filtering mechanism.

5.4 Case Study

We take the fact verification task in Table 7 as an example, which requires performing triple-level 2-*hop* reasoning over retrieved evidence clues to reach a reliable conclusion. To verify whether "*The Rodney King riots*" took place in "*the most populous county in the USA*", our model mines two crucial evidence clues, (*The Rodney King riots*, *occurred in*, *Los Angeles County*) and (*Los Angeles County*, *is*, *the most populous county in the USA*), to perform attention-based 2-*hop* reasoning. To better understand what our LFR layer has learned, we visualize the attention map from the LFR layer and the final graph pooling layer, as shown in Figure 5. It is clear to see that node 1 achieves the highest value score 0.982 in the last column by integrating information from surrounding nodes, mainly from nodes 2 and 3. Since node 2 is semantically worthless and has the lowest value score 0.125, it will be filtered out by the pooling layer, which implies that $1 \rightarrow 3$ is the optimal 2-*hop* reasoning chain. Finally, the two corresponding evidence clues *(The Rodney King riots, occurred in, Los Angeles County)* and *(Los Angeles County, is, the most populous county in the USA)* can be successfully selected and reasoned to make the final claim verification.

Claim:
The Rodney King riots took place in the most populous county in the USA.
Evidence:
(1) The 1992 Los Angeles riots, also known as the Rodney King riots were a series of riots, lootings, and civil disturbances that occurred in Los Angeles County, California in April and May 1992.
(2) Los Angeles County, officially the County of Los Angeles, is the most populous county in the USA.
Retrieved claim clue:
(The Rodney King riots, took place in, the most populous county in the USA)
Retrieved evidence clues:
① (The Rodney King riots, were, riots_lootings_civil_disturbances)
② (The Rodney King riots, occurred in, Los Angeles County)
③ (Los Angeles County, officially named, the County of Los Angeles)
④ (Los Angeles County, is, the most populous county in the USA)
Label: SUPPORTED

Table 7: A case study illustrating semantic-level 2-*hop* reasoning over fine-grained evidence clues.

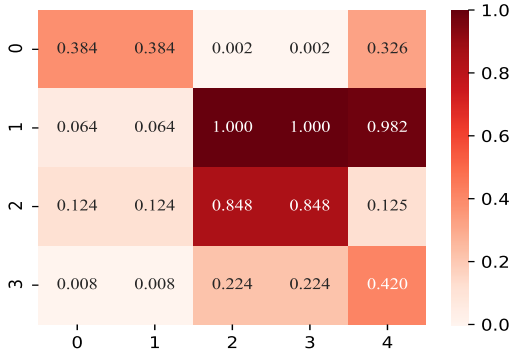


Figure 5: Attention map for the example in Table 7. The first four columns indicate the attention weights α_{fis} from nodes 0 to 3 (corresponding in turn to the four retrieved evidence clues in Table 7) in the LFR layer, and the last column visualizes the value score $M_{\text{pool}}^{i \rightarrow c}$ of the four nodes from the final graph pooling layer.

6 Conclusion

In this paper, we introduce a fresh perspective to revisit the fact verification task and propose a novel Global-to-Local Aggregation and Fission Network (GLAF) to capture latent logical relations hidden in evidence clues for more accurate fact verification. Instead of treating evidence as sentence-level or unstructured representations as in previous work, the proposed GLAF model first parses the evidence sentences as triple-level evidence clues, and then feeds them into contextual language model to obtain global semantic representations. Moreover, to capture latent logical relations between the clues, GLAF respectively employs a local fission layer to conduct fine-grained multi-hop reasoning, as well as a global aggregation layer to conduct interchanging of evidence clues in the graph. Experimental results on the benchmark dataset FEVER have demonstrated the effectiveness and superior performance of our model in both overall evaluation and ablation study.

References

Chonghao Chen, Fei Cai, Xuejun Hu, Wanyu Chen, and Honghui Chen. 2021a. HHGN: A hierarchical reasoning-based heterogeneous graph neural network for fact verification. *Inf. Process. Manag.*, 58(5):102659.

Chonghao Chen, Fei Cai, Xuejun Hu, Jianming Zheng, Yanxiang Ling, and Honghui Chen. 2021b. An entity-graph based reasoning method for fact verification. *Inf. Process. Manag.*, 58(3):102472.

Jiangjie Chen, Qiaoben Bao, Jiaze Chen, Changzhi Sun, Hao Zhou, Yanghua Xiao, and Lei Li. 2020. LOREN: logic enhanced neural reasoning for fact verification. *CoRR*, abs/2012.13577.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced LSTM for natural language inference. In *Proceedings of ACL*, pages 1657–1668.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL*, pages 4171–4186.

Andreas Hanselowski, Hao Zhang, Zile Li, Daniil Sorokin, Benjamin Schiller, Claudia Schulz, and Iryna Gurevych. 2018. Ukp-athene: Multi-sentence textual entailment for claim verification. *EMNLP 2018*, page 103.

Kelvin Jiang, Ronak Pradeep, and Jimmy Lin. 2021. Exploring listwise evidence reasoning with T5 for fact verification. In *Proceedings of ACL 2021*, pages 402–410.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *Proceedings of ICLR*. OpenReview.net.

Zhenghao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. 2020. Fine-grained fact verification with kernel graph attention network. In *Proceedings of ACL*, pages 7342–7351.

Jackson Luken, Nanjiang Jiang, and Marie-Catherine de Marneffe. 2018. Qed: A fact verification system for the fever shared task. In *Proceedings of the First Workshop on FEVER*, pages 156–160.

Yixin Nie, Haonan Chen, and Mohit Bansal. 2019. Combining fact extraction and verification with neural semantic matching networks. In *Proceedings of The 33th AAAI*, pages 6859–6866. AAAI Press.

Ankur P. Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. In *Proceedings of EMNLP*, pages 2249–2255.

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

Jiasheng Si, Deyu Zhou, Tongzhe Li, Xingyu Shi, and Yulan He. 2021. Topic-aware evidence reasoning and stance-aware aggregation for fact verification. In *Proceedings of ACL 2021*, pages 1612–1622.

Amir Soleimani, Christof Monz, and Marcel Worring. 2020. BERT for evidence retrieval and claim verification. In *Proceedings of ECIR*, pages 359–366.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of AAAI*, pages 4444–4451.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018a. FEVER: a large-scale dataset for fact extraction and verification. In *Proceedings of NAACL*, pages 809–819.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018b. The fact extraction and verification (fever) shared task. *EMNLP 2018*, 80(29,775):1.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *Proceedings of ICLR*. OpenReview.net.

Hai Wan, Haicheng Chen, Jianfeng Du, Weilin Luo, and Rongzhen Ye. 2021. A dqn-based approach to finding precise evidences for fact verification. In *Proceedings of ACL 2021*, pages 1030–1039.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Proceedings of NIPS*, pages 5754–5764.

Deming Ye, Yankai Lin, Jiaju Du, Zhenghao Liu, Peng Li, Maosong Sun, and Zhiyuan Liu. 2020. Coreferential reasoning learning for language representation. In *Proceedings of EMNLP 2020*, pages 7170–7186.

Wenpeng Yin and Dan Roth. 2018. Twowingos: A two-wing optimization strategy for evidential claim verification. In *Proceedings of EMNLP*, pages 105–114.

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 FEVER*, pages 97–102.

Wanjun Zhong, Jingjing Xu, Duyu Tang, Zenan Xu, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. 2020. Reasoning over semantic-level graph for fact

checking. In *Proceedings of ACL*, pages 6170–6180.

Jie Zhou, Xu Han, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2019. Gear: Graph-based evidence aggregating and reasoning for fact verification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 892–901.

Biru Zhu, Xingyao Zhang, Ming Gu, and Yangdong Deng. 2021. Knowledge enhanced fact checking and verification. *IEEE ACM TASLP*, pages 3132–3143.

A Appendices

A.1 Hyperparameters Setting

Hyperparameter Name	GLAF
Batch Size	8
Bert Embedding Size	768
Learning Rate	$2e^{-5}$
Warmup Proportion	0.1
Dropout	0.6
Max Epochs	6
Corpus Size	15
Max Length	140
Pooling k	[10,5,3]
Number of iterations N	2

Table 8: Hyperparameters we used for FEVER.

A.2 Error Analysis

To better understand the limitations of our model, we conduct an error analysis on GLAF. We randomly select 200 incorrectly predicted instances that achieve low test FEVER scores. We report several reasons for the low scores, which can roughly be classified into three categories. 1) Upstream document retrieval and sentence selection components extract insufficient evidence for inferring (56%); 2) Incomplete or even incorrect extraction of evidence clues, which may be due to limitations of the SRL toolkit (28%); 3) Lack of common sense knowledge for the claim verification (16%). For example, the claim states "*The Great Wall is a famous ancient building in China*", while the evidence states "*The Great Wall stretches from Lop Lake to Dandong, which is a famous ancient building*". The model fails to realize that "*Lop Lake*" and "*Dandong*" are located in "*China*" due to the lack of common sense knowledge. Solving this type of errors needs to involve external knowledge (e.g., ConceptNet proposed by (Speer et al., 2017)).

A.3 More complicated cases

For more complicated cases, such as a claim sentence contains multiple predicates, GLAF first parses the sentence to multiple triples by the SRL toolkit, and then verifies them separately before making a combined judgment. For example, "Microsoft was founded by Bill Gates and promoted by Tim Cook" can be parsed to two claim triples: (*Microsoft*, *was founded by*, *Bill Gates*) and (*Microsoft*, *was promoted by*, *Tim Cook*), GLAF first predicts their respective labels and then combines these labels to make the final judgment by Table 9.

Priority	Label 1	Label 2	Final Label
Refutes	0	1	1
Not Enough Info	0	0	0
Supports	1	0	0

Table 9: Example of multi-label decisions. Note the priority is in descending order, i.e., Refutes $>$ Not Enough Info $>$ Supports.