Article Reranking by Memory-Enhanced Key Sentence Matching for Detecting Previously Fact-Checked Claims

Qiang Sheng[1,2], **Juan Cao**[1,2], **Xueyao Zhang**[1,2], **Xirong Li**[3], **Lei Zhong**[1,2]
[1] Key Laboratory of Intelligent Information Processing,
Institute of Computing Technology, Chinese Academy of Sciences
[2]University of Chinese Academy of Sciences
[3]Key Lab of Data Engineering and Knowledge Engineering, Renmin University of China
{shengqiang18z,caojuan,zhangxueyao19s,zhonglei18s}@ict.ac.cn
xirong@ruc.edu.cn

Abstract

False claims that have been previously fact-checked can still spread on social media. To mitigate their continual spread, detecting previously fact-checked claims is indispensable. Given a claim, existing works retrieve fact-checking articles (FC-articles) for detection and focus on reranking candidate articles in the typical two-stage retrieval framework. However, their performance may be limited as they ignore the following characteristics of FC-articles: (1) claims are often quoted to describe the checked events, providing lexical information besides semantics; and (2) sentence templates to introduce or debunk claims are common across articles, providing pattern information. In this paper, we propose a novel reranker, MTM (Memory-enhanced Transformers for Matching), to rank FC-articles using key sentences selected using event (lexical and semantic) and pattern information. For event information, we propose to finetune the Transformer with regression of ROUGE. For pattern information, we generate pattern vectors as a memory bank to match with the parts containing patterns. By fusing event and pattern information, we select key sentences to represent an article and then predict if the article fact-checks the given claim using the claim, key sentences, and patterns. Experiments on two real-world datasets show that MTM outperforms existing methods. Human evaluation proves that MTM can capture key sentences for explanations. The code and the dataset are at https://github.com/ICTMCG/MTM.

1 Introduction

Social media posts with false claims have led to real-world threats on many aspects such as politics (Fisher et al., 2016), social order (Wang and Li, 2011), and personal health (Chen, 2020).

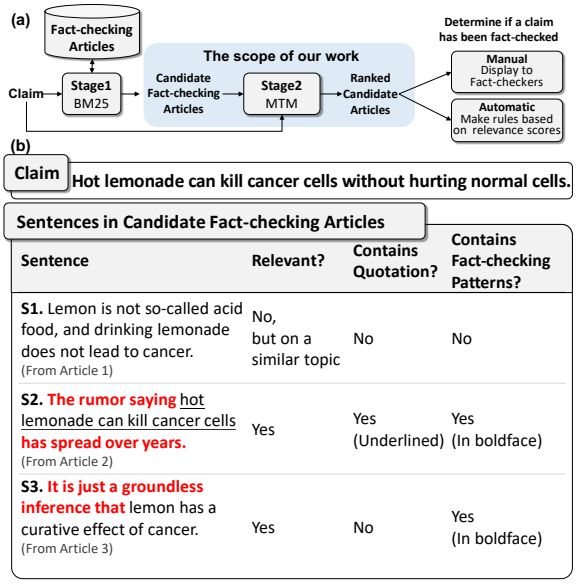

Figure 1: (a) Workflow of detecting a previously fact-checked claim. Our model MTM focuses on the second stage, i.e., reranking the candidates. (b) A claim and sentences in the candidate fact-checking articles (translated from Chinese). S1 is on a similar topic but actually irrelevant, while S2 and S3 which contain quotation or fact-checking patterns are relevant.

To tackle this issue, over 300 fact-checking projects have been launched, such as Snopes[1] and Jiaozhen[2] (Duke Reporters' Lab, 2020). Meanwhile, automatic systems have been developed for detecting suspicious claims on social media (Zhou et al., 2015; Popat et al., 2018a). This is however not the end. A considerable amount of false claims continually spread, even though they are already proved false. According to a recent report (Xinhua Net, 2019), around 12% of false claims published on Chinese social media, are actually "old", as they have been debunked previously. Hence, detecting previously fact-checked claims is an important

[1]https://www.snopes.com
[2]https://fact.qq.com/

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 5468–5481
August 1–6, 2021.

task.

According to the seminal work by Shaar et al. (2020), the task is tackled by a two-stage information retrieval approach. Its typical workflow is illustrated in Figure 1(a). Given a claim as a query, in the first stage a basic searcher (e.g., BM25 Robertson and Zaragoza, 2009) searches for candidate articles from a collection of fact-checking articles (FC-articles). In the second stage, a more powerful model (e.g., BERT, Devlin et al., 2019) reranks the candidates to provide evidence for manual or automatic detection. Existing works focus on the reranking stage: Vo and Lee (2020) model the interactions between a claim and the whole candidate articles, while Shaar et al. (2020) extract several semantically similar sentences from FC-articles as a proxy. Nevertheless, these methods treat FC-articles as *general* documents and ignore characteristics of FC-articles. Figure 1(b) shows three sentences from candidate articles for the given claim. Among them, S1 is more friendly to semantic matching than S2 and S3 because the whole S1 focuses on describing its topic and does not contain tokens irrelevant to the given claim, e.g., "has spread over years" in S2. Thus, a semantic-based model does not require to have strong filtering capability. If we use only general methods on this task, the relevant S2 and S3 may be neglected while irrelevant S1 is focused. To let the model focus on key sentences (i.e., sentences as a good proxy of article-level relevance) like S2 and S3, we need to consider two characteristics of FC-articles besides semantics: **C1**. Claims are often quoted to describe the checked events (e.g., the underlined text in S2); **C2**. Event-irrelevant patterns to introduce or debunk claims are common in FC-articles (e.g., bold texts in S2 and S3).

Based on the observations, we propose a novel reranker, `MTM` (Memory-enhanced Transformers for Matching). The reranker identifies key sentences per article using claim- and pattern-sentence relevance, and then integrates information from the claim, key sentences, and patterns for article-level relevance prediction. In particular, regarding **C1**, we propose `ROUGE`-guided Transformer (ROT) to score claim-sentence relevance literally and semantically. As for **C2**, we obtain the pattern vectors by clustering the difference of sentence and claim vectors for scoring pattern-sentence relevance and store them in the Pattern Memory Bank (PMB). The joint use of ROT and PMB allows us to identify key sentences that reflect the two characteristics of FC-articles. Subsequently, fine-grained interactions among claims and key sentences are modeled by the multi-layer Transformer and aggregated with patterns to obtain an article-level feature representation. The article feature is fed into a Multi-layer Perceptron (MLP) to predict the claim-article relevance.

To validate the effectiveness of our method, we built the first Chinese dataset for this task with 11,934 claims collected from Chinese Weibo[3] and 27,505 fact-checking articles from multiple sources. 39,178 claim-article pairs are annotated as relevant. Experiments on the English dataset and the newly built Chinese dataset show that `MTM` outperforms existing methods. Further human evaluation and case studies prove that `MTM` finds key sentences as explanations. Our main contributions are as follows:

- We propose a novel reranker `MTM` for fact-checked claim detection, which can better identify key sentences in fact-checking articles by exploiting their characteristics.
- We design `ROUGE`-guided Transformer to combine lexical and semantic information and propose a memory mechanism to capture and exploit common patterns in fact-checking articles.
- Experiments on two real-world datasets show that `MTM` outperforms existing methods. Further human evaluation and case studies prove that our model finds key sentences as good explanations.
- We built the first Chinese dataset for fact-checked claim detection with fact-checking articles from diverse sources.

2 Related Work

To defend against false information, researchers are mainly devoted to two threads: (1) **Automatic fact-checking** methods mainly retrieve relevant factual information from designated sources and judge the claim's veracity. Thorne et al. (2018) use Wikipedia as a fact tank and build a shared task for automatic fact-checking, while Popat et al. (2018b) and Wang et al. (2018) retrieve webpages as evidence and use their stances on claims for veracity prediction. (2) **Fake news detection** methods often use non-factual signals, such as styles (Przybyla, 2020; Qi et al., 2019), emotions (Ajao

[3] `https://weibo.com`

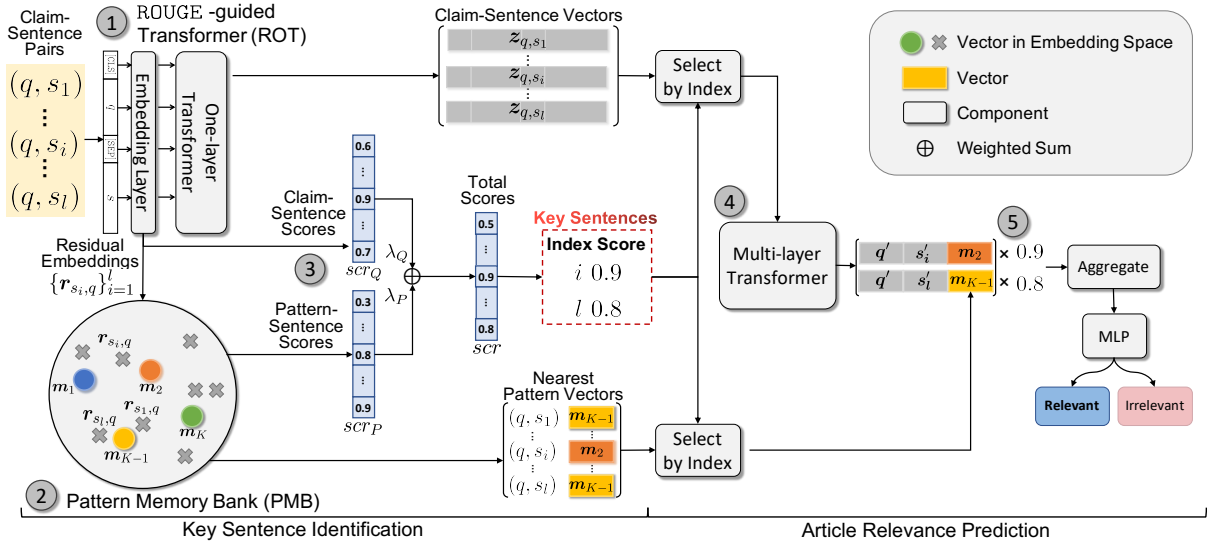

Figure 2: Architecture of MTM. Given a claim q and a candidate article d with l sentences, $s_1, \ldots, s_l$, MTM ① feeds (q, s) pairs into ROUGE-guided Transformer (ROT) to obtain claim-sentence scores in both lexical and semantic aspects; ② matches residual embeddings $\boldsymbol{r}_{s,q}$ with vectors in Pattern Memory Bank (PMB) (here, only four are shown) to obtain pattern-sentence scores; ③ identifies k_2 key sentences by combining the two scores (here, $k_2 = 2$, and s_i and s_l are selected); ④ models interaction among $\boldsymbol{q'}, \boldsymbol{s'}$, and the nearest memory vector $\boldsymbol{m}$ for each key sentence; and ⑤ perform score-weighted aggregation and predict the claim-article relevance.

et al., 2019; Zhang et al., 2021), source credibility (Nguyen et al., 2020), user response (Shu et al., 2019) and diffusion network (Liu and Wu, 2018; Rosenfeld et al., 2020). However, these methods mainly aim at newly emerged claims and do not address those claims that have been fact-checked but continually spread. Our work is in a new thread, **detecting previously fact-checked claims**. Vo and Lee (2020) models interaction between claims and FC-articles by combining GloVe (Pennington et al., 2014) and ELMo embeddings (Peters et al., 2018). Shaar et al. (2020) train a RankSVM with scores from BM25 and Sentence-BERT for relevance prediction. These methods ignore the characteristics of FC-articles, which limits the ranking performance and explainability.

3 Proposed Method

Given a claim q and a candidate set of k_1 FC-articles $\mathcal{D}$ obtained by a standard full-text retrieval model (BM25), we aim to rerank FC-articles truly relevant w.r.t. q at the top by modeling fine-grained relevance between q and each article $d \in \mathcal{D}$. This is accomplished by Memory-enhanced Transformers for Matching (MTM), which conceptually has two steps, (1) Key Sentence Identification and (2) Article Relevance Prediction, see Figure 2. For an article of l sentences, let $\mathcal{S} = \{s_1, \ldots, s_l\}$ be its sentence set. In Step (1), for each sentence, we derive claim-sentence relevance score from ROUGE-guided Transformer (ROT) and pattern-sentence relevance score from Pattern Memory Bank (PMB). The scores indicate how similar the sentence is to the claim and pattern vectors, i.e., how possible to be a key sentence. Top k_2 sentences are selected for more complicated interactions and aggregation with the claim and pattern vectors in Step (2). The aggregated vector is used for the final prediction. We detail the components and then summarize the training procedure below.

3.1 Key Sentence Identification

3.1.1 ROUGE-guided Transformer (ROT)

ROT (left top of Figure. 2) is used to evaluate the relevance between q and each sentence s in $\{\mathcal{S}_i\}_{i=1}^{k_1}$, both lexically and semantically. Inspired by (Gao et al., 2020), we choose to "inject" the ability to consider lexical relevance into the semantic model. As the BERT is proved to capture and evaluate semantic relevance (Zhang et al., 2020), we use a one-layer Transformer initialized with the first block of pretrained BERT to obtain the initial semantic representation of q and s:

$$\boldsymbol{z}_{q,s} = \text{Transformer}\left([\texttt{CLS}]\ q\ [\texttt{SEP}]\ s\right) \quad (1)$$

where [CLS] and [SEP] are preserved tokens and $\boldsymbol{z}_{q,s}$ is the output representation.

To force ROT to consider the lexical relevance, we finetune the pretrained Transformer with the guidance of ROUGE (Lin, 2004), a widely-used metric to evaluate the lexical similarity of two segments in summarization and translation tasks. The intuition is that lexical relevance can be characterized by token overlapping, which ROUGE exactly measures. We minimize the mean square error between the prediction and the precision and recall of ROUGE-2 between q and s ($\mathrm{R}_2 \in \mathbb{R}^2$) to optimize the ROT:

$$\hat{\mathrm{R}}(q,s) = \mathrm{MLP}\big(\boldsymbol{z}_{q,s}([\mathtt{CLS}])\big) \quad (2)$$

$$\mathcal{L}_R = \|\hat{\mathrm{R}}(q,s) - \mathrm{R}_2(q,s)\|_2^2 + \lambda_R \|\Delta\theta\|_2^2 \quad (3)$$

where the first term is the regression loss and the second is to constraint the change of parameters as the ability to capture semantic relevance should be maintained. λ_R is a control factor and $\Delta\theta$ represents the change of parameters.

3.1.2 Pattern Memory Bank (PMB)

The Pattern Memory Bank (PMB) is to generate, store, and update the vectors which represent the common patterns in FC-articles. The vectors in PMB will be used to evaluate pattern-sentence relevance (see Section 3.1.3). Here we detail how to formulate, initialize, and update these patterns below.

Formulation. Intuitively, one can summarize the templates, like "…has been debunked by…", and explicitly do *exact* matching, but the templates are costly to obtain and hard to integrate into neural models. Instead, we *implicitly* represent the common patterns using vectors derived from embeddings of our model, ROT. Inspired by (Wu et al., 2018), we use a memory bank $\mathcal{M}$ to store K common patterns (as vectors), i.e., $\mathcal{M} = \{\boldsymbol{m}_i\}_{i=1}^K$.

Initialization. We first represent each q in the training set and s in the corresponding articles by averaging its token embeddings (from the embedding layer of ROT). Considering that a pattern vector should be *event-irrelevant*, we heuristically remove the event-related part in s as possible by calculating the residual embeddings $\boldsymbol{r}_{s,q}$, i.e., subtracting $\boldsymbol{q}$ from $\boldsymbol{s}$. We rule out the residual embeddings that do not satisfy $t_{low} < \|\boldsymbol{r}_{s,q}\|_2 < t_{high}$, because they are unlikely to contain good pattern information: $\|\boldsymbol{r}_{s,q}\|_2 \le t_{low}$ indicates q and s are highly similar and thus leave little pattern information, while

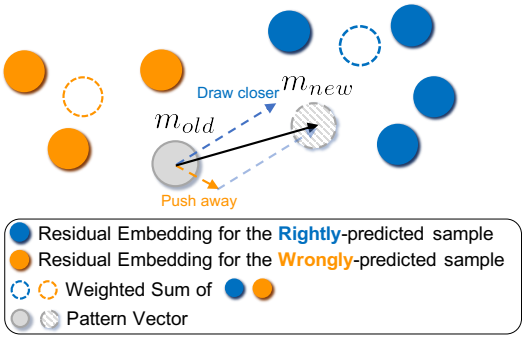

Figure 3: Illustration for Memory Vector Update.

$\|\boldsymbol{r}_{s,q}\|_2 \ge t_{high}$ indicates s may not align with q in terms of the event, so the corresponding $r_{s,q}$ is of little sense. Finally, we aggregate the valid residual embeddings into K clusters using K-means and obtain the initial memory bank $\mathcal{M}$:

$$\mathcal{M} = \text{K-means}\big(\{\boldsymbol{r}_{s,q}^{valid}\}\big) = \{\boldsymbol{m}_1, \ldots, \boldsymbol{m}_K\} \quad (4)$$

where $\{\boldsymbol{r}_{s,q}^{valid}\}$ is the set of valid residual embeddings.

Update. As the initial K vectors may not accurately represent common patterns, we update the memory bank according to the feedbacks of results during training: If the model predicts rightly, the key sentence, say s, should be used to update its nearest pattern vector m. To maintain stability, we use an epoch-wise update instead of an iteration-wise update.

Take updating $\boldsymbol{m}$ as an example. After an epoch, we extract all n key sentences whose nearest pattern vector is $\boldsymbol{m}$ and their n corresponding claims, which is denoted as a tuple set $(\mathcal{S}, \mathcal{Q})^m$. Then $(\mathcal{S}, \mathcal{Q})^m$ is separated into two subsets, $\mathcal{R}^m$ and $\mathcal{W}^m$, which contain n_r and n_w sentence-claim tuples from the rightly and wrongly predicted samples, respectively. The core of our update mechanism (Figure 3) is to draw $\boldsymbol{m}$ closer to the residual embeddings in $\mathcal{R}^m$ and push it away from those in $\mathcal{W}^m$. We denote the i^{th} residual embedding from the two subsets as $\boldsymbol{r}_{\mathcal{R}_i^m}$ and $\boldsymbol{r}_{\mathcal{W}_i^m}$, respectively.

To determine the update direction, we calculate a weighted sum of residual embeddings according to the predicted matching scores. For (s,q), suppose MTM output $\hat{y}_{s,q} \in [0,1]$ as the predicted matching score of q and d (whose key sentence is s), the weight of $\boldsymbol{r}_{s,q}$ is $|\hat{y}_{s,q} - 0.5|$ (denoted as $w_{s,q}$). Weighted residual embeddings are respectively summed and normalized as the components

of the direction vector (Eq. 5):

$$\boldsymbol{u}^{mr} = \left(\sum_{i=1}^{n_r} w_{\mathcal{R}_i^m} \boldsymbol{r}_{\mathcal{R}_i^m}\right), \boldsymbol{u}^{mw} = \left(\sum_{i=1}^{n_w} w_{\mathcal{W}_i^m} \boldsymbol{r}_{\mathcal{W}_i^m}\right) \tag{5}$$

where $\boldsymbol{u}^{mr}$ and $\boldsymbol{u}^{mw}$ are the aggregated residual embeddings. The direction is determined by Eq. 6:

$$\boldsymbol{u}^m = w_r \underbrace{(\boldsymbol{u}^{mr} - \boldsymbol{m})}_{\text{draw closer}} + w_w \underbrace{(\boldsymbol{m} - \boldsymbol{u}^{mw})}_{\text{push away}} \tag{6}$$

where w_r and w_w are the normalized sum of corresponding weights used in Eq. 5 ($w_r + w_w = 1$). The pattern vector m is updated with:

$$\boldsymbol{m}_{new} = \boldsymbol{m}_{old} + \lambda_m \|\boldsymbol{m}_{old}\|_2 \frac{\boldsymbol{u}^m}{\|\boldsymbol{u}^m\|_2} \tag{7}$$

where $\boldsymbol{m}_{old}$ and $\boldsymbol{m}_{new}$ are the memory vector $\boldsymbol{m}$ before and after updating; the constant λ_m and $\|\boldsymbol{m}_{old}\|_2$ jointly control the step size.

3.1.3 Key Sentence Selection

Whether a sentence is selected as a key sentence is determined by combining claim- and pattern-sentence relevance scores. The former is calculated with the distance of q and s trained with ROT (Eq. 8) and the latter uses the distance between the nearest pattern vector in PMB and the residual embedding (Eq. 9). The scores are scaled to $[0, 1]$. For each sentence s in d, the relevance score with q is calculated by Eq. 10:

$$scr_Q(q, s) = \text{Scale}(\|\boldsymbol{r}_{s,q}\|_2) \tag{8}$$

$$scr_P(q, s) = \text{Scale}(\|\boldsymbol{m}_u - \boldsymbol{r}_{s,q}\|_2) \tag{9}$$

$$scr(q, s) = \lambda_Q scr_Q(q, s) + \lambda_P scr_P(q, s) \tag{10}$$

where $\text{Scale}(x) = 1 - \frac{x - min}{max - min}$ and max and min are the maximum and minimum distance of s in d, respectively. $u = \arg\min_i \|\boldsymbol{m}_i - \boldsymbol{r}_{s,q}\|_2$, and λ_Q and λ_P are hyperparameters whose sum is 1.

Finally, sentences with top-k_2 scores, denoted as $\mathcal{K} = \{s_i^{key}(q, d)\}_{i=1}^{k_2}$, are selected as the *key sentences* in d for the claim q.

3.2 Article Relevance Prediction (ARP)

Sentence representation. We model more complicated interactions between the claim and the key sentences by feeding each $z_{q,s^{key}}$ (derived from ROT) into a multi-layer Transformer (MultiTransformer):

$$\boldsymbol{z}'_{q,s^{key}} = \text{MultiTransformer}(\boldsymbol{z}_{q,s^{key}}) \tag{11}$$

Following (Reimers and Gurevych, 2019), we respectively compute the mean of all output token vectors of q and s in $z'_{q,s^{key}}$ to obtain the fixed sized sentence vectors $\boldsymbol{q}' \in \mathbb{R}^{dim}$ and $\boldsymbol{s}^{key\prime} \in \mathbb{R}^{dim}$, where dim is the dimension of a token in Transformers.

Weighted memory-aware aggregation. For final prediction, we use a score-weighted memory-aware aggregation. To make the predictor aware of the pattern information, we append the corresponding nearest pattern vectors to the claim and key sentence vectors:

$$\boldsymbol{v}_i = [\boldsymbol{q}', \boldsymbol{s}_i^{key\prime}(q, d), \boldsymbol{m}_j] \tag{12}$$

where $i = 1, \ldots, k_2$. $j = \arg\min_k \left\|\boldsymbol{m}_k - \boldsymbol{r}_{s_i^{key},q}\right\|_2$.

Intuitively, a sentence with higher score should be attended more. Thus, the concatenated vectors (Eq. 12) are weighted by the relevance scores from Eq. 10 (normalized across the top-k_2 sentences). The weighted aggregating vector is fed into a MLP which outputs the probability that d fact-checks q:

$$scr'(q, s_i^{key}) = \text{Normalize}\big(scr(q, s_i^{key})\big) \tag{13}$$

$$\hat{y}_{q,d} = \text{MLP}\Big(\sum_{i=1}^{k_2} scr'(q, s_i^{key}) \boldsymbol{v}_i\Big) \tag{14}$$

where $\hat{y}_{q,d} \in [0, 1]$. If $\hat{y}_{q,d} > 0.5$, the model predicts that d fact-checks q, otherwise does not. The loss function is cross entropy:

$$\mathcal{L}_M = \text{CrossEntropy}(\hat{y}_{q,d}, y_{q,d}) \tag{15}$$

where $y_{q,d} \in \{0, 1\}$ is the ground truth label. $y_{q,d} = 1$ if d fact-checks q and 0 otherwise. The predicted values are used to rank all k_1 candidate articles retrieved in the first stage.

3.3 Training `MTM`

We summarize the training procedure of `MTM` in Algorithm 1, including the pretraining of ROT, the initialization of PMB, the training of ARP, and the epoch-wise update of PMB.

Algorithm 1 MTM Training Procedure

Input: Training set $\mathcal{T} = [(q_0, d_{00}), \ldots, (q_0, d_{0k_1}), \ldots, (q_n, d_{nk_1})]$ where the k_1 candidate articles for each claim are retrieved by BM25.

Pre-train ROUGE-guided Transformer.
Initialize the Pattern Memory Bank (PMB).
for each epoch **do**
  for (q, d) in $\mathcal{T}$ **do**
    // Key Sentence Identification
    Calculate $scr_Q(q, s)$ via ROT and $scr_P(q, s)$ via PMB.
    Calculate $scr(q, s)$ using Eq.10.
    Select key sentences $\mathcal{K}$.
    // Article Relevance Prediction (ARP)
    Calculate $\boldsymbol{v}$ for each s in $\mathcal{K}$ and $\hat{y}_{q,d}$.
    Update the ARP to minimize $\mathcal{L}_M$.
  end for
  Update the PMB using Eq. 7.
end for

4 Experiments

In this section, we mainly answer the following experimental questions:

EQ1: Can MTM improve the ranking performance of FC-articles given a claim?

EQ2: How effective are the components of MTM, including ROUGE-guided Transformer, Pattern Memory Bank, and weighted memory-aware aggregation in Article Relevance Prediction?

EQ3: To what extent can MTM identify key sentences in the articles, especially in the longer ones?

4.1 Data

We conducted the experiments on two real-world datasets. Table 1 shows the statistics of the two datasets. The details are as follows:

Twitter Dataset

The Twitter[4] dataset is originated from (Vo and Lee, 2019) and processed by Vo and Lee (2020). The dataset pairs the claims (tweets) with the corresponding FC-articles from Snopes. For tweets with images, it appends the OCR results to the tweets. We remove the manually normalized claims in Snopes' FC-articles to adapt to more general scenarios. The data split is the same as that in (Vo and Lee, 2020).

Weibo Dataset

We built the first Chinese dataset for the task of detecting previously fact-checked claims in this article. The claims are collected from Weibo and the FC-articles are from *multiple fact-checking sources* including Jiaozhen, Zhuoyaoji[5], etc. We recruited annotators to match claims and FC-articles based on basic search results. Appendix A introduce the details.

Table 1: Statistics of the Twitter and the Weibo dataset. #: Number of. C-A Pairs: Claim-article pairs.

Dataset	Twitter			Weibo		
	Train	Val	Test	Train	Val	Test
#Claim	8,002	1,000	1,001	8,356	1,192	2,386
#Articles	1,703	1,697	1,697	17,385	8,353	11,715
C-A Pairs	8,025	1,002	1,005	28,596	3,337	7,245
Relevant Fact-checking Articles Per Claim						
Average	1.003	1.002	1.004	3.422	2.799	3.036
Medium	1	1	1	2	1	2
Maximum	2	2	2	50	18	32

4.2 Baseline Methods

BERT-based rankers from general IR tasks

BERT (Devlin et al., 2019): A method of pre-training language representations with a family of pretrained models, which has been used in general document reranking to predict the relevance. (Nogueira and Cho, 2019; Akkalyoncu Yilmaz et al., 2019)

DuoBERT (Nogueira et al., 2019): A popular BERT-based reranker for multi-stage document ranking. Its input is a query and a pair of documents. The pairwise scores are aggregated for final document ranking. Our first baseline, BERT (trained with query-article pairs), provides the inputs for DuoBERT.

BERT(Transfer): As no sentence-level labels are provided in most document retrieval datasets, Yang et al. (2019) finetune BERT with short text matching data and then apply to score the relevance between query and each sentence in documents. The three highest scores are combined with BM25 score for document-level prediction.

Rankers from related works of our task

Sentence-BERT: Shaar et al. (2020) use pre-trained Sentence-BERT models to calculate cosine similarity between each sentence and the given claim. Then the top similarity scores are fed into a neural network to predict document relevance.

RankSVM: A pairwise RankSVM model for reranking using the scores from BM25 and sentence-BERT (mentioned above), which achieves the best results in (Shaar et al., 2020).

[4] https://twitter.com

[5] https://piyao.sina.cn

Table 2: Performance of baselines and MTM. Best results are in **boldface**.

Method	Selecting Sentences?	Weibo						Twitter					
		MRR	MAP@			HIT@		MRR	MAP@			HIT@	
			1	3	5	3	5		1	3	5	3	5
BM25		0.709	0.355	0.496	0.546	0.741	0.760	0.522	0.460	0.489	0.568	0.527	0.568
BERT		0.834	0.492	0.649	0.693	0.850	0.863	0.895	0.875	0.890	0.890	0.908	0.909
DuoBERT		0.885	0.541	0.713	0.756	0.886	0.887	0.923	**0.921**	0.922	0.922	0.923	0.923
BERT(Transfer)	✓	0.714	0.361	0.504	0.553	0.742	0.764	0.642	0.567	0.612	0.623	0.668	0.719
Sentence-BERT	✓	0.750	0.404	0.543	0.589	0.810	0.861	0.794	0.701	0.775	0.785	0.864	0.905
RankSVM	✓	0.809	0.408	0.607	0.661	0.887	0.917	0.846	0.778	0.832	0.840	0.898	0.930
CTM		0.856	0.356	0.481	0.525	0.894	0.935	0.926	0.889	0.919	0.922	0.952	0.964
MTM	✓	**0.902**	**0.542**	**0.741**	**0.798**	**0.934**	**0.951**	**0.931**	0.899	**0.926**	**0.928**	**0.957**	**0.967**

Table 3: Ablation study of MTM. Best results are in **boldface**. AG: Ablation Group.

AG	Variant	Weibo						Twitter					
		MRR	MAP@			HIT@		MRR	MAP@			HIT@	
			1	3	5	3	5		1	3	5	3	5
-	MTM	**0.902**	**0.542**	**0.741**	**0.798**	0.934	0.951	**0.931**	0.899	**0.926**	**0.928**	**0.957**	**0.967**
1	*w/o ROUGE guidance*	0.892	0.535	0.729	0.786	0.925	0.943	0.929	**0.905**	0.924	0.926	0.945	0.952
2	*w/ rand mem init*	0.879	0.516	0.700	0.753	0.912	0.935	0.897	0.860	0.890	0.893	0.922	0.938
	w/o mem update	0.898	0.541	0.736	0.790	0.935	0.948	0.925	0.897	0.860	0.890	0.922	0.938
	w/o PMB	0.897	0.537	0.734	0.792	0.931	0.948	0.920	0.885	0.913	0.917	0.944	0.960
3	*w/ avg. pool*	0.901	0.540	0.739	0.796	**0.938**	**0.958**	0.923	0.892	0.917	0.919	0.944	0.954
	w/o pattern aggr.	0.896	0.535	0.734	0.791	0.930	0.945	0.922	0.890	0.917	0.919	0.947	0.954

CTM (Vo and Lee, 2020): This method leverages GloVe and ELMo to jointly represent the claims and the FC-articles for predicting the relevance scores. Its multi-modal version is not included as MTM focuses on key textual information.

4.3 Experimental Setup

Evaluation Metrics. As this is a binary retrieval task, we follow Shaar et al. (2020) and report Mean Reciprocal Rank (MRR), Mean Average Precision@k (MAP@k, $k = 1, 3, 5$) and HIT@k ($k = 3, 5$). See equations in Appendix B.

Implementation Details. In MTM, the ROT and ARP components have one and eleven Transformer layers, respectively. The initial parameters are obtained from pretrained BERT models[6]. Other parameters are randomly initialized. The dimension of claim and sentence representation in ARP and pattern vectors are 768. Number of Clusters in PMB K is 20. Following (Shaar et al., 2020) and (Vo and Lee, 2020), we use $k_1 = 50$ candidates retrieved by BM25. $k_2 = 3$ (Weibo, hereafter, W) / 5 (Twitter, hereafter, T) key sentences are selected. We use Adam (P. Kingma and Ba, 2015) for optimization with $\epsilon = 10^{-6}, \beta_1 = 0.9, \beta_2 = 0.999$. The learning rates are 5×10^{-6} (W) and 1×10^{-4} (T). The batch size is 512 for pretraining ROT, 64 for the main task. According to the quantiles on training sets, we set $t_{low} = 0.252$ (W) / 0.190 (T), $t_{high} = 0.295$ (W) / 0.227 (T). The following hyperparameters are selected according to the best validation performance: $\lambda_R = 0.01$ (W) / 0.05 (T), $\lambda_Q = 0.6$, $\lambda_P = 0.4$, and $\lambda_m = 0.3$. The maximum epoch is 5. All experiments were conducted on NVIDIA V100 GPUs with PyTorch (Paszke et al., 2019). The implementation details of baselines are in Appendix C.

[6]We use `bert-base-chinese` for Weibo and `bert-base-uncased` for Twitter.

4.4 Performance Comparison

To answer **EQ1**, we compared the performance of baselines and our method on the two datasets, as shown in Table 2. We see that: (1) MTM ourperforms all compared methods on the two datasets (the exception is only the MAP@1 on Twitter), which indicates that it can effectively find related FC-articles and provide evidence for determining if a claim is previously fact-checked. (2) For all methods, the performance on Weibo is worse than that on Twitter because the Weibo dataset contains more claim-sentence pairs (from multiple sources) than Twitter and is more challenging. Despite this, MTM's improvement is significant. (3) BERT(Transfer), Sentence-BERT and RankSVM use transferred sentence-level knowledge from other pretext tasks but did not outperform the document-level BERT. This is because FC-articles have their own characteristics, which may not be covered by transferred knowledge. In con-

trast, our observed characteristics help MTM achieve good performance. Moreover, MTM is also efficiency compared to BERT(Transfer), which also uses 12-layer BERT and selects sentences, because our model uses only one layer for all sentences (other 11 layers are for key sentences), while all sentences are fed into the 12 layers in BERT(Transfer).

4.5 Ablation Study

To answer **EQ2**, we evaluated three ablation groups of MTM's variants (AG1∼AG3) to investigate the effectiveness of the model design.[7] Table 3 shows the performance of variants and MTM.

AG1: With vs. Without ROUGE. The variant removes the guidance of ROUGE (MTM *w/o ROUGE guidance*) to check the effectiveness of ROUGE-guided finetuning. The variant performs worse on Weibo, but MAP@1 slightly increases on Twitter. This is probably because there are more lexical overlapping between claims and FC-articles in the Weibo dataset, while most of the FC-articles in the Twitter dataset choose to summarize the claims to fact-check.

AG2: Cluster-based Initialization vs. Random Initialization vs. Without update vs. Without PMB. The first variant (MTM *w/ rand mem init*) uses random initialization and the second (MTM *w/o mem update*) uses pattern vectors without updating. The last one (MTM *w/o PMB*) removes the PMB. We see that the variants all perform worse than MTM on MRR, of which *w/ rand mem init* performs the worst. This indicates that cluster-based initialization provides a good start and facilitates the following updates while the random one may harm further learning.

AG3: Score-weighted Pooling vs. Average pooling, and With vs. Without pattern vector. The first variant, MTM *w/ avg. pool*, replace the score-weighted pooling with average pooling. The comparison in terms of MRR and MAP shows the effectiveness of using relevance scores as weights. The second, MTM *w/o pattern aggr.*, does not append the pattern vector to claim and sentence vectors before aggregation. It yields worse results, indicating the patterns should be taken into consideration for final prediction.

[7] We do not run MTM without sentence selection due to its high computational overhead which makes it unfeasible for training and inference.

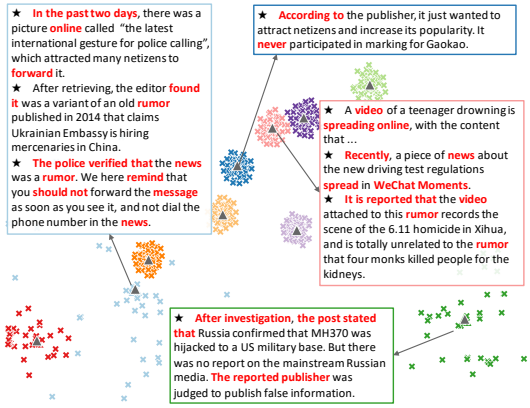

Figure 4: Visualization of pattern vectors (▲) and near residual embeddings (✖). The sentences are translated from Chinese.

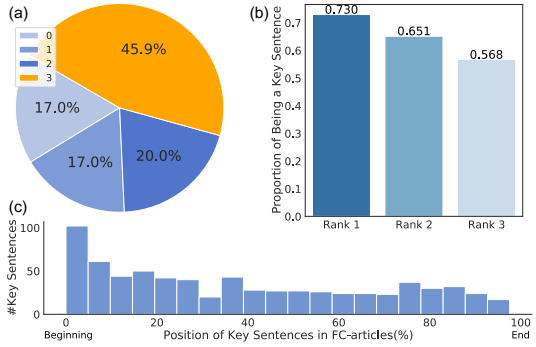

Figure 5: Results of human evaluation. (a)The proportion of the FC-articles where MTM found $\{0, 1, 2, 3\}$ key sentences. (b) The proportion of key sentences at rank $\{1, 2, 3\}$. (c) The positional distribution of key sentences in the FC-articles.

4.6 Visualization of Memorized Patterns

To probe what the PMB summarizes and memorizes, we selected and analyzed the key sentences corresponding to the residual embeddings around pattern vectors. Figure 4 shows example sentences where highly frequent words are in boldface. These examples indicate that the pattern vectors do cluster key sentences with common patterns like "...spread in WeChat Moments".

4.7 Human Evaluation and Case Study

The quality of selected sentences cannot be automatically evaluated due to the lack of sentence-level labels. To answer **EQ3**, we conducted a human evaluation. We randomly sampled 370 claim-article pairs whose articles were with over 20 sentences from the Weibo dataset. Then we showed each claim and top three sentences selected from the corresponding FC-article by MTM. Three anno-

Claim	Is this to make the so-called artificial eggs? Surprising! #video
Key Sentences	
KS1. **Recently, a short video that claims** a production process of the artificial fake eggs **has widely spread in WeChat Groups.**	
KS2. **The reporter of Shanghai Observer found that** the video actually recorded making toy eggs, which were not to pretend as real eggs for sale.	
KS3. Relating the video of toy egg production to food safety issues **is just a gimmick used by spreaders.**	
Claim	State FDA: 60% of the drugs will be stopped selling within 2 or 3 years and will be replaced by nutraceutical industry. State will invest 8 trillion!
Key Sentences	
KS1. **It's been reported that FDA has proposed** that 60% of the drugs will be stopped selling within the next 2 or 3 years and replaced by nutraceuticals and foods.	
KS2. **The reporter visited** the website of the FDA but found no such official documents, indicating **the details in the claim were purely fabricated.**	
KS3. **It's verified that** the claim that nutraceuticals will replace drugs is a malicious propaganda by companies to confuse the netizens.	

Figure 6: Cases in the set of human evaluation. Quotations are underlined and patterns are in **boldface**.

tators were asked to check if an auto-selected sentence helped match the given query and the source article (i.e., key sentences). Figure 5 shows (a) MTM hit at least one key sentence in 83.0% of the articles; (b) 73.0% of the sentences at Rank 1 are key sentences, followed by 65.1% at Rank 2 and 56.8% at Rank 3. This proves that MTM can find the key sentences in long FC-articles and provide helpful explanations. We also show the positional distribution in Figure 5(c), where key sentences are scattered throughout the articles. Using MTM to find key sentences can save fact-checkers' time to scan these long articles for determining whether the given claim was fact-checked.

Additionally, we exhibit two cases in the evaluation set in Figure 6. These cases prove that MTM found the key sentences that correspond to the characteristics described in Section 1. Please refer to Appendix D for further case analysis.

5 Conclusions

We propose MTM to select from fact-checked articles key sentences that introduce or debunk claims. These auto-selected sentences are exploited in an end-to-end network for estimating the relevance of the fact-checked articles w.r.t. a given claim. Experiments on the public Twitter dataset and the private Weibo dataset show that MTM outperforms the state of the art. Moreover, human evaluation and case studies demonstrate that the selected sentences provide helpful explanations of the results.

Acknowledgments

The authors thank Guang Yang, Tianyun Yang, Peng Qi and anonymous reviewers for their insightful comments. Also, we thank Rundong Li, Qiong Nan, and other annotators for their efforts. This work was supported by the National Key Research and Development Program of China (2017YFC0820604), the National Natural Science Foundation of China (U1703261), and the Fundamental Research Funds for the Central Universities and the Research Funds of Renmin University of China (No. 18XNLG19). The corresponding authors are Juan Cao and Xirong Li.

Broader Impact Statement

Our work involves two scenarios that need the ability to detect previously fact-checked claims: (1) For social media platforms, our method can check whether a newly published post contains false claims that have been debunked. The platform may help the users to be aware of the text's veracity by providing the key sentences selected from fact-checking articles and their links. (2) For manual or automatic fact-checking systems, it can be a filter to avoid redundant fact-checking work. When functioning well, it can assist platforms, users, and fact-checkers to maintain more credible cyberspace. But in the failure cases, some well-disguised claims may escape. This method functions with reliance on the used fact-checking article databases. Thus, authority and credibility need to be carefully considered in practice. We did our best to make the new Weibo dataset for academic purpose reliable. Appendix A introduces more details.

References

Oluwaseun Ajao, Deepayan Bhowmik, and Shahrzad Zargari. 2019. Sentiment Aware Fake News Detection on Online Social Networks. In *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2507–2511.

Zeynep Akkalyoncu Yilmaz, Shengjin Wang, Wei Yang, Haotian Zhang, and Jimmy Lin. 2019. Applying BERT to Document Retrieval with Birch. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 19–24, Hong Kong, China. Association for Computational Linguistics.

Wanxiang Che, Yijia Liu, Yuxuan Wang, Bo Zheng, and Ting Liu. 2018. Towards Better UD Parsing:

Deep Contextualized Word Embeddings, Ensemble, and Treebank Concatenation. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 55–64, Brussels, Belgium. Association for Computational Linguistics.

Qingqing Chen. Coronavirus rumors trigger irrational behaviors among chinese netizens [online]. 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Duke Reporters' Lab. Fact-checking count tops 300 for the first time [online]. 2020.

Murhaf Fares, Andrey Kutuzov, Stephan Oepen, and Erik Velldal. 2017. Word vectors, reuse, and replicability: Towards a community repository of large-text resources. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 271–276, Gothenburg, Sweden. Association for Computational Linguistics.

Marc Fisher, John Woodrow Cox, and Peter Hermann. 2016. Pizzagate: From rumor, to hashtag, to gunfire in DC. *Washington Post*, 6.

Luyu Gao, Zhuyun Dai, Zhen Fan, and Jamie Callan. 2020. Complementing Lexical Retrieval with Semantic Residual Embedding. *arXiv*, arXiv:2004.13969. Version 2.

Thorsten Joachims. 2006. Training Linear SVMs in Linear Time. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 217–226, New York, NY, USA. Association for Computing Machinery.

Shen Li, Zhe Zhao, Renfen Hu, Wensi Li, Tao Liu, and Xiaoyong Du. 2018. Analogical Reasoning on Chinese Morphological and Semantic Relations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 138–143, Melbourne, Australia. Association for Computational Linguistics.

Xirong Li, Tiberio Uricchio, Lamberto Ballan, Marco Bertini, Cees G. M. Snoek, and Alberto Del Bimbo. 2016. Socializing the Semantic Gap: A Comparative Survey on Image Tag Assignment, Refinement, and Retrieval. *ACM Computing Surveys*, 49(1).

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Jimmy Lin, Miles Efron, Yulu Wang, and Garrick Sherman. 2014. Overview of the TREC-2014 Microblog Track. In *Twenty-Third Text REtrieval Conference (TREC 2014) Proceedings*.

Xin Liu, Qingcai Chen, Chong Deng, Huajun Zeng, Jing Chen, Dongfang Li, and Buzhou Tang. 2018. LCQMC:A Large-scale Chinese Question Matching Corpus. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1952–1962, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Yang Liu and Yi-Fang Wu. 2018. Early Detection of Fake News on Social Media through Propagation Path Classification with Recurrent and Convolutional Networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pages 354–361. AAAI Press.

Van-Hoang Nguyen, Kazunari Sugiyama, Preslav Nakov, and Min-Yen Kan. 2020. FANG: Leveraging Social Context for Fake News Detection Using Graph Representation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 1165–1174, New York, NY, USA. Association for Computing Machinery.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *arXiv*, arXiv:1901.04085. Version 5.

Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-stage document ranking with BERT. *arXiv*, arXiv:1910.14424. Version 1.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, volume 32, pages 8026–8037. Curran Associates, Inc.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages

2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Kashyap Popat, Subhabrata Mukherjee, Jannik Strötgen, and Gerhard Weikum. 2018a. CredEye: A Credibility Lens for Analyzing and Explaining Misinformation. In *Companion Proceedings of the The Web Conference 2018*, pages 155–158, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Kashyap Popat, Subhabrata Mukherjee, Andrew Yates, and Gerhard Weikum. 2018b. DeClarE: Debunking Fake News and False Claims using Evidence-Aware Deep Learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 22–32, Brussels, Belgium. Association for Computational Linguistics.

Piotr Przybyla. 2020. Capturing the Style of Fake News. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 490–497. AAAI Press.

Peng Qi, Juan Cao, Tianyun Yang, Junbo Guo, and Jintao Li. 2019. Exploiting Multi-domain Visual Information for Fake News Detection. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 518–527. IEEE.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making Monolingual Sentence Embeddings Multilingual using Knowledge Distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online. Association for Computational Linguistics.

Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389.

Nir Rosenfeld, Aron Szanto, and David C. Parkes. 2020. A Kernel of Truth: Determining Rumor Veracity on Twitter by Diffusion Pattern Alone. In *Proceedings of The Web Conference 2020*, page 1018–1028, New York, NY, USA. Association for Computing Machinery.

Shaden Shaar, Nikolay Babulkov, Giovanni Da San Martino, and Preslav Nakov. 2020. That is a Known Lie: Detecting Previously Fact-Checked Claims. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3607–3618, Online. Association for Computational Linguistics.

Kai Shu, Limeng Cui, Suhang Wang, Dongwon Lee, and Huan Liu. 2019. dEFEND: Explainable Fake News Detection. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, page 395–405, New York, NY, USA. Association for Computing Machinery.

TensorFlow. Classification on imbalanced data [online]. 2021.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018. The Fact Extraction and VERification (FEVER) Shared Task. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 1–9, Brussels, Belgium. Association for Computational Linguistics.

Nguyen Vo and Kyumin Lee. 2019. Learning from Fact-checkers: Analysis and Generation of Fact-checking Language. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 335–344. Association for Computing Machinery.

Nguyen Vo and Kyumin Lee. 2020. Where Are the Facts? Searching for Fact-checked Information to Alleviate the Spread of Fake News. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7717–7731, Online. Association for Computational Linguistics.

Jingqiong Wang and Xinzhu Li. 2011. Radiation fears prompt panic buying of salt. *China Daily*, 18:2011–03.

Xuezhi Wang, Cong Yu, Simon Baumgartner, and Flip Korn. 2018. Relevant Document Discovery for Fact-Checking Articles. In *Companion Proceedings of the The Web Conference 2018*, pages 525–533, Lyon, France. International World Wide Web Conferences Steering Committee.

Wikipedia. Mean reciprocal rank [online]. 2021.

Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. 2018. Unsupervised Feature Learning via Non-Parametric Instance Discrimination. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3733–3742.

Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. 2020. Unsupervised Data Augmentation for Consistency Training. In *Advances in Neural Information Processing Systems*, volume 33. Curran Associates, Inc.

Xinhua Net. Three common types of internet rumors present a new trend of visual spread [online]. 2019. in Chinese.

Wei Yang, Haotian Zhang, and Jimmy Lin. 2019. Simple Applications of BERT for Ad Hoc Document Retrieval. *arXiv*, arXiv:1903.10972. Version 1.

Xiwang Yang, Harald Steck, Yang Guo, and Yong Liu. 2012. On Top-k Recommendation Using Social Networks. In *Proceedings of the Sixth ACM Conference on Recommender Systems*, pages 67–74, New York, NY, USA. Association for Computing Machinery.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations*.

Xueyao Zhang, Juan Cao, Xirong Li, Qiang Sheng, Lei Zhong, and Kai Shu. 2021. Mining Dual Emotion for Fake News Detection. In *Proceedings of the The Web Conference 2021*. International World Wide Web Conferences Steering Committee.

Xing Zhou, Juan Cao, Zhiwei Jin, Fei Xie, Yu Su, Dafeng Chu, Xuehui Cao, and Junqiang Zhang. 2015. Real-Time News Certification System on Sina Weibo. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15 Companion, pages 983–988, New York, NY, USA. Association for Computing Machinery.

A Constructing the New Weibo Dataset

To construct datasets for fact-checked claim detection on social media, we need to (1) collect the fact-checked claims (social media posts); (2) collect fact-checking articles (FC-articles); and (3) generate claim-article pairs.

Collection. In Step (1), we used posts whose labels are *fake* from the datasets for fake news detection (Zhang et al., 2021; Zhou et al., 2015), because their labels were determined by fact-checking. In Step (2), we crawled fact-checking articles from multiple sources to enrich the article base. The sources are partially listed in Table 4 due to the space limit. For the claims and articles which contained much text in the attached images, we recognized the text using OCR service on Baidu AI platform[8]. Note that we only crawled the claims and articles that were publicly available at the crawling time. To protect privacy, the publishers' names were removed. However, we preserved names and offensive words in the main text because they were crucial for summarizing the events and performing the matching process.

[8] https://ai.baidu.com/tech/ocr

Annotation. In Step (3), we performed a model-assisted human annotation. We first duplicated the data collected in Step (1) and (2) and then used BM25 to retrieve the relevant FC-articles as candidates with the claims as queries. Twenty-six annotators (postgraduates) were instructed (by a Chinese guideline with examples written by the first author) to check whether the candidates did fact-check the given claims. We dropped the claims that are annotated as irrelevant to all candidates. For claims that were with highly overlapping candidates but different annotation results, the authors manually checked and corrected the wrongly annotated samples.

B Calculation of Evaluation Metrics

Assume that query set Q has $|Q|$ queries and the i^{th} query has n_i relevant documents. We calculate the evaluation metrics using the following equations:

$$\text{MRR} = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i} \tag{16}$$

where $rank_i$ refers to the rank position of the first relevant answer for the i^{th} query in the corresponding retrieving result. (Wikipedia, 2021)

$$\text{MAP@}k = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{n_i}\sum_{j=1}^{n_i}P_i(j)rel_i(j) \tag{17}$$

where $P_i(j)$ is the proportion of returned documents in the top-j set for the i^{th} query that are relevant. $rel_i(j)$ is an indicator function equaling 1 if the document at rank j in the returned list for the i^{th} query is relevant and 0 otherwise. (Li et al., 2016)

$$\text{HIT@}k = \frac{1}{|Q|}\sum_{i=1}^{|Q|}has_i(k) \tag{18}$$

where $has_i(k)$ is an indicator function equaling 1 if $rank_i \leq k$ and 0 otherwise. (Yang et al., 2012)

Note that we guarantee that a query has at least one relevant document in its candidate list, so the corner case of empty ground truth set is ignored.

C Implementation of BM25 and Baselines

BM25: The articles were indexed with `gensim` (Řehůřek and Sojka, 2010).

Table 4: Part of the Sources of fact-checking articles in the Weibo dataset.

Source	Description	URL
Jiaozhen	A fact-checking platform operated by Tencent.	https://fact.qq.com/, https://new.qq.com/omn/author/5107513
Liuyanbaike	A debunking website operated by Guokr.	http://www.liuyanbaike.com/
Baidu Piyao	A fact-checking account operated by Baidu.	https://author.baidu.com/home?app_id=15060
ScienceFacts	A platform to fact-check scientific claims supported by China Association for Science and Technology	https://piyao.kepuchina.cn/
Qiuzhen	A fact-checking column of People's Daily Online	http://society.people.com.cn/GB/229589/index.html
Dingxiang Doctor	A platform for doctors and experts in life science	https://dxy.com/
China Joint Internet Rumor-Busting Platform	A platform operated by Cyberspace Administration of China	http://www.piyao.org.cn/
Zhuoyaoji	Sina News official fact-checking account	http://piyao.sina.cn/, https://weibo.com/u/6590980486
Weibo Piyao	Weibo official fact-checking account	https://weibo.com/weibopiyao

BERT: We finetuned the last Transformer layer of `bert-base-chinese` for Chinese and `bert-base-uncased` for English. Following the commonly used strategy (e.g., Xie et al., 2020), we truncated the sequences to the maximum length of 512. The maximum length of claims is the same as `MTM` and the rest tokens are from articles.

DuoBERT : We used top 20 articles from the results of BERT as candidates to construct article pairs. For each article, the score is obtained by summing its pairwise scores. The used pretrained models are the same as BERT (mentioned above) and we finetuned the layers except the embedding layer and the first Transformer layer.

BERT(Transfer): For the Twitter data, we used the models provided in Birch (Akkalyoncu Yilmaz et al., 2019) that was finetuned on TREC Microblog Track data (Lin et al., 2014); for the Weibo data, we used LCQMC dataset (Liu et al., 2018) containing 260,068 text pairs to finetune `bert-based-chinese` for 20 epochs. Considering the value difference between BM25 and BERT scores, the weight of BM25 score was learned by grid search in $[0, 1]$ but the weights of others were in $[0, 5]$. The step size was 0.1. We got the best results with BM25 weight $= 0.2$ (Weibo) / 0.1 (Twitter) and the weights of top-3 sentences $= 1.2, 0.4, 0.9$ (Weibo) / $4.8, 4, 2.5$ (Twitter), respectively.

Sentence-BERT: We used the base versions in `Sentence-Transformers` (Reimers and Gurevych, 2019) to obtain the embeddings against the claims and sentences. Specifically, we used `stsb-xlm-r-multilingual` (Reimers and Gurevych, 2020) for the Weibo data and `stsb-bert-base` for Twitter[9] . According to Shaar et al. (2020), we calculated the cosine similarity of each claim-sentence pair and fed the top-5 scores into a simple neural network (20-ReLU-10-ReLU) for classification. We trained the model for 20 epochs with class weighted cross entropy as the loss function. The class weights were calculated across the dataset (TensorFlow, 2021).

RankSVM: We combined the scores and their reciprocal ranks obtained from Sentence-BERT models and BM25. Then we fed them into a RankSVM[10] (Joachims, 2006) for classification. We used Sentence-BERT models trained with $\{3, 4, 5, 6\}$ sentences for Twitter and those trained with $\{6, 7, 8, 9\}$ sentences for Weibo. We kept the default settings in the package.

CTM: For the Twitter dataset, we followed (Vo and Lee, 2020) to use `glove.6B`[11] (Pennington et al., 2014) and the `ELMo Original (5.5B)`[12] (Peters et al., 2018); for the Weibo data, we used `sgns.weibo.bigram-char`[13] (Li et al., 2018) and `simplified-Chinese`

[9] https://www.sbert.net/docs/pretrained_models.html

[10] http://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html

[11] https://nlp.stanford.edu/projects/glove/

[12] https://allennlp.org/elmo

[13] https://github.com/Embedding/Chinese-Word-Vectors

Claim	As reported by Korean People's Daily, Jae-Seo Jung, professor of Ewha Womans University Korean, refutes the claim that Cao Cao's tomb is at Anyang. According to his research on Chinese and Korean history, Professor Jung finds that Cao Cao is a Korean.
Auto-selected Sentences by MTM	
S1. (The only key sentence) **Some Chinese media quoted** the Korean People's Daily as saying that Professor Jae-Seo Jung claimed that Cao Cao is a Korean.	
S2. (Not key sentence) **Some Chinese media quoted** Korean Daily's news, which said that Professor Huanjing Park of Sungkyunkwan University published a report saying that Sun Yat-Sen, the founding father of modern China, was a Korean.	
S3. (Not key sentence) **According to the report,** Cheng-Soo Park, a history professor at Seoul University in South Korea, said that after ten years of research, he believed that it was the Korean people who first invented Chinese characters. Later, the Korean people brought Chinese characters to China, forming the present Chinese culture.	

Figure 7: A case with only one key sentence being hit by MTM. Patterns are in **boldface**.

Claim	I noticed it in my WeChat Moments. It was very sad, but I still hope this is fake! [The plane crashed in Vietnam. All the people on board were probably dead!] CNN reported that the MH370 was confirmed to have fallen within 100 kilometers north of Ho Chi Minh City, Vietnam. Because of the rainstorm, the local people thought it was a falling meteorite. At present, it is still raining in the local area. As it is a mountainous area, so it is difficult to carry out the search and rescue work.
Auto-selected Sentences by MTM	
S1. (Not key sentence) On the evening of the 8th, a short message purportedly from "Vietnam News Agency" said: "Vietnam News Agency Express at 19:32 on March 8th: 17 hours after Malaysia Airlines flight MH370 lost contact, it was found by Philippine maritime vessels carrying out search and rescue mission in the sea area of 06 55 15" N and 103 34 43 "E.	
S2. (Not key sentence) Since then, Boeing China President deleted the Weibo post, saying that "the plane has been found" is the wrong message, and the search continues.	
S3. (Not key sentence) On the afternoon of the 8th, the South China Sea Rescue Bureau said that it was a misunderstanding that the two search and rescue vessels previously reported by the media set out from Xisha and Haikou at 10:49 and 11:30 respectively.	
Ground Truth Key Sentences	
GT1. CNN did not release the news that the losing-contact airplane crashed.	
GT2. On the 8th of this month, it was spread online that "CNN said that the flight MH370 crashed in Vietnam".	
GT3. CNN's official account on Twitter is still using the term "lost contact", and the TV lives also use "missing" to modify MH370.	

Figure 8: A case with no key sentence being hit by MTM.

ELMo[14] (Che et al., 2018; Fares et al., 2017). We kept the default settings provided by the authors[15].

D Further Case Analysis

We reviewed the fact-checking articles in the set for human evaluation wherein MTM hit less than two key sentences. We here exhibit two situations that make MTM did not perform well: (1) In Figure 7, the claim is about where Cao Cao was born. MTM found three sentences with significant patterns (shown in boldface). However, only S1 is related to the claim. S2 and S3 introduce similar but irrelevant claims. This is because that the fact-checking article is actually a collection of rumors about South Korea on the Chinese social media. The claims in this article are all similar to each other, and thus, to differentiate them needs more delicate semantic understanding. (2) Figure 8 shows a case where MTM found no key sentence from the article. We append the key sentences selected manually below. We speculate that the failure is due to the length of the given claim. The claim is longer than general posts on Weibo and contains many details, making the model lose focus on the key elements of the event description. Thus, S1 describing another news about MH370's activity in Vietnam was selected, instead of the ground truth sentences. To achieve better performance, future work may consider improving the semantic modeling and summarizing key information from both fact-checking articles and claims.

[14] https://github.com/HIT-SCIR/ELMoForManyLangs

[15] https://github.com/nguyenvo09/EMNLP2020