Information Sciences 720 (2025) 122505

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Learning to mine all minimal evidences for unverified claims

Yanan Liu [a], Hai Wan [a,*], Jianfeng Du [b,c,**], Yao Wang [a], Kunxun Qi [a], Weilin Luo [a]

[a] School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China
[b] Guangdong University of Foreign Studies, Guangzhou, China
[c] Bigmath Technology, Shenzhen, China

ARTICLE INFO

Keywords:
Information retrieval and text mining
Unverified claims

ABSTRACT

Mining evidences is crucial in checking unverified claims. Most existing methods usually find a single evidence expressed by a set of sentences to verify a given claim. However, treating a set of sentences as unique evidence is insufficient or even misleading, *e.g.*, when it involves both supporting and refuting information. Besides, gathering evidence from different perspectives helps us better analyze and understand the claim. In this article, we suggest mining *all* irreducible evidences for supporting or refuting a claim, where we treat a minimal set of sentences in the given text corpus for either the support or the refutation as an irreducible point of view and call it a *minimal evidence*. We develop a neural-symbolic framework to mine all minimal evidences. It exploits a logical algorithm to compute all minimal evidences one by one, where every minimal evidence is computed through two neural models *scorer* and *reasoner* learnt from the annotated minimal evidences. Experimental results demonstrate that our framework is effective in finding multiple minimal evidences for both textual and structural claims. Furthermore, we investigate several implementation combinations for scorers and reasoners so as to seek the best practice on our framework.

1. Introduction

In recent years, the rapid development of information mediums has led to the dissemination of unverified information around the Web. This has triggered a wave of finding *evidences* to confirm the unverified information. For example, fact verification, aiming to verify the truthfulness of a *claim* by extracting the relevant evidence from a text corpus, has become a crucial task in tackling misinformation spreading through the current Web.

A practical requirement for judging an unverified claim is to retrieve evidences for supporting or refuting the claim, where a piece of evidence is often expressed as a set of sentences and is simply called an *evidence* in this article. A careful way to analyze the truthfulness of a claim should consider different evidences, such as supporting and refuting. Fig. 1 illustrates two such examples. In example 1 of Fig. 1, the claim "Nephi is Lehi's father" needs to be verified, and we can infer the claim through the first sentence. However, there also exists another sentence, *i.e.*, the second sentence, showing that Nephi is a sibling of Lehi. Since the father and sibling relations are disjoint, this claim is both supported and refuted. We cannot draw a conclusion about the claim from a single

* Corresponding author.
** Corresponding author at: Guangdong University of Foreign Studies, Guangzhou, China.
E-mail addresses: liuyn56@mail2.sysu.edu.cn (Y. Liu), wanhai@mail.sysu.edu.cn (H. Wan), jfdu56@gdufs.edu.cn (J. Du), wangy2385@mail2.sysu.edu.cn (Y. Wang), qikx@mail2.sysu.edu.cn (K. Qi), luowlin5@mail.sysu.edu.cn (W. Luo).

https://doi.org/10.1016/j.ins.2025.122505
Received 2 January 2025; Received in revised form 9 July 2025; Accepted 9 July 2025

Available online 14 July 2025

Example 1
Claim: Nephi is Lehi's *father*.
Sentences:
• It is the story of Nephi, his wife Sariah , and their four *sons*: Laman, Lemuel, Sam, and Lehi.
• In the Book of Helaman, after Nephi abdicated the Chief Judgment Seat to Cezoram, he and his brother Lehi went to preach to the Lamanites, who imprison them.
• …

Example 2
Claim: Janet Leigh *was incapable of writing*.
Sentences:
• Janet Leigh -LRB- born Jeanette Helen Morrison;
July 6 , 1927 – October 3 , 2004 -RRB- was an American actress , singer , dancer and *author*.
• Janet Leigh also *wrote* four books between 1984 and 2002 ,
including two novels.
• …

Fig. 1. Examples in reconstructed Wiki80 and FEVER datasets respectively.

evidence. In philosophy and computational logic, conflicting conclusions can be resolved by voting or, more ideally, by argumentation [1]. Furthermore, we can also see that the claim in example 2 of Fig. 1 can be independently supported by both the first sentence and the second sentence, where each sentence expresses a viewpoint on the unverified claim. Therefore, it is crucial to collect multiple or even all evidences for unverified claims.

In the field of fact verification, "retrieve-then-verify" is the dominant paradigm. Given an unverified claim, this paradigm first retrieves a set of sentences as evidence from a text corpus, and then verifies the claim based on the retrieved evidence. Most existing studies focus on the claim verification phase, such as GEAR [2] and LOREN [3]. They usually verify the claim by regarding the sentences extracted from the evidence retrieval phase as a single evidence. The effectiveness of the verification relies on the quality of retrieved sentences. Recently, several methods have integrated evidence retrieval and claim verification into a joint learning framework, by adopting reinforcement learning [4] and ensemble learning [5]. However, these methods still treat the target evidence as a bag of sentences containing one annotated evidence, which may contain conflicting information.

In this article, we argue that drawing the conclusion from one point of view is biased. We focus on the computation of all evidences for a given claim, including both supporting and refuting, which is crucial in supporting decision-making by voting or argumentation. The challenge of mining all evidences lies in determining whether a new evidence exists after a set of evidences has been found. Only when this problem is solved can all evidences be retrieved. To tackle this challenge, we introduce the notion of *minimal evidences*. A minimal evidence for a given claim is a minimal set of sentences (in terms of set-inclusion) in the given text corpus that supports or refutes the claim. It can be treated as an irreducible point of view for the support or the refutation. This notion enables a logically sound solution to the above problem, *i.e.*, a new supporting (*resp.* refuting) minimal evidence does not exist after a set of supporting (*resp.* refuting) minimal evidences has been found, if and only if for every *minimal hitting set* (*MHS*) for the set of found minimal evidences, the complement of this MHS in the given text corpus does not contain a supporting (*resp.* refuting) evidence. A hitting set for a set of evidences is a set of sentences and shares at least one common sentence with each evidence. And a hitting set is said to be *minimal* if none of its proper subsets is also a hitting set.

Based on these new notions, we design a neural-symbolic framework to compute all supporting and refuting minimal evidences for a given claim. This framework conducts an efficient search of minimal evidences by finding new minimal evidences one by one. In the routing of finding a new minimal evidence, a model called *scorer* is used to rank all sentences given to retrieve an evidence, and a model called *reasoner* is used to determine whether the top-rank sentences support or refute the given claim. Both models can be implemented as neural networks, based on a pretrained language model such as RoBERTa [6], trained from annotated minimal evidences in the training set.

To verify the effectiveness of the proposed framework, we have enhanced three existing datasets FEVER [7] from the field of fact verification as well as SemEval [8] and Wiki80 [9] from the field of relation extraction, by constructing supporting or refuting minimal evidences. For FEVER, all claims are textual. For SemEval and Wiki80, all claims are of the triple form. Experimental results on the reconstructed datasets demonstrate that the proposed framework can effectively mine all minimal evidences for support and refutation of given claims. Furthermore, we investigate several implementation combinations for scorers and reasoners, thereby seeking the best practice on our framework. The main contributions of this work can be summarized as follows:

- As far as we know, this work is the first one to target on computing all evidences for unverified claims.
- We propose a neural-symbolic framework to compute all minimal evidences supporting and refuting a given claim, which is sound and complete if finding a minimal evidence can be accurately implemented.
- We enhance three datasets to facilitate empirical evaluation. Extensive experiments conducted on these datasets verify the effectiveness of the proposed framework.

2. Related work

In response to the escalating spread of misinformation, automated fact-checking has attracted mounting interest. A variety of fact verification datasets have been developed for different languages, including CHEF [10] in Chinese, DanFEVER [11] in Danish, and CSFEVER [12] in Czech. In this paper, we focus on FEVER 1.0 shared task [13] in English that aims to develop an automatic fact verification system to determine the truthfulness of a textual claim by extracting related evidences from Wikipedia. Thorne et al.[7] release a large-scale benchmark dataset FEVER and propose a three-stage approach: document retrieval, sentence retrieval and claim verification.

Most existing studies follow the three-stage approach and focus on claim verification. Some studies treat the problem of claim verification as a problem of natural language inference (NLI) and adapt deep learning methods for NLI to claim verification. In more detail, the work [14] presents a connected system consisting of three homogeneous neural semantic matching models. [15] verifies the given claims by utilizing a pretrained sequence-to-sequence transformer T5 [16]. To better capture the interaction between different sentences, some studies construct a graph of selected sentences and treat claim verification as a graph reasoning task, such as DGAT [17] and DREAM [18]. KGAT [19] employs a kernel graph attention network to capture fine-grained information over evidences for more accurate claim verification. And some others build a hierarchy of selected sentences for level-wise inference [20]. MLA [21] is trained on sequence reasoning with multi-level attentions. Additionally, some methods introduce external knowledge such as commonsense knowledge [22] and structural knowledge [23] to constrain the reasoning process. More recently, some researches introduce logic theory to help verify the given claim. Specifically, LOREN [3] is regularized by aggregation logical rules and ProoFVer [24] introduces natural logic inference for claim verification. Since most above methods work on a fixed set of sentences retrieved by the second stage, they only result in a single and probably imprecise evidence for supporting or refuting the given claim.

Although substantial advancements have been made in the claim verification phase, most existing research has not focused on evidence retrieval. The discovery of evidence can directly help us better analyze and understand the unverified claims. Recently, some methods have combined evidence retrieval and claim verification for joint learning [25]. Most of the above methods still regard a bag of sentences as an evidence, which may mislead the judgement. It is worth noting that DQN [4] has proposed to compute *precise evidences*, which represent minimal sets of sentences that can determine the truthfulness of the claim, whether support or refutation. While precise evidences effectively eliminate the distraction of irrelevant sentences, they do not account for the possibility of a claim being simultaneously supported and refuted.

Existing methods for evidence retrieval usually select sentences based on feature engineering [26], neural ranking [20], and pre-trained models [15]. All of those extract a set of sentences as an evidence to the claim verification phase. Recently, GERE [27] introduces a generative evidence retrieval mechanism to find a more precise evidence. However, GERE still focuses on generating one point of view for supporting or refuting the claim.

The aforementioned methods rely solely on a single collection of sentences to evaluate the unverified claim, overlooking the fact that there may be various perspectives. In contrast, our work introduces the notion of minimal evidence for a given claim, which is a minimal set of sentences in the given text corpus that supports or refutes the claim. By observing that divergent perspectives may arise in the process of verifying the same claim, this work addresses a more challenging yet practical issue: identifying all supporting minimal evidences and refuting minimal evidences for a given claim.

3. Approach

To analyze an unverified claim effectively, it is crucial to find all irreducible points of view for supporting or refuting the claim. To this end, we propose a neural-symbolic framework to mine all minimal evidences. We present the below type of evidence to represent different views of support or refutation.

Definition 1. Given a text corpus Π composed of sentences, a claim c, a label $t \in \{\text{T, F}\}$ and a monotonic function $f(S, c, t)$, for $S \subseteq \Pi$, such that $f(S, c, \text{T}) = \text{TRUE}$ if and only if there exists $S' \subseteq S$ supporting c, $f(S, c, \text{F}) = \text{TRUE}$ if and only if there exists $S' \subseteq S$ refuting c, a *t-minimal evidence* E for c in Π *w.r.t.* f is a subset of Π such that $f(E, c, t) =$ TRUE and there exists no proper subset E' of E fulfilling $f(E', c, t) = \text{TRUE}$.

Throughout the paper, a T-minimal evidence is also called a *supporting minimal evidence*, whereas an F-minimal evidence is also said to be a *refuting minimal evidence*. In Definition 1, $f(S, c, t)$ is said to be *monotonic* if $f(S, c, t) = \text{TRUE}$ implies $f(S', c, t) = \text{TRUE}$ for all supersets S' of S. According to this definition, monotonic function $f(S, c, \text{T})$ can detect whether there exists a T-minimal evidence in S. Given a claim c, a set of sentences $\Pi = \{s_1, s_2, \cdots\}$, a true label $t \in \{\text{T, F}\}$, where T and F respectively denotes "SUPPORTS" and "REFUTES", we target on finding all t-minimal evidences for the given claim c from the textual corpus Π.

3.1. Mining all minimal evidences

We propose a neural-symbolic framework *MAME* to mine all minimal evidences for unverified claims. All t-minimal evidences for the given claim c are calculated one by one in a logical way. The logical algorithm for mining all minimal evidences is shown in Algorithm 1. The algorithm takes the depth-first strategy to calculate all t-minimal evidences based on minimal hitting sets defined below.

Algorithm 1 FINDALLMINEVIDENCES(c, t, Π, f).

Input: A claim c, a true label t for c, a set of sentences Π, and a monotonic function f introduced in Definition 1.
Output: The set of t-minimal evidences for c in Π *w.r.t.* f.

```
n ← 0; 𝔼 ← ∅
ConstructMinEvidence(0, ∅, c, t, f)
return 𝔼
procedure ConstructMinEvidence(i, M, c, t, f)
Input: A minimal hitting set M for E_0, ···, E_{i−1}
    if i = n then
        if f(Π \ M, c, t) = TRUE then
            E_n ← FindMinEvidence(∅, Π \ M, c, t, f)
            𝔼 ← 𝔼 ∪ {E_n}; n ← n + 1
        else
            return
        end if
    end if
    if E_i ∩ M ≠ ∅ then
        ConstructMinEvidence(i + 1, M, c, t, f)
    else
        for each s ∈ E_i such that M ∪ {s} is a minimal hitting set for E_0, ···, E_i do
            ConstructMinEvidence(i + 1, M ∪ {s}, c, t, f)
        end for
    end if
end procedure
function FindMinEvidence(S_f, S_d, c, t, f)
Input: A fixed set of sentences S_f, and a changeable set of sentences S_d.
    if S_d = ∅ or f(S_f, c, t) = TRUE then
        return ∅
    else if |S_d| = 1 then
        return S_d
    else
        Divide S_d into two sets S_1 and S_2 with equal cardinality as possible
        E_2 ← FindMinEvidece(S_f ∪ S_1, S_2, c, t, f)
        E_1 ← FindMinEvidece(S_f ∪ E_2, S_1, c, t, f)
        return E_1 ∪ E_2
    end if
end function
```

Definition 2. Let $\mathbb{S}$ be a set of sets. A set H is called a *hitting set* for $\mathbb{S}$ if $H \cap S \neq \emptyset$ for all $S \in \mathbb{S}$. A hitting set for $\mathbb{S}$ is said to be *minimal* if there exists no other hitting set H' for $\mathbb{S}$ such that $H' \subset H$.

Two global variables n and $\mathbb{E}$ are introduced in Algorithm 1, where $\mathbb{E}$ records all currently computed t-minimal evidences and n represents the number of elements in $\mathbb{E}$. Algorithm 1 generates all t-minimal evidences for c in Π by invoking a recursive procedure CONSTRUCTMINEVIDENCE(i, M, c, t, f) where the variable i denotes the level of algorithm and M is a minimal hitting set for $\{E_0, E_1, \cdots, E_{i-1}\}$. By means of FINDMINEVIDENCE, CON BSTRUCTMINEVIDENCE appends a new minimal evidence into $\mathbb{E}$ and updates n. At each level i, it calculates the *minimal hitting set* M for $\{E_0, E_1, \cdots, E_i\}$. The function FINDMINEVIDENCE(S_f, S_d, c, t, f) returns a t-minimal evidence for claim c. It exploits a divide-and-conquer approach [28] to add sentences from S_d into S_f by keeping the resulting set S' enlarged from S_f to satisfy $f(S', c, t)$=TRUE. Due to the monotonicity of f, FINDMINEVIDENCE($\emptyset$, $\Pi \setminus M, c, t, f$) finally computes a t-minimal evidence if and only if $f(\Pi \setminus M, c, t)$ = TRUE. To judging whether $M \cup \{s\}$ forms a minimal hitting set for $\{E_0, E_1, \cdots, E_i\}$, we traverse all proper subsets of $M \cup \{s\}$. According to the Definition 2, if there exists a proper subset of $M \cup \{s\}$ satisfies the condition of having non-empty intersections with all evidence in $\{E_0, E_1, \cdots, E_i\}$, then it is a hitting set for $\{E_0, E_1, \cdots, E_i\}$. Thereby $M \cup \{s\}$ is not a minimal hitting set. Conversely, if such a proper subset does not exist, $M \cup \{s\}$ is confirmed as a minimal hitting set for $\{E_0, E_1, \cdots, E_i\}$. In the end, with the termination of Algorithm 1, $\mathbb{E}$ will contain all t-minimal evidences for the claim. Theorem 1 shows the completeness of the algorithm. In order to prove the completeness of the recursive algorithm FINDALLMINEVIDENCES(c, t, Π, f), the certification of Lemma 1 is necessary.

Lemma 1. *Suppose $\mathbb{E}$ is a set of t-minimal evidences for c in Π w.r.t. f, there exists a new t-minimal evidence $E \notin \mathbb{E}$ for c in Π w.r.t. f if and only if there exists a minimal hitting set M for $\mathbb{E}$ such that $f(\Pi \setminus M, c, t)$ = TRUE.*

Proof. According to Definition 2, each minimal hitting set M for $\mathbb{E}$ is a set of sentences *s.t.* for every $E \in \mathbb{E}$, $M \cap E \neq \emptyset$, which means for every $E \in \mathbb{E}$, there exists at least a sentence s *s.t.* $s \in E$ and $s \in M$. When there exists a new t-minimal evidence E' for c, every evidence $E \in \mathbb{E}$ has at least one sentence $s \notin E'$. We extract only a different sentence $s \notin E'$ from each evidence $E \in \mathbb{E}$ to form a hitting set M'. Therefore, there must exist a minimal hitting set M'' for $\mathbb{E}$ *s.t.* $M'' \subseteq M'$. We can conclude that for each $s \in E'$, *s.t.* $s \notin M''$. Thus, $E' \subseteq \Pi \setminus M''$ and $f(\Pi \setminus M'', c, t)$ = TRUE.

Similarly, if there exists a minimal hitting set M for $\mathbb{E}$ *s.t.* $f(\Pi \setminus M, c, t)$=TRUE, then a t-minimal evidence E' *s.t.* $E' \subseteq \Pi \setminus M$ for c in Π *w.r.t.* f can be found. For each evidence $E \in \mathbb{E}$, there exists at least one sentence $s \in E$, but not in $\Pi \setminus M$. That is to say, E' is different from any $E \in \mathbb{E}$ and $\Pi \setminus M$ contains a new t-minimal evidence for c in Π *w.r.t.* f. Lemma 1 can be proved. □

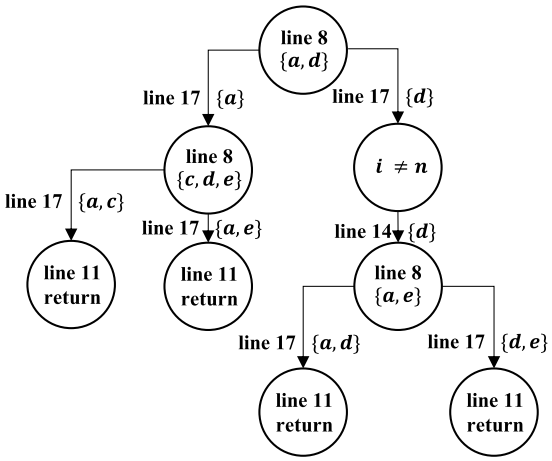

Fig. 2. The trace tree of Algorithm 1 for Example 1.

Theorem 1. *FindAllMinEvidences(c, t, Π, f) returns the set of all t-minimal evidences for c in Π w.r.t. f.*

Proof. We consider $\mathbb{E} = \{E_0, \cdots, E_{n-1}\}$ as the set of t-minimal evidence for c in Π *w.r.t.* f, where n is the size of $\mathbb{E}$. The set of all minimal hitting sets of $\mathbb{E}$ is denoted as $\mathsf{MHS}(\mathbb{E})$.

According to the Lemma 1, there doesn't exist a new t-minimal evidence $E \notin \mathbb{E}$ for c in Π *w.r.t.* f if and only if for all $M \in \mathsf{MHS}(\mathbb{E})$, $f(\Pi \setminus M, c, t) =$ FALSE. Thus, we must consider all $\mathsf{MHS}(\mathbb{E})$ for any size of $\mathbb{E}$ to make sure the final return $\mathbb{E}$ is the set of all t-minimal evidences for c in Π *w.r.t.* f.

If $n = 1$, from the definition of the hitting set, $\mathsf{MHS}(\{E_0\}) = \{\alpha | \alpha \in E_0\})$. And for any $n \geq 2$, if $M \in \mathsf{MHS}(\{E_0, \cdots, E_{n-1}\})$ holds, one of the following conditions must be satisfied:

- $M \in \mathsf{MHS}(\{E_0, \cdots, E_{n-2}\})$
- $M = M' \cup \{\alpha\}$, where $\alpha \in E_n$ and there exists a M', $M' \in \mathsf{MHS}(\{E_0, \cdots, E_{n-2}\})$

For any $M' \in \mathsf{MHS}(\{E_0, \cdots, E_{n-2}\})$, $E' \in \{E_0, \cdots, E_{n-2}\}$ satisfies $E' \cap M' \neq \emptyset$. When a new t-minimal evidence E_{n-1} is calculated, $\mathsf{MHS}(\{E_0, \cdots, E_{n-1}\})$ is a subset of $HS = \{M' \cup \{\alpha\} | M' \in \mathsf{MHS}(\{E_0, \cdots, E_{n-2}\}), \alpha \in E_{n-1}\}$. If $M' \cup \{\alpha\}$ is not a minimal hitting set for $\{E_0, \cdots, E_{n-1}\}$, there must exist an $E \in \{E_0, \cdots, E_{n-2}\}$ such that $\alpha \in E$. In other words, there must exist another $M'' \in \mathsf{MHS}(\{E_0, \cdots, E_{n-2}\})$ such that $M'' \in \mathsf{MHS}(\{E_0, \cdots, E_{n-1}\})$ and $\alpha \in M''$. Thus the above two conditions are proved. We can obtain $\mathsf{MHS}(\{E_0, \cdots, E_{n-1}\})$ from each $M' \in \mathsf{MHS}(\{E_0, \cdots, E_{n-2}\})$ and E_{n-1}.

From ConstructMinEvidence(i, M, c, t, f) in Algorithm 1, M is a minimal hitting set for $\{E_0, \cdots, E_{i-1}\}$. If $i = n$, there needs to judge whether exists a new t-minimal evidence for c in $\Pi \setminus M$ *w.r.t.* f. And $i \neq n$ means there exist other evidences in $\mathbb{E}$ that not are considered. If $M \cap E_i \neq \emptyset$, $M \in \mathsf{MHS}(\{E_0, \cdots, E_{i-1}, E_i\})$. If $M \cap E_i = \emptyset$, we obtain minimal hitting sets for $\{E_0, \cdots, E_{i-1}, E_i\}$ by judging $M \cup \alpha$, for each $\alpha \in E_i$. Thus, for each size of $\mathbb{E}$, we calculate all minimal hitting sets for $\mathbb{E}$. Finally, there is not $M \in \mathsf{MHS}(\{E_0, \cdots, E_{i-1}, E_i\})$ satisfying $f(\Pi \setminus \mathsf{MHS}(\mathbb{E}), c, t) =$ True. FindAllMinEvidences(c, t, Π, f) returns the set of all t-minimal evidences for c in Π *w.r.t.* f. □

Example 1. Given a claim u, a label t, the text corpus is $\Pi = \{a, b, c, d, e\}$, the trace tree of FindAllMinEvidences(u, t, Π_t, f) is depicted in Fig. 2, where the set beside a branch denotes a currently computed minimal hitting set, the set within a node shows a minimal evidence found concerning the minimal hitting set, and the line number comes from the ConstructMinEvidence procedure. The complete set of minimal evidences ultimately mined is $\mathbb{E} = \{\{a, d\}, \{c, d, e\}, \{a, e\}\}$.

3.2. Scorer and reasoner

As shown in Section 3.1, $f(S, c, t)$ in Algorithm 1 is used to determine whether the given set of sentences S supports or refutes the given claim c with respect to the given label t. And we implement $f(S, c, t)$ through two models, named *scorer* and *reasoner* respectively. Function 1 defines how f works based on a scorer and a reasoner. Initially, the scorer targets to screen sentences related to the given claim according to the label and to enforce the maximum number θ of tokens in output sentences that will be fed into the reasoner. And the reasoner focuses on judging the relation between the input sentences and the claim. By comparing the output of the reasoner with the given label, we can obtain the judgment of the function $f(S, c, t)$. Theorem 1 holds by assuming that the function f is monotonic. To ensure monotonicity as closely as possible, we develop separate scorers for different labels. In addition, we construct specifically the training set of reasoners to enhance the monotonicity of the function f.

Scorer. Given a claim c, a set of candidate sentences $S = \{s_1, s_2, \cdots, s_m\}$, a threshold parameter θ, and a true label $t \in \{\mathrm{T}, \mathrm{F}\}$, a scorer selects the most relevant sentences S_{ranking} for supporting/refuting c. We utilize the parameter θ to constrain the total number of

Function 1 $f(S, c, t)$.

Input: A set of sentences S, a claim c, and a label t.
Output: A boolean value that represents whether there exists a t-minimal evidence for c in S.

θ is a threshold to constrain the total number of tokens used in all output sentences of Scorer.
Scorer and Reasoner are two trained models.
$S_{\text{ranking}} \leftarrow \text{Scorer}(S, c, t, \theta)$
$\hat{t} \leftarrow \text{Reasoner}(S_{\text{ranking}}, c)$
return TRUE if and only if $\hat{t} = t$

> You are a fact-checker reviewing a claim. You need to rank all the evidence sentences according to the degree of supporting the claim. Here are the claim and the candidate sentences. Please sort the sentences by the degree of supporting the claim from high to low. Only reply a list of the ids of sentences. Use ',' to separate each id. The claim is: c. The candidate sentences are: Π.

Fig. 3. The prompt for guiding GPT-4o as a T-scorer.

tokens used in all output sentences. We divide scorers into two types: T-scorer estimates the relevance of a sentence for supporting a claim, whereas F-scorer estimates the relevance of a sentence for refuting a claim. We explore three strategies to select sentences with high relevance.

First of all, considering that the pretrained language models demonstrate high performance in many NLP tasks, the proposed first strategy for scorer is built upon a pretrained model RoBERTa [6]. The input is organized as the following sequence: $\langle s\rangle, c, \langle/s\rangle, s_k, \langle/s\rangle$, where $s_k \in S$, $\langle s\rangle$ and $\langle/s\rangle$ are special tokens introduced by the RoBERTa model. Through the final softmax layer, a fine-tuned RoBERTa-based model can produce a relevance probability score between 0 and 1. To enforce f to be as monotonic as possible, T-scorers and F-scorers are trained separately. The set S of candidate sentences is sorted in the descending order of relevance calculated by the T-scorer and the F-scorer, respectively.

For the second strategy, we utilize the distance between tf-idf vectors of each sentence in S and the sequence constructed by concatenating the claim c and the label t to evaluate the degree of relevance.

Regarding the last strategy, inspired by the recent success of large language models (LLMs), especially the GPT family [29], we uniformly utilize the GPT-4o[1] as a T-scorer and a F-scorer. Give a text corpus Π and a claim c, the prompt for guiding GPT-4o as a T-scorer is exhibited in Fig. 3, whereas the prompt for guiding GPT-4o as a F-scorer is almost the same except the word "supporting" is replaced with the word "refuting". In practice, the text corpus Π comprises sentences derived from the outcomes of the document retrieval process, which constitutes the initial stage of fact verification as discussed in Section 2.

Reasoner. After a set of the most relevant sentences S_{ranking} is retrieved by a scorer, a reasoner determines the relationship between S_{ranking} and c. Three different models are adopted as reasoners. First of all, similar to scorers, we employ a RoBERTa-based multi-classification model. We construct a sequence "$\langle s\rangle, c, \langle/s\rangle, s_1, \cdots, s_n, \langle/s\rangle$" as the input, where $s_1, \cdots, s_n$ are sentences ranked by their relevance to c. The reasoner is trained to predict the label $\hat{t} \in \{\text{T, F, N}\}$, where T, F, N respectively denote "SUPPORTS", "REFUTES" and "NOT ENOUGH INFO", determining whether the sequence supports or refutes the claim c or does not contain enough information to make a decision. Beyond that, we also use KGAT [19] and MLA [21] as reasoners. KGAT utilizes a kernel graph attention network to aggregate information of sentences. MLA is a simple model that enables inter-sentence attentions at different levels and can benefit from joint training. We choose the top five sentences from S_{ranking} into KGAT and MLA. KGAT and MLA reasoners are also trained to predict the label $\hat{t} \in \{\text{T, F, N}\}$.

4. Experiments

Due to the missing of true annotations for all minimal evidences, existing benchmark datasets for evidence retrieval cannot be directly used to evaluate the performance of mining all minimal evidences. Proper adjustments must be made to the large-scale dataset FEVER [7]. Besides, we construct other two datasets SemEval and Wiki80 whose claims are structural. These two datasets were modified from two benchmark datasets for relation extraction, respectively from SemEval-2010 Task 8 [8] and Wiki80 [9]. The implementation of MAME requires the training of both the scorer and the reasoner. Therefore, besides constructing the datasets for MAME, we also constructed datasets for scorers and reasoners.

4.1. Datasets for MAME

Each example in the reconstructed data for MAME includes a given claim c, a set $\mathbb{E}_T$ of T-minimal evidences for supporting c, a set $\mathbb{E}_F$ of F-minimal evidences for refuting c, a T-specific text corpus Π_T and a F-specific text corpus Π_F. Due to the requirement for training and evaluating scorers and reasoners, we divided each dataset into a training set, a development set and a test set. Statistical details for these datasets are reported in Table 1, including the numbers of claims, the claims with supporting evidences, the claims with refuting evidences, as well as the claims that have at least two minimal evidences.

[1] https://openai.com/index/hello-gpt-4o/.

Table 1
Statistical information for datasets.

Dateset		#claims	#supports	#refutes	#multi_evids
FEVER	Train	135,484	73,957	28,401	18,421
	Dev	9,738	3,254	3,260	1,113
	Test	9,841	3,298	3,280	1,199
SemEval	Train	38,768	25,622	13,150	8,704
	Dev	5,604	2,517	3,091	686
	Test	11,858	6,241	5,617	2,096
Wiki80	Train	96,309	67,150	29,169	4,952
	Dev	5,281	3,708	1,573	201
	Test	5,282	3,705	1,577	192

Example 1 (FEVER**)**
Claim: Taipei is the center of Taiwan.
Supporting Minimal Evidences:
["Taipei is the political, economic, educational, and cultural center of Taiwan, and one of the major hubs of the Chinese-speaking world. "]
Refuting Minimal Evidences:
[" Sitting at the northern tip of the island, Taipei City is an enclave of the municipality of New Taipei City. "]
***T*-specific Text Corpus:**
[" Taipei is the political, economic, educational, and cultural center of Taiwan, and one of the major hubs of the Chinese-speaking world. ",
" During the day most have programs that cater to Taiwanese seniors , with varied programs towards the evening that aim at the general public. ", ...]
***F*-specific Text Corpus:**
[" Sitting at the northern tip of the island, Taipei City is an enclave of the municipality of New Taipei City. ",
" During the day most have programs that cater to Taiwanese seniors , with varied programs towards the evening that aim at the general public. ", ...]

Example 2 (Wiki80**)**
Claim: Lu Kang's child is Lu Ji.
Supporting Minimal Evidences:
[" Lu Kang 's son , Lu Ji , was a scholar who served as an official under Wu 's founding emperor , Sun Quan . "]
[" Lu Ji , a son of Lu Xun 's granduncle Lu Kang, was one of the 24 Filial Exemplars and served as an official under Sun Quan. "]
Refuting Minimal Evidences:
[]
***T*-specific Text Corpus:**
[" Lu Kang 's son , Lu Ji , was a scholar who served as an official under Wu 's founding emperor , Sun Quan . ",
" Lu Ji , a son of Lu Xun 's granduncle Lu Kang, was one of the 24 Filial Exemplars and served as an official under Sun Quan. ",
" She married Lu Jing , who was born to Lu Kang and another daughter of Zhang Cheng ; both Sun He 's daughter and Lu Jing therefore were Zhang Cheng 's maternal grandchildren . ", ...]
***F*-specific Text Corpus:**
[]

Fig. 4. Examples in the reconstructed FEVER and Wiki80 datasets.

FEVER. Each claim c in FEVER have already been labeled as "SUPPORTS", "REFUTES" or "NOT ENOUGH INFO" (corresponding to T, F or N in our notations) and given the annotated evidences $\hat{\mathbb{E}} = \{\hat{E}_1, \hat{E}_2, \cdots\}$ for the support/refutation. To construct t-minimal evidences for $t \in \{T, F\}$, we performed the following procedure. Firstly, for each claim c and the annotated evidences $\hat{\mathbb{E}}$ for supporting(*resp.* refuting) c, and for every $\hat{E}_i \in \hat{\mathbb{E}}$, if $\exists \hat{E}_j \in \hat{\mathbb{E}}$ such that $\hat{E}_j \subset \hat{E}_i$, we deleted $\hat{E}_i$ from $\hat{\mathbb{E}}$. By this procedure we obtained the set of t-minimal evidences $\mathbb{E}_t = \{E_1, E_2, \cdots\}$. For each claim c, we utilized all sentences in the output of document retrieval phase of [30], mentioned in Section 2, as the initial text corpus of c. The t-specific text corpus Π_t for c is the union of the initial text corpus and all sentences in $E \in \mathbb{E}_t$.

SemEval and Wiki80 We enriched the two benchmark datasets for relation extraction, SemEval and Wiki80, with logical rules, since a set of sentences S verifies the fact $r(h,t)$ if and only if the union of a set of background logical rules and the set of triples supported by S entails $r(h,t)$ under the semantics of first-order logic. The original datasets do not come with logical rules, thus we applied the AMIE+ [31] tool[2] to mine rules that have up to three body atoms from the complete set of triples. All rules mined are Horn and have binary predicates only. We manually pick out the rules that are correct according to the consensus of two experts, yielding a set of background rules.

Based on the set of background rules, for every unverified fact $r(h,t)$, we translated $r(h,t)$ into a natural language sentence as a claim c and constructed its T-minimal evidences $\mathbb{E}_T$ and the T-specific text corpus through the following three-step procedure:

- We obtained all supporting minimal explanations for the unverified fact $r(h,t)$, where a supporting minimal explanation for $r(h,t)$ is a minimal set of triples in the original dataset which together with the set of background rules entails $r(h,t)$.
- We calculated the set of supporting minimal evidences $\mathbb{E}_T$ for $r(h,t)$, where every supporting minimal evidence is a minimal set of sentences that supports all triples in a supporting minimal explanation.
- All sentences in supporting minimal evidences can be treated as relevant to $r(h,t)$. To train scorers to distinguish relevant sentences from irrelevant ones, the T-specific text corpus for $r(h,t)$, which is used to trained scorers, should not only contain sentences in supporting minimal evidences but also those sentences that are outside but closely related to supporting minimal evidences. We initialized the T-specific text corpus for $r(h,t)$ as the set of sentences appearing in supporting minimal evidences. To add irrelevant sentences outside any supporting minimal evidences, we coped with triples in the original dataset but not in

[2] https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/amie/.

supporting minimal explanations one by one: if the head entity or tail entity appears in supporting minimal explanations, and adding the triple to the union of all supporting minimal explanations will not yield new supporting minimal explanations under the set of background rules, all sentences that support the triple were added to the T-specific text corpus for $r(h,t)$.

Regarding the computation of refuting minimal evidences for $r(h,t)$, since each supporting minimal evidence for $s(h,t)$ can be treated as a refuting minimal evidence for $r(h,t)$ when the binary relations s and r are disjoint, we obtained all refuting minimal evidences for $r(h,t)$ by adding logical rules $s(h,t) \rightarrow not\_r(h,t)$ into the set of background rules for all binary relations s disjoint with r, and by treating the set of supporting minimal evidences for $not\_r(h,t)$ as the set of refuting minimal evidences for $r(h,t)$. Accordingly, we treated the T-specific text corpus for $not\_r(h,t)$ as the F-specific text corpus for $r(h,t)$.

Fig. 4 provides two examples to respectively showcase the appearance of the reconstructed FEVER and Wiki80 datasets.

4.2. Data generation for scorers and reasoners

We generated data for scorers and reasoners by using the same train/dev/test splits of the MAME datasets.

Scorers The train/dev/test sets of scorers were constructed from the corresponding train/dev/test sets of MAME by one by one converting examples in MAME to examples of scorers. For each example in MAME which consists of a claim c, a set $\mathbb{E}_T$ of T-minimal evidences for supporting c, a set $\mathbb{E}_F$ of F-minimal evidences for refuting c, a T-specific text corpus Π_T and a F-specific text corpus Π_F, one example of T-scorer was constructed for every sentence $s \in \Pi_T$, i.e., if $s \in E$ for some $E \in \mathbb{E}_T$, then the constructed example is a triple (c, s, **relevant**), otherwise the constructed example is a triple (c, s, **irrelevant**); likewise, one example of F-scorer was constructed for every sentence $s \in \Pi_F$, i.e., if $s \in E$ for some $E \in \mathbb{E}_F$, then the constructed example is a triple (c, s, **relevant**), otherwise the constructed example is a triple (c, s, **irrelevant**).

Reasoners Reasoners are used to determine whether a text corpus supports, refutes or has no enough information about the support/refutation of a claim. We exploited two ways to respectively construct two versions of train/dev/test sets of reasoners. The first version, named the *basic* version, is directly derived from the corresponding train/dev/test sets of MAME, whereas the second one, named the *enhanced* version, is derived from all invocations of the function f defined in Function 1. In more detail, for each example in MAME which consists of a claim c, a set $\mathbb{E}_T$ of T-minimal evidences for supporting c, a set $\mathbb{E}_F$ of F-minimal evidences for refuting c, a T-specific text corpus Π_T and a F-specific text corpus Π_F, multiple examples in the corresponding train/dev/test set of reasoners were constructed, each of which consists of the claim c, a text corpus $\mathcal{P}$ and a label $l \in \{T, F, N\}$, where the label T denotes that $\mathcal{P}$ supports c, F denotes that $\mathcal{P}$ refutes c, and N denotes that $\mathcal{P}$ has no enough information about the support/refutation of c.

Regarding the basic version, for every minimal evidence $E \in \mathbb{E}_T$, we constructed one example (c, $\mathcal{P}$, T) for $\mathcal{P} = E \cup (\Pi_T \setminus \bigcup \mathbb{E}_T)$, as well as one example (c, $\mathcal{P}_e$, N) for each $e \in E$, where $\mathcal{P}_e = E \setminus \{e\} \cup (\Pi_T \setminus \bigcup \mathbb{E}_T)$; likewise, for every minimal evidence $E \in \mathbb{E}_F$, we constructed one example (c, $\mathcal{P}$, F) for $\mathcal{P} = E \cup (\Pi_F \setminus \bigcup \mathbb{E}_F)$), as well as one example (c, $\mathcal{P}_e$, N) for each $e \in E$, where $\mathcal{P}_e = E \setminus \{e\} \cup (\Pi_F \setminus \bigcup \mathbb{E}_F)$.

Regarding the enhanced version, we invoked FINDALLMINEVIDENCES(c, T, Π_T, f) and FINDALLMINEVIDENCES(c, F, Π_F, f) in turn. During the invocation of FINDALLMINEVIDENCES(c, t, Π_t, f) no matter whether t is T or F, we constructed one example (c, $\mathcal{P}$, l) for each inner invocation of $f(\mathcal{P}, c, t)$, where $l = T$ if $\mathcal{P}$ is a superset of some $E \in \mathbb{E}_T$ but not a superset of any $E \in \mathbb{E}_F$, $l = F$ if $\mathcal{P}$ is a superset of some $E \in \mathbb{E}_F$ but not a superset of any $E \in \mathbb{E}_T$, otherwise $l = N$.

4.3. Evaluation metrics

The purpose of scorers is to select the most related sentences for supporting or refuting a claim. Therefore, we ranked all sentences with the same claim based on the relevance scores output by the scorers, and subsequently calculated recall@5 and NDCG to measure the performance of the scorers. Recall@5 refers to the recall rate of the first five sentences. The metric NDCG (short for Normalized Discounted Cumulative Gain) is defined as follows:

$$\mathrm{NDCG} = \frac{\mathrm{DCG}}{\mathrm{IDCG}} = \frac{\sum_{i=1}^{p} \frac{r_i}{\log_2(i+1)}}{\sum_{i=1}^{q} \frac{1}{\log_2(i+1)}} \tag{1}$$

where q is the number of relevant sentences, p is the number of total sentences with the same claim, and $r_i = 1$ if the i-th sentence in the sorted list is relevant with the given claim or $r_i = 0$ otherwise. NDCG measures the gap between the current sorted list and the ideal sorted list where all the relevant sentences are at the top place. The larger NDCG is, the closer the sorted list is to the ideal one.

Reasoners actually conduct a multi-classification task, so we utilize accuracy, F1-score and Kappa to measure its performance. F1-score is divided into two categories $\mathrm{F1}_T$ and $\mathrm{F1}_F$, which represent F1 of binary classification where regard SUPPORTS and REFUTES as the positive class respectively. Kappa, short for Cohen's Kappa, is a metric that quantifies the level of agreement between the true labels and the predicted labels beyond what would be expected by chance alone. Recall(R), precision(P) and F1-score(F1) are utilized to evaluate the performance of mining all minimal evidences. Specifically, recall, precision and F1-score are defined as follows respectively: $\mathrm{recall} = \frac{c}{g}$, $\mathrm{precision} = \frac{c}{p}$, $\mathrm{F1} = \frac{2 \times \mathrm{precision} \times \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}$ where c is the number of the predicted correct minimal evidences, g is the number of ground truth minimal evidences and p is the number of the predicted minimal evidences. Note that a predicted minimal evidence is considered to be entirely correct if and only if it is completely identical to one of the annotated evidences. This strict equality judgment is rarely used in evaluating traditional evidence retrieval, but is crucial to estimate the effectiveness in retrieving minimal evidences. We calculate the average of the metrics for all examples. All numbers are reported in percentage.

Table 2
Comparison results of different scorers on different test sets.

		FEVER		SemEval		Wiki80	
Scorer	Dataset	Rec@5	NDCG	Rec@5	NDCG	Rec@5	NDCG
RoBERTa	sup	**96.99**	**88.63**	**95.53**	**90.90**	**99.87**	**98.48**
	ref	**95.39**	**84.38**	**100.00**	**98.72**	**99.56**	**96.79**
Tf-idf	sup	65.40	52.12	83.39	72.53	86.92	77.62
	ref	62.94	50.24	88.11	77.04	97.75	93.27
GPT-4o	sup	91.64	87.12	72.12	75.30	92.49	96.00
	ref	89.54	80.60	75.77	60.88	78.84	86.67

Table 3
Comparison results of different reasoners on different test sets.

		FEVER				SemEval				Wiki80			
Reasoner	Dataset	Acc	$F1_T$	$F1_F$	Kappa	Acc	$F1_T$	$F1_F$	Kappa	Acc	$F1_T$	$F1_F$	Kappa
RoBERTa	basic	72.52	67.93	59.91	50.32	87.28	67.49	91.61	68.99	91.79	92.10	85.98	86.42
	enhanced	74.68	78.67	65.50	60.30	**94.69**	**92.63**	**95.93**	**90.80**	91.74	94.03	86.09	86.53
KGAT	basic	76.28	65.54	56.82	50.94	61.51	48.08	92.44	33.80	92.10	92.75	86.03	87.00
	enhanced	89.81	84.95	**82.22**	**82.18**	71.87	56.35	91.65	47.87	**97.04**	**96.82**	**92.21**	**95.03**
MLA	basic	67.86	67.74	60.50	47.76	69.92	51.89	92.43	42.04	88.83	91.08	77.42	80.47
	enhanced	**90.10**	**85.63**	81.53	80.76	80.40	56.45	92.32	54.74	96.11	94.70	84.24	91.61

Table 4
Comparison results of MAME instantiated by the RoBERTa-based scorers and different reasoners on different test sets.

		FEVER			SemEval			wiki80		
Reasoner	Dataset	P	R	F1	P	R	F1	P	R	F1
RoBERTa	basic	43.39	46.86	43.22	**76.58**	79.42	76.91	84.55	87.70	85.45
	enhanced	53.42	72.28	57.16	76.56	**82.10**	**77.77**	**88.61**	94.05	**90.21**
KGAT	basic	53.40	52.38	51.50	68.26	70.65	68.66	88.13	91.51	89.15
	enhanced	51.94	67.25	54.73	70.76	74.56	71.67	85.90	**95.65**	88.89
MLA	basic	**55.14**	65.94	56.95	74.99	81.19	76.43	86.22	91.64	87.85
	enhanced	55.12	**70.25**	**58.07**	75.52	80.82	76.61	86.44	90.10	87.55

4.4. Experiments setup

We trained RoBERTa-based scorers and all reasoners by setting the learning rate to 5e-5, except that the learning rate of KGAT on SemEval and Wiki80 was set to 1e-5. The batch size of all scorers was set to 24 for Wiki80 and 48 for SemEval and FEVER. The batch size of all reasoners was set to 16 for all datasets. The maximum total number of tokens that can be used in all output sentences of scorers was set to 512. All experiments were conducted on a workstation with 128 GB memory and GeForce RTX 3090 GPU.

4.5. Results and analysis

Results of Scorer. We compared all scorers by measuring their outputs on the test sets of the three datasets presented in Section 4.2. Table 2 reports the comparison results of all scorers. It can be noticed that the RoBERTa-based scorers consistently outperform other scorers. The tf-idf scorers are often the worst because they calculate the distances among tf-idf vectors without exploiting the training sets. As for GPT-4o scorers, although good performance can be achieved on some datasets, they are very time-consuming and costly. As a whole, it can draw a conclusion that the RoBERTa-based scorers are sufficiently good at estimating the relevance of sentences.
Results of Reasoner. We compared all reasoners by measuring their outputs on the test sets of the three datasets presented in Section 4.2. Table 3 reports the comparison results. It can be seen that reasoners trained on the enhanced data outperform the corresponding reasoners trained on the basic data. This is probably due to the construction process which makes the enhanced version of datasets have more examples than the basic version of datasets. It is worth to noted that more training examples will lead to stronger monotonicity in determining the supporting/refuting evidences. We can clearly see that the KGAT and MLA clearly outperform the RoBERTa-based reasoner on FEVER, probably because KGAT and MLA are more effective in reasoning about textual claims. On the contrary, the three reasoners perform comparably well on Wiki80, whereas the RoBERTa-based reasoner outperforms KGAT and MLA on SemEval. These results imply that complex graph networks may not consistently perform well on textual data translated from structural information.

You are a minimal evidence finder. An evidence is a set of sentences. The minimal evidence means that every sentence is indispensable. You need to find all minimal evidences that support the claim. Here are the claim c and the candidate sentences Π. Please return all minimal evidence sets. Only reply multiple lists. Each minimal evidence is a list[] of the ids of sentences. Use ';' to separate each minimal evidence. The claim is: c. The candidate sentences are: Π.

Fig. 5. The prompt for guiding GPT-4o to mine all T-minimal evidences.

Table 5
Comparison results for mining all minimal evidences on the test set of FEVER.

Method		P	R	F1
KGAT		0.135	0.135	0.135
DQN		8.31	7.92	8.04
GPT-4o		37.31	45.10	38.99
MAME	$\text{RoBERTa}_{\text{basic}}$	43.39	46.86	43.22
	$\text{RoBERTa}_{\text{enhanced}}$	53.42	72.28	57.16
	$\text{KGAT}_{\text{basic}}$	53.40	52.38	51.50
	$\text{KGAT}_{\text{enhanced}}$	51.94	67.25	54.73
	$\text{MLA}_{\text{basic}}$	**55.14**	65.94	56.95
	$\text{MLA}_{\text{enhanced}}$	55.12	**70.25**	**58.07**

Results of MAME. Our proposed MAME framework can be instantiated by different scorers and different reasoners. As the RoBERTa-based scorers have demonstrated the best performance across different datasets, we instantiated MAME with the RoBERTa-based T-scorer and the RoBERTa-based F-scorer, and compared different MAME instantiations with different reasoners. All scorers and reasoners used were trained in our previous experiments. As can be seen from the table, the performance of the MAME instantiations depends on the performance of the reasoner used in MAME. By comparing Table 4 with Table 3, it can further be seen that the performance of the MAME instantiations is proportional to the performance of the inner reasoner. In particular, an MAME instantiation with the inner reasoner trained on the enhanced data outperforms that with the inner reasoners trained on the basic data. On the other hand, no matter which reasoner is used, MAME is shown to be effective in mining all minimal evidences.

Discussion. Traditional evidence retrieval methods consider a sentence set to be correct if it includes at least one minimal evidence. In contrast, our proposed framework MAME adopts more rigorous metrics, requiring all discovered evidences to be minimal.

We evaluated KGAT [19] and DQN [4] and GPT-4o in terms of our metrics. In particular, as for GPT-4o, we guided it to mine all T-minimal evidences and all F-minimal evidences separately, where the prompt for mining all T-minimal evidences is given by Fig. 5, and the prompt for mining all F-minimal evidences is obtained from the former prompt by replacing the word "support" with the word "refute".

Table 5 reports the comparison results among KGAT, DQN, GPT-4o and different MAME instantiations, where all numbers are shown in percentage. It can be seen from the table that KGAT achieves extremely low performance, since KGAT can only compute the top-5 most relevant sentences as evidence. Although DQN does not fix the number of sentences found in the resulting evidence, it is still challenging to be completely consistent with a specific annotated evidence. GPT-4o, as a powerful model for understanding natural language, is still inferior to MAME in extracting all minimal evidences. These results highlight the effectiveness of our neural-symbolic framework MAME in mining all minimal evidences.

To demonstrate how effective MAME is in mining multiple minimal evidences, we separately report the F1-score for supporting examples (denoted F1_{sup}) and the F1-score for refuting examples (denoted F1_{ref}) against different number of minimal evidences on the test set of FEVER, as shown in Table 6. The table shows that, no matter which reasoner trained on the enhanced data is employed in MAME, MAME is still effective in discovering multiple minimal evidences. As the number of annotated minimal evidences for a test example increases, the performance of MAME drops as expected, since the discovery of a larger number of minimal evidences for a test example is more challenging due to error propagation.

Case study. Table 7 provides three examples to illustrate the results of MAME in mining all minimal evidences from the test set of FEVER, where MAME is instantiated with the RoBERTa-based scorers and the KGAT reasoner trained on the enhanced data. For case #1, the claim has both supporting and refuting minimal evidences. MAME accurately computes all minimal evidences for supporting and refuting the claim. As for case#2 and case #3, there are multiple minimal evidences for either supporting or refuting the corresponding claims. MAME still accurately computes all minimal evidences. These cases reveal that MAME is able to accurately mine minimal evidences even when there are multiple minimal evidences or there exist both supporting evidences and refuting evidences.

5. Conclusions and future work

In this article, we have investigated finding all irreducible perspectives that can support or refute an unverified claim. We have introduced a novel concept of minimal evidence and developed a neural-symbolic framework based on this concept. Our framework

Table 6
Comparison results of MAME on the test set of FEVER against different number of minimal evidences.

evidence_num	RoBERTa		KGAT		MLA	
	$F1_{sup}$	$F1_{ref}$	$F1_{sup}$	$F1_{ref}$	$F1_{sup}$	$F1_{ref}$
1	66.27	53.00	63.98	51.70	65.73	57.05
2	55.81	47.97	50.98	41.65	51.55	45.15
3	46.42	44.00	42.37	36.80	43.48	41.04
4	38.81	44.15	37.65	37.82	37.49	43.75
5	34.58	30.05	28.76	26.76	26.14	30.48
6	49.05	37.20	37.07	26.98	47.42	38.64
7	23.08	29.02	25.42	16.12	13.68	28.25
8	22.60	18.57	25.30	25.54	27.81	25.08
9	35.71	22.50	22.78	12.15	22.86	15.43
10	21.67	21.11	20.00	17.17	6.7	30.77

Table 7
Examples for mining all minimal evidences on the test set of FEVER. Gold Evidences represent the annotated evidences and Predicted Evidences represent the mined evidences from MAME.

Case	Claim	Label	Gold Evidences	Predicted Evidences
#1	Penny Dreadful premiered on May 11, 2014.	SUPPORTS	The series premiered on Showtime on May 11, 2014, the first in an eight-episode season .	The series premiered on Showtime on May 11, 2014, the first in an eight-episode season .
		REFUTES	It premiered at the South by Southwest film festival on March 9 and began airing on television on April 28, 2014, on Showtime on Demand .	It premiered at the South by Southwest film festival on March 9 and began airing on television on April 28, 2014, on Showtime on Demand .
#2	Janet Leigh was incapable of writing.	REFUTES	Janet Leigh -LRB- born Jeanette Helen Morrison; July 6, 1927 – October 3, 2004 -RRB- was an American actress, singer, dancer and author .	Janet Leigh -LRB- born Jeanette Helen Morrison; July 6, 1927 – October 3, 2004 -RRB- was an American actress, singer, dancer and author .
			She also wrote four books between 1984 and 2002, including two novels .	She also wrote four books between 1984 and 2002, including two novels .
#3	Ernest Medina participated in the My Lai Massacre.	SUPPORTS	He was court martialed in 1971 for his role in the My Lai Massacre, but acquitted the same year .	He was court martialed in 1971 for his role in the My Lai Massacre, but acquitted the same year .
			He was the commanding officer of Company C, 1st Battalion, 20th Infantry of the 11th Brigade, Americal Division, the unit responsible for the My Lai Massacre of 16 March 1968 .	He was the commanding officer of Company C, 1st Battalion, 20th Infantry of the 11th Brigade, Americal Division, the unit responsible for the My Lai Massacre of 16 March 1968 .

leverages two neural models, a scorer and a reasoner, which are trained on annotated minimal evidences to mine all minimal evidences in a logical way. Experimental results demonstrate that our framework is effective in mining all minimal evidences for both textual claims and structural claims. Future work will introduce joint learning of the scorer and reasoner to mitigate the error propagation issue.

CRediT authorship contribution statement

Yanan Liu: Writing – review & editing, Writing – original draft, Validation, Project administration, Methodology, Investigation, Data curation, Conceptualization. **Hai Wan:** Writing – review & editing, Supervision, Methodology, Funding acquisition, Conceptualization. **Jianfeng Du:** Writing – review & editing, Supervision, Methodology, Conceptualization. **Yao Wang:** Writing – original draft, Validation, Investigation. **Kunxun Qi:** Writing – review & editing, Methodology, Conceptualization. **Weilin Luo:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by National Natural Science Foundation of China (No. 62276284), Guangdong Basic and Applied Basic Research Foundation (No. 2023A1515011470, 2022A1515011355), Guangzhou Science and Technology Project (No. 202201011699), Shenzhen Science and Technology Program (KJZD20230923114059O2), as well as the Fundamental Research Funds for the Central Universities, Sun Yat-sen University (No. 23ptpy31).

Data availability

Data will be made available on request.

References

[1] A.C. Kakas, L. Michael, Abduction and argumentation for explainable machine learning: a position survey, CoRR, arXiv:2010.12896, 2020.
[2] J. Zhou, X. Han, C. Yang, Z. Liu, L. Wang, C. Li, M. Sun, GEAR: graph-based evidence aggregating and reasoning for fact verification, in: ACL, 2019, pp. 892–901.
[3] J. Chen, Q. Bao, C. Sun, X. Zhang, J. Chen, H. Zhou, Y. Xiao, L. Li, LOREN: logic-regularized reasoning for interpretable fact verification, in: AAAI, 2022, pp. 10482–10491.
[4] H. Wan, H. Chen, J. Du, W. Luo, R. Ye, A DQN-based approach to finding precise evidences for fact verification, in: ACL, 2021, pp. 1030–1039.
[5] M. Fajcik, P. Motlícek, P. Smrz, Claim-dissector: an interpretable fact-checking system with joint re-ranking and veracity prediction, in: ACL (Findings), 2023, pp. 10184–10205.
[6] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, RoBERTa: a robustly optimized BERT pretraining approach, CoRR, arXiv:1907.11692, 2019.
[7] J. Thorne, A. Vlachos, C. Christodoulopoulos, A. Mittal, FEVER: a large-scale dataset for fact extraction and verification, in: NAACL-HLT, 2018, pp. 809–819.
[8] I. Hendrickx, S.N. Kim, Z. Kozareva, P. Nakov, D.Ó. Séaghdha, S. Padó, M. Pennacchiotti, L. Romano, S. Szpakowicz, SemEval-2010 task 8: multi-way classification of semantic relations between pairs of nominals, in: SemEval@ACL, 2010, pp. 33–38.
[9] X. Han, T. Gao, Y. Yao, D. Ye, Z. Liu, M. Sun, OpenNRE: an open and extensible toolkit for neural relation extraction, in: EMNLP-IJCNLP, 2019, pp. 169–174.
[10] X. Hu, Z. Guo, G. Wu, A. Liu, L. Wen, P. Yu, CHEF: a pilot Chinese dataset for evidence-based fact-checking, in: NAACL-HLT, 2022, pp. 3362–3376.
[11] J. Nørregaard, L. Derczynski, DanFEVER: claim verification dataset for Danish, in: NoDaLiDa, 2021, pp. 422–428.
[12] H. Ullrich, J. Drchal, M. Rỳpar, H. Vincourová, V. Moravec, CsFEVER and CTKFacts: acquiring Czech data for fact verification, Lang. Resour. Eval. 57 (2023) 1571–1605.
[13] J. Thorne, A. Vlachos, O. Cocarascu, C. Christodoulopoulos, A. Mittal, The fact extraction and verification (FEVER) shared task, FEVER (2018) 1–9.
[14] Y. Nie, H. Chen, M. Bansal, Combining fact extraction and verification with neural semantic matching networks, in: AAAI, 2019, pp. 6859–6866.
[15] K. Jiang, R. Pradeep, J. Lin, Exploring listwise evidence reasoning with T5 for fact verification, in: ACL, 2021, pp. 402–410.
[16] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P.J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, J. Mach. Learn. Res. 21 (2020) 140:1–140:67.
[17] Y. Wang, C. Xia, C. Si, B. Yao, T. Wang, Robust reasoning over heterogeneous textual information for fact verification, IEEE Access 8 (2020) 157140–157150.
[18] W. Zhong, J. Xu, D. Tang, Z. Xu, N. Duan, M. Zhou, J. Wang, J. Yin, Reasoning over semantic-level graph for fact checking, in: ACL, 2020, pp. 6170–6180.
[19] Z. Liu, C. Xiong, M. Sun, Z. Liu, Fine-grained fact verification with kernel graph attention network, in: ACL, 2020, pp. 7342–7351.
[20] S. Subramanian, K. Lee, Hierarchical evidence set modeling for automated fact extraction and verification, in: EMNLP, 2020, pp. 7798–7809.
[21] C. Kruengkrai, J. Yamagishi, X. Wang, A multi-level attention model for evidence-based fact checking, in: ACL, 2021, pp. 2447–2460.
[22] A. Zou, Z. Zhang, H. Zhao, Decker: double check with heterogeneous knowledge for commonsense fact verification, in: ACL (Findings), 2023, pp. 11891–11904.
[23] J. Kim, S. Park, Y. Kwon, Y. Jo, J. Thorne, E. Choi, FactKG: fact verification via reasoning on knowledge graphs, in: ACL, 2023, pp. 16190–16206.
[24] A. Krishna, S. Riedel, A. Vlachos, ProoFVer: natural logic theorem proving for fact verification, Trans. Assoc. Comput. Linguist. 10 (2022) 1013–1030.
[25] W. Yin, D. Roth, TwoWingOS: a two-wing optimization strategy for evidential claim verification, in: EMNLP, 2018, pp. 105–114.
[26] T. Chakrabarty, T. Alhindi, S. Muresan, Robust document retrieval and individual evidence modeling for fact extraction and verification, in: FEVER@EMNLP, 2018, pp. 127–131.
[27] J. Chen, R. Zhang, J. Guo, Y. Fan, X. Cheng, GERE: generative evidence retrieval for fact verification, in: SIGIR, 2022, pp. 2184–2189.
[28] U. Junker, QUICKXPLAIN: preferred explanations and relaxations for over-constrained problems, in: D.L. McGuinness, G. Ferguson (Eds.), AAAI, 2004, pp. 167–172.
[29] T.B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D.M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, in: NeurIPS, 2020.
[30] A. Hanselowski, H. Zhang, Z. Li, D. Sorokin, B. Schiller, C. Schulz, I. Gurevych, UKP-Athene: multi-sentence textual entailment for claim verification, in: FEVER@EMNLP, 2018, pp. 103–108.
[31] L. Galárraga, C. Teflioudi, K. Hose, F.M. Suchanek, Fast rule mining in ontological knowledge bases with AMIE+, VLDB J. 24 (2015) 707–730.