A Multi-Level Attention Model for Evidence-Based Fact Checking

Canasai Kruengkrai **Junichi Yamagishi** **Xin Wang**
National Institute of Informatics, Japan
{canasai,jyamagishi,wangxin}@nii.ac.jp

Abstract

Evidence-based fact checking aims to verify the truthfulness of a claim against evidence extracted from textual sources. Learning a representation that effectively captures relations between a claim and evidence can be challenging. Recent state-of-the-art approaches have developed increasingly sophisticated models based on graph structures. We present a simple model that can be trained on sequence structures. Our model enables inter-sentence attentions at different levels and can benefit from joint training. Results on a large-scale dataset for Fact Extraction and VERification (FEVER) show that our model outperforms the graph-based approaches and yields 1.09% and 1.42% improvements in label accuracy and FEVER score, respectively, over the best published model.[1]

1 Introduction

False or misleading claims spread through online media faster and wider than the truth (Vosoughi et al., 2018). False claims can occur in many different forms, e.g., fake news, rumors, hoaxes, propaganda, etc. Identifying false claims that are likely to cause harm in the real world is important. Generally, claims can be categorized into two types: verifiable and unverifiable. Verifiable claims can be confirmed to be true or false as guided by evidence from credible sources, while unverifiable claims cannot be confirmed due to insufficient information.

Verifying the truthfulness of a claim with respect to evidence can be regarded as a special case of recognizing textual entailment (RTE) (Dagan et al., 2006) or natural language inference (NLI) (Bowman et al., 2015), where the premise (evidence) is not given. Thus, the task of claim verification is to first retrieve documents relevant to a given claim from textual sources, then select sentences likely to contain evidence, and finally assign a veracity relation label to support or refute the claim. For example, the false claim "*Rabies is a foodborne illness.*" can be refuted by the evidence "*Rabies is spread when an infected animal scratches or bites another animal or human.*" extracted from the Wikipedia article "*Rabies*". Figure 1 shows other examples that require multiple evidence sentences to support or refute claims. All of these claims are taken from a benchmark dataset for Fact Extraction and VERification (FEVER) (Thorne et al., 2018). A key challenge is to obtain a representation for claim and evidence sentences that can effectively capture relations among them.

ID: 8143
Claim: Moscovium is a transactinide element.
Label: SUPPORTED
Evidence: [*Moscovium*]
Moscovium is a superheavy synthetic element with symbol Mc and atomic number 115.[0]
In the periodic table, it is a p-block transactinide element.[7]
[*Transactinide_element*]
In chemistry, transactinide elements (also, transactinides, or super-heavy elements) are the chemical elements with atomic numbers from 104 to 120.[0]

ID: 201459
Claim: A dynamic web page does not involve computer programming.
Label: REFUTED
Evidence: [*Web_page*]
A static web page is delivered exactly as stored, as web content in the web server's file system, while a dynamic web page is generated by a web application that is driven by server-side software or client-side scripting.[14]
[*Dynamic_web_page*]
A dynamic web page is then reloaded by the user or by a computer program to change some variable content.[9]

Figure 1: Examples from the FEVER dev set, where true evidence sentences are present in the selected sentences, and veracity relation labels are correctly predicted by our proposed model. Wikipedia article titles are in [*italics*]. Superscripts indicate the positions of the sentences in each article.

[1]The code and model checkpoints are available at: https://github.com/nii-yamagishilab/mla.

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 2447–2460
August 1–6, 2021.

Recent state-of-the-art approaches have attempted to meet this challenge by applying graph-based neural networks (Kipf and Welling, 2017; Velickovic et al., 2018). For example, Zhou et al. (2019) regard an evidence sentence as a graph node, while Liu et al. (2020) use a more fine-grained node representation based on token-level attention. Zhong et al. (2020) use semantic role labeling (SRL) to build a graph structure, where a node can be a word or a phrase depending on the SRL's outputs.

In this paper, we argue that such sophisticated graph-based approaches may be unnecessary for the claim verification task. We propose a simple model that can be trained on a sequence structure. We also observe mismatches between training and testing. At test time, the model predicts the veracity of a claim based on retrieved documents and selected sentences, which contain prediction errors, while at training time, only ground-truth documents and true evidence sentences are available. We empirically show that our model, trained with a method that helps reduce training-test discrepancies, outperforms the graph-based approaches.

In addition, we observe that most of the previous work neglects sentence-selection labels when training veracity prediction models. Thus, we propose leveraging those labels to further improve veracity relation prediction through joint training. Unlike previous work that jointly trains two models (Yin and Roth, 2018; Li et al., 2020; Hidey et al., 2020; Nie et al., 2020), our approach is still a pipeline process where only a subset of potential candidate sentences produced by *any* sentence selector can be used for joint training. This approach makes it possible to explore different sentence-selection models trained with different methods.

Our contributions are as follows. We develop a method for mitigating training-test discrepancies by using a mixture of predicted and true examples for training. We propose a multi-level attention (MLA) model that enables token- and sentence-level self-attentions and that benefits from joint training. Experiments on the FEVER dataset show that MLA outperforms all the published models, despite its simplicity.

2 Background and related work

2.1 Problem formulation

The input of our task is a claim and a collection of Wikipedia articles $\mathcal{D}$. The goal is to extract a set of evidence sentences from $\mathcal{D}$ and assign a veracity relation label $y \in \mathcal{Y} = \{\mathrm{S}, \mathrm{R}, \mathrm{N}\}$ to a claim with respect to the evidence set, where S = SUPPORTED, R = REFUTED, and N = NOTENOUGHINFO. The definition of our labels is identical to that of the FEVER Challenge (Thorne et al., 2018).

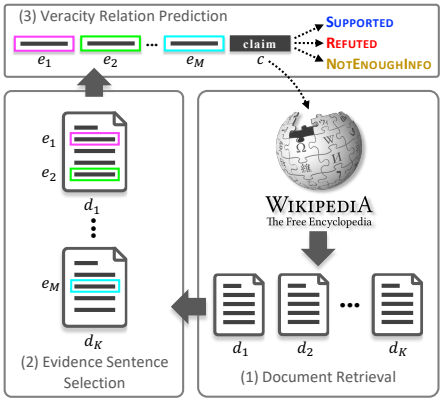

Figure 2: Process of evidence-based fact checking: retrieving documents relevant to a given claim from Wikipedia, selecting sentences likely to contain evidence, and predicting a veracity relation label based on selected sentences.

2.2 Overview of evidence-based fact checking

The process of evidence-based fact checking, shown in Figure 2, commonly involves the following three subtasks.

Document retrieval

Given a claim, the task is to retrieve the top K relevant documents from $\mathcal{D}$. Thorne et al. (2018) suggest using the document retriever from DrQA (Chen et al., 2017a), which ranks documents on the basis of the term frequency-inverse document frequency (TF-IDF) model with unigram-bigram hashing. Hanselowski et al. (2018) use a hybrid approach that combines search results from the MediaWiki API and the results of using exact matching on all Wikipedia article titles. In this paper, our main focus is to improve evidence sentence selection and veracity relation prediction, so we directly use the document retrieval results from Hanselowski et al. (2018). This allows us to fairly compare our model with a series of previous methods (Soleimani et al., 2019; Zhou et al., 2019; Liu et al., 2020; Ye et al., 2020) that also rely on Hanselowski et al. (2018)'s results.

Evidence sentence selection

The task is to select the top M sentences from the retrieved documents. Thorne et al. (2018) again

use the TF-IDF model to rank sentences similar to a given claim. Nie et al. (2019a); Hanselowski et al. (2018) use the enhanced sequential inference model (ESIM) (Chen et al., 2017b) to encode and align a claim-sentence pair. Liu et al. (2020); Hanselowski et al. (2018) use a pairwise hinge loss to rank sentences, while Soleimani et al. (2019) explore both pointwise and pairwise losses and suggest selecting difficult negative examples for training. The pairwise hinge loss aims to maximize the margin between the scores of positive and negative examples, while the pointwise loss is a vanilla cross-entropy loss. Our model uses a pointwise loss trained with examples sampled from both ground-truth and predicted documents.

Veracity relation prediction

Given a claim and a set of M selected sentences, the task is to predict their veracity relation label y. Previous work on the FEVER Challenge modified existing RTE/NLI models to deal with multiple sentences (Nie et al., 2019a; Yoneda et al., 2018; Hanselowski et al., 2018; Thorne et al., 2018), used heuristic rules to combine predictions from individual claim-sentence pairs (Malon, 2018), or concatenated all sentences (Stammbach and Neumann, 2019). A line of recent work has applied graph-based neural networks (Zhou et al., 2019; Liu et al., 2020; Zhong et al., 2020). Our model is simply trained on linear sequences by using self- and cross-attention to learn inter-sentence interactions.

2.3 Pre-trained language models

A key to the success of state-of-the-art approaches is the use of pre-trained language models (Devlin et al., 2019; Liu et al., 2019; Yang et al., 2019; Lan et al., 2020). Here, we use BERT (Devlin et al., 2019), a bidirectional transformer (Vaswani et al., 2017), to obtain the vector representation of a token sequence. Each BERT layer transforms an input token sequence (one or two sentences) by using self-attention. The first hidden state vector of the final layer represents a special classification token (CLS), which can be used in downstream tasks. We denote the above process by $\mathrm{BERT}_{\mathrm{CLS}}(\cdot) \in \mathbb{R}^{d_h}$, where d_h means the dimensionality of BERT hidden state vectors.

3 Proposed method

In this section, we describe our contributions, including (1) our method for training the sentence-selection model and (2) our veracity prediction model that can be extended with inter-sentence attentions and joint training.

3.1 Learning to select sentences from mixed ground-truth and retrieved documents

Our goal is to select a subset of evidence sentences from all candidate sentences in the retrieved documents. We consider this task to be a binary classification problem that takes as input a pair of a claim c and a candidate sentence e_j and maps it to the output $z \in \mathcal{Z} = \{-1, +1\}$, where $+1$ indicates an evidence sentence and -1 otherwise. We train our sentence-selection model by minimizing the standard cross-entropy loss for each example:

$$\mathcal{L}_{e_j}(\boldsymbol{\phi}) = -\sum_{z \in \mathcal{Z}} \mathbb{1}\{\hat{z} = z\} \log p_{\boldsymbol{\phi}}(\hat{z}|c, e_j), \quad (1)$$

where $\mathbb{1}\{\cdot\}$ is the indicator function, and $p_{\boldsymbol{\phi}}$ is the probability distribution of the two classes generated by our model. We compute $p_{\boldsymbol{\phi}}$ by applying a multi-layer perceptron (MLP) to the vector representation of e_j followed by a softmax function:

$$p_{\boldsymbol{\phi}}(\hat{z}|c, e_j) = \mathrm{softmax}(\mathrm{MLP}(\mathbf{e}_j)), \quad (2)$$
$$\mathbf{e}_j = \mathrm{BERT}_{\mathrm{CLS}}(c, e_j).$$

The MLP contains two affine transformations that map $\mathbf{e}_j$ to the output space. Feeding the pair of c and e_j to BERT allows us to obtain hidden state vectors that capture interactions between c and e_j at the token level. This is due to the self-attention mechanism inside the BERT layers. We expect the final hidden state vector of the CLS token (i.e., $\mathbf{e}_j$) to encode useful information from e_j with respect to c. The parameters $\boldsymbol{\phi}$ include those in MLP and BERT.

Training our model seems straightforward. However, two technical issues exist. First, each document typically contains one or two (or no) evidence sentences. Training with a few positive examples (i.e., evidence sentences) against all negative examples (i.e., non-evidence sentences) may be neither efficient nor effective. Soleimani et al. (2019) use hard negative mining (HNM) to repeatedly select a subset of difficult negative examples for training their sentence selector. Second, at test time, the model must examine all candidate sentences in the relevant documents returned by the document retriever. However, at training time, the model has no chance to learn the characteristics of non-evidence sentences in the irrelevant but highly ranked documents if only the ground-truth documents are used.

We propose to mitigate the aforementioned issues by using both the ground-truth and retrieved documents to create negative examples for a claim. First, we randomly choose r non-evidence sentences from each ground-truth document, where r is twice the number of true evidence sentences. Then, we sample two other non-evidence sentences from each retrieved document. For positive examples, we use the true evidence sentences in the ground-truth documents. Our scheme is more efficient than HNM of Soleimani et al. (2019). At test time, we select the top M sentences according to the probabilities assigned to the positive class.

3.2 Multi-level attention and joint training for veracity relation prediction

Training-test discrepancies also occur in veracity relation prediction. At test time, the model predicts the veracity of a claim on the basis of the predicted evidence sentences. At training time, only true evidence sentences are available for SUPPORTED and REFUTED, but not for NOTENOUGHINFO. In other words, we have no example sentences that more or less relate to a claim but may not be sufficient to support or refute the claim to train the model. Thorne et al. (2018) simulate training examples for NOTENOUGHINFO by sampling a sentence from the highest-ranked page returned by the document retriever.

We propose to reduce this discrepancy by using a mixture of true and predicted evidence sentences for training. First, we pair each claim with a list of the top M predicted sentences obtained through a sentence selector. At training time, we then prepend the true evidence sentences (if available) to the list and keep the number of all the sentences at most M.[2] At test time, we use the top M predicted sentences without requiring a predefined threshold to filter them. This is in contrast to previous work (Zhou et al., 2019; Nie et al., 2019b; Wadden et al., 2020) and helps reduce engineering effort. Our example sentences for NOTENOUGHINFO are from the sentence selector, not from the document retriever as in (Thorne et al., 2018). We expect our training examples to be similar to what our model may encounter at test time.

On the basis of the above scheme, each example is a pair of a claim c and a set of evidence sentences $\{e_j\}_{j=1}^{M}$. Our goal is to predict the veracity relation label $y \in \mathcal{Y} = \{\mathrm{S}, \mathrm{R}, \mathrm{N}\}$. We train our veracity prediction model by minimizing the class-weighted cross-entropy loss for each example:

$$\mathcal{L}_p(\boldsymbol{\theta}) = -\sum_{y \in \mathcal{Y}} \beta_y \mathbb{1}\{\hat{y} = y\} \log p_{\boldsymbol{\theta}}(\hat{y}|c, \{e_j\}_{j=1}^{M}), \tag{3}$$

where β_y is the class weight for dealing with the class imbalance problem (detailed in Section 4.2). Similar to Eq. (2), we compute the probability distribution $p_{\boldsymbol{\theta}}$ of veracity relation labels as:

$$p_{\boldsymbol{\theta}}(\hat{y}|c, \{e_j\}_{j=1}^{M}) = \mathrm{softmax}(\mathrm{MLP}(\mathbf{a})). \tag{4}$$

Here, $\mathbf{a}$ is the vector representation of aggregated evidence about a claim that is obtained through the multi-head attention (MHA) function:

$$\mathbf{a} = \mathrm{MHA}(\mathbf{Q} = \mathbf{c}, \mathbf{K} = \mathbf{E}, \mathbf{V} = \mathbf{E}), \tag{5}$$

where $\mathbf{c}$ is the claim vector, $\mathbf{E}$ is the set of evidence vectors $\{\mathbf{e}_j\}_{j=1}^{M}$, and $\mathbf{Q}$, $\mathbf{K}$, $\mathbf{V}$ denote the query, keys, and values, respectively. All the claim and evidence vectors are derived from BERT:

$$\mathbf{c} = \mathrm{BERT}_{\mathrm{CLS}}(c),$$
$$\mathbf{e}_j = \mathrm{BERT}_{\mathrm{CLS}}(c, e_j).$$

The parameters $\boldsymbol{\theta}$ are those in MLP, MHA, and BERT.

Now let us explain the MHA function, because we use and/or modify it in other components. The MHA function is based on the scaled dot-production attention (Vaswani et al., 2017):

$$\mathrm{attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{softmax}\Big(\frac{\mathbf{Q}\mathbf{K}^\top}{\gamma}\Big)\mathbf{V}, \tag{6}$$

where $\gamma = \sqrt{d_h/n}$ is the scaling factor. The above function is the weighted sum of the values (i.e., the evidence vectors), where the weight assigned to each value is the result of applying a softmax function to the scaled dot products between the query (i.e., the claim vector) and the keys (i.e., the evidence vectors).

The MHA function contains a number of parallel heads (i.e., attention layers). We expect each head to capture different aspects of the input. We achieve this by linearly projecting $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ to new representations and feeding them to the scaled dot-product attention. Specifically, the MHA function is given by:

$$\mathrm{MHA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = [\mathrm{head}_1, \ldots, \mathrm{head}_n]\mathbf{W}^O, \tag{7}$$

$$\mathrm{head}_i = \mathrm{attn}(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V), \tag{8}$$

[2] True evidence sentences may already exist in the list because the sentence selector can correctly identify them.

where n is the number of parallel heads, and $\mathbf{W}_i^Q, \mathbf{W}_i^K, \mathbf{W}_i^V \in \mathbb{R}^{d_h \times \frac{d_h}{n}}$; $\mathbf{W}^O \in \mathbb{R}^{d_h \times d_h}$ are the weight matrices of the linear projections.

Inter-sentence attentions

Although Eq. (5) helps aggregate the evidence from multiple selected sentences, our model still has no mechanism to learn interactions among these sentences. Unlike previous work that uses graph-based attention (Zhou et al., 2019; Liu et al., 2020; Zhong et al., 2020), our main tool is just the described MHA function.

Let $\mathbf{H}_j = [\mathbf{h}_{j,1}, ..., \mathbf{h}_{j,L}]$ be a sequence of the hidden state vectors of e_j generated by BERT, where L is the maximum sequence length. Let $\mathbf{H}$ be the concatenation of all the sequences $\{\mathbf{H}_j\}_{j=1}^M$. We obtain a new representation $\mathbf{G}$ of the concatenated sequence by applying a residual connection between $\mathbf{H}$ and token-level self-attention:

$$\mathbf{G} = \mathbf{H} + \mathrm{MHA}(\mathbf{H}), \tag{9}$$

where $\mathrm{MHA}(\cdot)$ is a simplified MHA function with one argument because $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ all come from the same $\mathbf{H}$.

In practice, we also add the static (sinusoid) positional encodings (PE) to the input of MHA.[3] We adopt this procedure from the original Transformer's sub-layer (Vaswani et al., 2017). The computation cost of Eq. (9) is not high. Concretely, let $L = 128$ and $M = 5$. The length of the concatenated sequence is thus 640 ($L \times M$), which is slightly longer than the maximum length of BERT's input sequence (i.e., 512 tokens).

Next, we perform sentence-level self-attention using a similar procedure. First, we split $\mathbf{G}$ back into individual sequences $\{\mathbf{G}_j\}_{j=1}^M$. Then, we pick the first hidden state vector from each $\mathbf{G}_j$, which corresponds to that of the CLS token. Let $\mathbf{F}$ be the concatenation of all the first hidden state vectors $\{\mathbf{g}_{j,1}\}_{j=1}^M$. We obtain the final representation $\mathbf{E}$ of the evidence sentences:

$$\mathbf{E} = \mathbf{F} + \mathrm{MHA}(\mathbf{F}). \tag{10}$$

We can use $\mathbf{E}$ as the keys and values in Eq. (5). Note that we do not share the parameters among the different MHA layers.

[3]During development, we tried the other basic components, i.e., layer normalization and position-wise feed-forward, but found it yielded no improvements in our task.

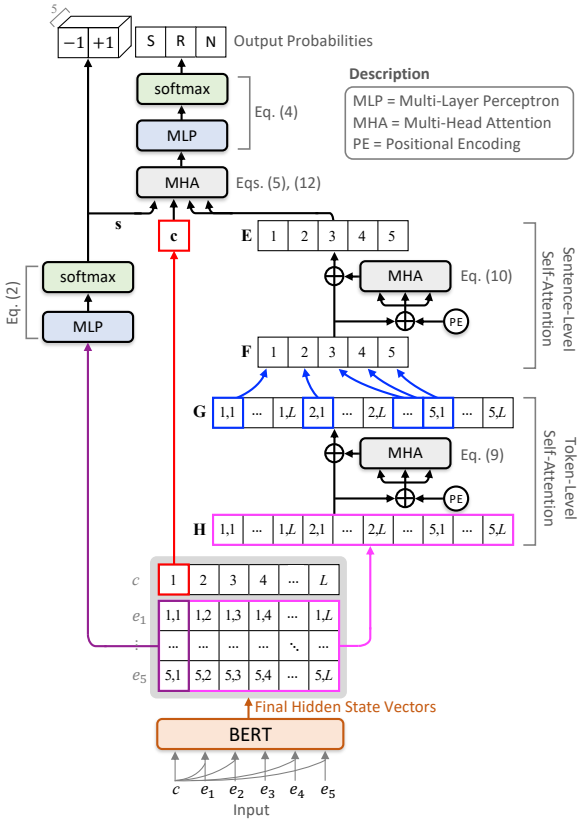

Figure 3: Architecture of our multi-level attention (MLA) model. The model takes as input a claim together with five evidence sentences. These sentences can be derived from any sentence selector. BERT encodes each sentence into a sequence of hidden state vectors, each of which is denoted by a squared box. The first hidden state vector (corresponding to the CLS token) is used for classification. MLA applies token- and sentence-level self-attentions and combines all the hidden state vectors as well as the sentence-selection scores at the final attention layer.

Joint training

Since the sentence-selection label assigned to each evidence sentence is available at training time, we can use it to guide our veracity prediction model. We apply the idea of multi-task learning (MTL) (Caruana, 1993; Ruder, 2017), in which we consider veracity relation prediction to be our main task and evidence sentence selection to be our auxiliary task. Our goal is to leverage training signals from our auxiliary task to improve the performance of our main task. Note that the sentence-selection component here is independent of the stand-alone model (i.e., our model in Section 3.1 or an alternative model in Section 4.3).

Let $\mathbf{s} = [s_1, \ldots, s_M]$ be the vector of sentence-selection scores, where s_m denotes the probability distribution of the positive class returned by our

sentence-selection component. We propose using $\mathbf{s}$ as a gate vector to determine how much of the values should be maintained before applying a residual connection followed by a linear projection. Thus, we modify Eq. (8) with:

$$\text{head}_i = \text{attn}(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \widetilde{\mathbf{V}}\mathbf{W}_i^V), \quad (11)$$

$$\widetilde{\mathbf{V}} = \mathbf{V} + \mathbf{s} \odot \mathbf{V}, \quad (12)$$

where $\odot$ means the element-wise multiplication.

Our modification is close to Shaw et al. (2018)'s method in which extra vectors are added to the keys and the values after applying the linear projections. During development, we found that their method does not work well in our task. We compare different strategies in Section 4.4, including applying the gate vector to the keys or both the keys and the values.

Finally, we combine Eqs. (1) and (3) to get our composite loss function:

$$\min_{\boldsymbol{\theta},\boldsymbol{\phi}} \mathcal{L} = \mathcal{L}_p + \lambda \sum_{j=1}^{M} \mathcal{L}_{e_j}, \quad (13)$$

where λ is the weighting factor of the sentence-selection component.

To summarize, our model, shown in Figure 3, contains token-level attention over a claim-evidence pair through BERT, token- and sentence-level self-attentions across an evidence set, and claim-evidence cross-attention incorporating the sentence-selection scores through joint training. Hence, we call it the multi-level attention (MLA) model.

4 Experiments

4.1 Dataset and evaluation metrics

Table 1 shows the statistics of the FEVER dataset. We used the corpus of the June 2017 Wikipedia dump, which contains 5,416,537 articles preprocessed by Thorne et al. (2018). We used the document retrieval results given by Hanselowski et al. (2018), containing the predicted Wikipedia article titles (i.e., document IDs) for all the claims in the training/dev/test sets. Following (Stammbach and Neumann, 2019; Soleimani et al., 2019; Liu et al., 2020), we prefixed the Wikipedia article titles to the candidate sentences to alleviate the co-reference resolution problem.

We evaluated performance by using the label accuracy (LA) and FEVER score. LA measures the 3-way classification accuracy of veracity relation prediction. The FEVER score reflects the performance of both evidence sentence selection and veracity relation prediction, where a complete set of true evidence sentences is present in the selected sentences, and the claim is correctly labeled. We used the official FEVER scorer during development.[4] We limited the number of the selected sentences to five (M = 5) according to the FEVER scorer. The performance on the blind test set was evaluated through the FEVER Challenge site.

Split	SUPPORTED	REFUTED	NOTENOUGHINFO
Training	80,035	29,775	35,659
Dev	6,666	6,666	6,666
Test	6,666	6,666	6,666

Table 1: Statistics of the FEVER dataset. Veracity relation labels and evidence sentences of the test set are not publicly available.

4.2 Training details

We implemented our model on top of HuggingFace's Transformer (Wolf et al., 2020). The dimension of hidden state vectors d_h and the number of heads n varied according to those of the pre-trained models. We used BERT-base (d_h = 786; n = 12) for our stand-alone sentence-selection model and tried various BERT-style models for MLA.

We trained all models using Adafactor (Shazeer and Stern, 2018) with a batch size of 256, a linear learning rate decay, a warmup ratio of 0.06, and a gradient clipping of 1.0. Following the default configuration of HuggingFace's Transformer, we initialized all parameters by sampling from $\mathcal{N}(0, 0.02)$ and setting the biases to 0, except for the pre-trained models. We set λ in Eq. (13) to 1. We trained each model for 2 epochs with a learning rate of 5e-5, unless otherwise specified.

For regularization, we applied dropout (Hinton et al., 2012) with a probability of 0.1 to the MHA layers, MLP layers, and gated values in Eq. (12). We computed the class weight β_y in Eq. (3) by:

$$\beta_y = \frac{\hat{\beta}_y}{\sum_{y \in \mathcal{Y}} \hat{\beta}_y}, \quad \hat{\beta}_y = \frac{N}{|\mathcal{Y}| \times N_y},$$

where $\hat{\beta}_y$ is the balanced heuristic used in scikit-learn (Pedregosa et al., 2011) and β_y is normalized to sum to 1. In our case, N = 145,469 is the total

[4] https://github.com/sheffieldnlp/fever-scorer

number of training examples, $|\mathcal{Y}| = 3$ is the number of classes, and N_y is the number of training examples in y (i.e., the first row in Table 1). We interpreted $\hat{\beta}_y$ as the ratio of the balanced class distribution ($N/|\mathcal{Y}|$) to the observed one (N_y). Here, we wanted to penalize the less-observed classes, like REFUTED and NOTENOUGHINFO, more.

4.3 Results

Baselines

The use of different pre-trained and pipeline models in the previous work makes a fair comparison difficult. For this reason, we chose baseline models that use BERT-base for pre-training and Hanselowski et al. (2018)'s document retrieval results. We designed two sets of experiments.

First, we required that all the models use the same sentence-selection model, which is Hanselowski et al. (2018)'s ESIM.[5] For the veracity relation prediction, Hanselowski et al. (2018) incorporate ESIM with attention and pooling operations to get a representation of a claim and top five selected sentences. Soleimani et al. (2019) make five independent predictions for each claim-evidence pair and use a heuristic (Malon, 2018) to get a final prediction. GEAR (Zhou et al., 2019) is a graph-based model for evidence aggregating and reasoning. KGAT (Liu et al., 2020) is a kernel graph attention model. Second, we allowed different sentence-selection models. Soleimani et al. (2019) use HNM to select negative examples with the highest loss values, while our negative examples are sampled once from both the ground-truth and retrieved documents, as described in Section 3.1.

Table 2 shows the results of the two settings on the dev set. MLA outperforms the other baselines in both settings. Table 3 shows the sentence-selection results returned by the FEVER scorer. The precision, recall@5, and F1 are consistent across the three models. Hanselowski et al. (2018) use ESIM with a pairwise hinge loss, while Soleimani et al. (2019) use a pointwise loss with HNM. Our model is also a pointwise approach but simpler to train. Without sampling non-evidence sentences from the retrieved documents, all the scores drop by around 2%, indicating that our technique is useful. In the following sections, we will refer to our BERT-base sentence-selection results with MLA.

[5]We used the sentence-selection results reproduced by Zhou et al. (2019).

Model	LA	FEVER
Sentence selection with ESIM		
Hanselowski et al. (2018)	68.49	64.74
Soleimani et al. (2019)	71.70	69.79
GEAR† (Zhou et al., 2019)	74.84	70.69
KGAT† (Liu et al., 2020)	75.51	71.61
MLA (Ours)	**76.30**	**72.83**
Sentence selection with BERT-base		
Soleimani et al. (2019)‡	73.54	71.33
MLA (Ours)	**76.92**	**73.78**

Table 2: LA and FEVER score results on the dev set. All the models use the document retrieval results from Hanselowski et al. (2018). Results marked with † indicate using ESIM with a threshold filter, and ‡ indicates using BERT-base with HNM.

Model	Prec	Rec@5	F1
ESIM (Hanselowski et al., 2018)	24.08	86.72	37.69
BERT-base‡ (Soleimani et al., 2019)	25.13	88.29	39.13
BERT-base (Ours)	**25.63**	**88.64**	**39.76**
w/o sampling from retrieved docs.	23.59	87.18	37.13

Table 3: Sentence selection results on the dev set. Result marked with ‡ indicates using HNM.

Effect of pre-trained models

The next set of experiments examined the benefits of using different pre-trained models. ALBERT (Lan et al., 2020) is a lite BERT training approach that uses cross-layer parameter sharing and replaces next sentence prediction with sentence ordering. RoBERTa (Liu et al., 2019) is a robustly optimized BERT approach that introduces better training schemes, including dynamic masking, larger batch size, and other techniques. We chose these two BERT-style models because they can be easily plugged into our implementation without much modification.

Table 4 shows the results of the different pre-trained models on the dev set. For all the large pre-trained models, we decreased the learning rate to 2e-5 and trained them for 3 epochs. Additional results including training times can be found in Appendix A. As shown in the table, BERT and ALBERT perform similarly, while ALBERT has fewer parameters. RoBERTa offers consistent improvements over the other two models and achieves the best performance with its large model. Therefore,

Pre-trained model	# Params	LA	FEVER
BERT-base	117M	76.92	73.78
BERT-large	349M	77.27	74.10
ALBERT-base	20M	76.58	73.83
ALBERT-large	33M	76.94	74.24
RoBERTa-base	132M	77.54	74.41
RoBERTa-large	370M	**79.31**	**75.96**

Table 4: LA and FEVER score results of MLA on the dev set using different pre-trained models. The second column shows the number of parameters, including those from the pre-trained model and our task-specific layers (i.e., MHA and MLP layers).

Model	LA	FEVER
Hanselowski et al. (2018)	65.46	61.58
Yoneda et al. (2018)	67.62	62.52
Nie et al. (2019a)	68.21	64.21
GEAR† (Zhou et al., 2019)	71.60	67.10
SR-MRS† (Nie et al., 2019b)	72.56	67.26
BERT‡ (Soleimani et al., 2019)	71.86	69.66
KGAT♢ (Liu et al., 2020)	74.07	70.38
DREAM♣ (Zhong et al., 2020)	76.85	70.60
HESM♠ (Subramanian and Lee, 2020)	74.64	71.48
CorefRoBERTa♢ (Ye et al., 2020)	75.96	72.30
MLA♢(Ours)	**77.05**	**73.72**

Table 5: LA and FEVER score results on the blind test set. Results marked with † indicate using BERT-base, ‡ BERT-large, ♢ RoBERTa-large, ♣ XLNet, and ♠ ALBERT-large.

we applied MLA with RoBERTa-large to the blind test set.

Comparison with state-of-the-art methods

Table 5 shows the results on the blind test set.[6] The results are divided into two groups. The first group represents the top scores of the FEVER shared task, including those of Hanselowski et al. (2018); Yoneda et al. (2018); Nie et al. (2019a). The second group contains recently published results after the shared task. GEAR (Zhou et al., 2019), KGAT (Liu et al., 2020), and DREAM (Zhong et al., 2020) are graph-based models. SR-MRS (Nie et al., 2019b) uses a semantic retrieval module for selecting evidence sentences. HESM (Subramanian and Lee, 2020) uses a multi-hop evidence retriever and a hierarchical evidence aggregation model. CorefRoBERTa (Ye et al., 2020) trains KGAT by using a pre-trained model that combines a coreference prediction loss. Their pre-trained model is initialized with RoBERTa-large's parameters and further trained on Wikipedia. MLA outperforms all the published models and yields 1.09% and 1.42% improvements in LA and FEVER score, respectively, over CorefRoBERTa. Additional sentence-selection results can be found in Appendix B.

[6]The results can also be found on the FEVER leaderboard: https://competitions.codalab.org/competitions/18814#results

Model	LA	FEVER
MLA (full)	**76.92**	**73.78**
w/o token-level self-attention	76.30	73.20
w/o sentence-level self-attention	76.50	73.41
w/o class weighting	76.44	73.14
w/o joint training	76.65	73.22

Table 6: Ablation studies of the proposed components on the dev set with BERT-base.

Model	LA	FEVER
MLA (w/ value)	**76.92**	**73.78**
w/ key	76.74	73.65
w/ key & value	76.82	73.60
w/ dot-product	76.70	73.51
w/o using $\mathbf{s}$	76.64	73.47

Table 7: Ablation studies of different strategies for using the sentence-selection scores $\mathbf{s}$ on the dev set with BERT-base.

4.4 Ablation study

We conducted two sets of ablation studies on the dev set using MLA with BERT-base. First, we examined the effect of our proposed components. Table 6 shows that all the components contribute to the final results. Without class weighting, Eq. (3) falls back to the standard cross-entropy loss. Without joint training, MLA is a stand-alone veracity prediction model. These results suggest that token-level self-attention and class weighting are the two most important components of our model.

Second, we explored a number of strategies for exploiting the sentence-selection scores $\mathbf{s}$. MLA basically uses $\mathbf{s}$ as a gate vector and only applies it to the values, as described in Eq. (12). We can apply the same calculation to the keys or both the keys and the values. In addition, we can use $\mathbf{s}$ as a bias vector and add it to the scaled dot-product term, as done by Yang et al. (2018). Table 7 shows the results of the aforementioned strategies. These results indicate that applying $\mathbf{s}$ to the values produces the best results.

ID:	35237
Claim:	Philomena is a film nominated for seven awards.
Evidence:	[*Philomena_(film)*] It was also nominated for four BAFTA Awards and three Golden Globe Awards.[9]
Annotated label:	SUPPORTED
Predicted label:	REFUTED

(a)

ID:	33547
Claim:	Mick Thomson was born in Ohio.
Evidence:	[*Mick_Thomson*] Born in Des Moines, Iowa, he is best known as one of two guitarists in Slipknot, in which he is designated #7.[1]
Annotated label:	SUPPORTED
Predicted label:	REFUTED

(b)

ID:	73443
Claim:	Heavy Metal music was developed in the United Kingdom.
Evidence:	[*Heavy_metal_music*] Heavy metal (or simply metal) is a genre of rock music that developed in the late 1960s and early 1970s, largely in the United Kingdom and the United States.[0]
Annotated label:	REFUTED
Predicted label:	SUPPORTED

(c)

ID:	212780
Claim:	Harvard University is the first University in the U.S.
Evidence:	[*Harvard_University*] Established originally by the Massachusetts legislature and soon thereafter named for John Harvard (its first benefactor), Harvard is the United States' oldest institution of higher learning ...[3]
Annotated label:	SUPPORTED
Predicted label:	NOTENOUGHINFO

(d)

Table 8: Examples where the models disagree with the annotated labels.

4.5 Error analysis

To better understand the limitations of our method, we manually inspected 100 prediction errors on the dev set, where the true evidence sentences are present in the predicted sentences but MLA failed to predict the veracity relation labels. Here, we required that both BERT-base and RoBERTa-large MLA models produce the same errors.

Table 8(a) shows a prediction error requiring complex reasoning that our models are unable to deal with. The claim "*Philomena is a film nominated for **seven** awards.*" is supported by the evidence "*It was also nominated for **four** BAFTA Awards and **three** Golden Globe Awards.*". In this case, the models must understand that four plus three equals seven.

Table 8(b) shows a possible annotation error. The claim "*Mick Thomson was born in **Ohio**.*" is annotated as SUPPORTED, while the evidence "*Born in Des Moines, **Iowa**, he is best known as ...*" refutes the claim. Our models also predict REFUTED.

Table 8(c) shows the half-true claim "*Heavy Metal music was developed in **the United Kingdom**.*", which is annotated as REFUTED. However, the evidence "*Heavy metal (or simply metal) is ... developed ... in **the United Kingdom and the United States**.*" would indicate that the claim is partly true. The half-true label is defined in some previous smaller datasets (Vlachos and Riedel, 2014; Wang, 2017), but not in the FEVER dataset.

Table 8(d) shows the questionable claim "*Harvard University is **the first University** in the U.S.*", which is annotated as SUPPORTED by the evidence "*... Harvard is the United States' **oldest institution** of higher learning ...*". However, this evidence does not directly support the claim.[7] Our models predict NOTENOUGHINFO. Our analysis results suggest that probing disagreements between an ensemble of models and annotators may help improve annotation consistency. Additional results on error analysis are given in Appendix C.

5 Conclusion

We have presented a multi-level attention model that operates on linear sequences. We find that, when trained properly, the model outperforms its graph-based counterparts. Our results suggest that a sequence model is sufficient and can serve as a strong baseline. Using better upstream components (i.e., a better document retriever or sentence selector) or larger pre-trained models would further improve the performance of our model. Training models that are robust to adversarial examples while maintaining high performance for normal ones is an important direction for our future work.

Acknowledgments

We thank Erica Cooper (NII) for providing valuable feedback on an earlier draft of this paper. This work is supported by JST CREST Grants (JPMJCR18A6 and JPMJCR20D3) and MEXT KAKENHI Grants (21H04906), Japan.

[7]The topic is still under debate: https://en.wikipedia.org/wiki/First_university_in_the_United_States.

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of EMNLP*, pages 632–642.

Richard Caruana. 1993. Multitask learning: A knowledge-based source of inductive bias. In *Proceedings of ICML*, pages 41–48.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017a. Reading Wikipedia to answer open-domain questions. In *Proceedings of ACL*, pages 1870–1879.

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

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The PASCAL recognising textual entailment challenge. In *Machine Learning Challenges*, volume 3944 of *LNAI*, pages 177–190. Springer.

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

Andreas Hanselowski, Hao Zhang, Zile Li, Daniil Sorokin, Benjamin Schiller, Claudia Schulz, and Iryna Gurevych. 2018. UKP-athene: Multi-sentence textual entailment for claim verification. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 103–108.

Christopher Hidey, Tuhin Chakrabarty, Tariq Alhindi, Siddharth Varia, Kriste Krstovski, Mona Diab, and Smaranda Muresan. 2020. DeSePtion: Dual sequence prediction and adversarial examples for improved fact-checking. In *Proceedings of ACL*, pages 8593–8606.

Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580.

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

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *Proceedings of ICLR*.

Xiangci Li, Gully Burns, and Nanyun Peng. 2020. A paragraph-level multi-task learning model for scientific fact-verification. In *Proceedings of The AAAI-21 Workshop on Scientific Document Understanding*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

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

Christopher Malon. 2018. Team papelo: Transformer networks at FEVER. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 109–113.

Yixin Nie, Lisa Bauer, and Mohit Bansal. 2020. Simple compounded-label training for fact extraction and verification. In *Proceedings of the Third Workshop on Fact Extraction and VERification (FEVER)*, pages 1–7.

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

Yixin Nie, Songhe Wang, and Mohit Bansal. 2019b. Revealing the importance of semantic retrieval for machine reading at scale. In *Proceedings of EMNLP-IJCNLP*, pages 2553–2566.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *CoRR*, abs/1706.05098.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of NAACL*, pages 464–468.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *Proceedings of ICML*, pages 4596–4604.

Amir Soleimani, Christof Monz, and Marcel Worring. 2019. BERT for evidence retrieval and claim verification. In *Proceedings of European Conference on Information Retrieval*, pages 359–366.

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

Shyam Subramanian and Kyumin Lee. 2020. Hierarchical Evidence Set Modeling for automated fact extraction and verification. In *Proceedings of EMNLP*, pages 7798–7809.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of NIPS*, pages 5998–6008.

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

Andreas Vlachos and Sebastian Riedel. 2014. Fact checking: Task definition and dataset construction. In *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*, pages 18–22.

Soroush Vosoughi, Deb Roy, and Sinan Aral. 2018. The spread of true and false news online. *Science*, 359:1146–1151.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *Proceedings of EMNLP*, pages 7534–7550.

William Yang Wang. 2017. "liar, liar pants on fire": A new benchmark dataset for fake news detection. In *Proceedings of ACL*, pages 422–426.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of EMNLP: System Demonstrations*, pages 38–45.

Baosong Yang, Zhaopeng Tu, Derek F. Wong, Fandong Meng, Lidia S. Chao, and Tong Zhang. 2018. Modeling localness for self-attention networks. In *Proceedings of EMNLP*, pages 4449–4458.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Proceedings of NeurIPS*, pages 5753–5763.

Deming Ye, Yankai Lin, Jiaju Du, Zhenghao Liu, Peng Li, Maosong Sun, and Zhiyuan Liu. 2020. Coreferential Reasoning Learning for Language Representation. In *Proceedings of EMNLP*, pages 7170–7186.

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

Takuma Yoneda, Jeff Mitchell, Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. UCL machine reading group: Four factor framework for fact finding (HexaF). In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 97–102.

Wanjun Zhong, Jingjing Xu, Duyu Tang, Zenan Xu, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. 2020. Reasoning over semantic-level graph for fact checking. In *Proceedings of ACL*, pages 6170–6180.

Jie Zhou, Xu Han, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2019. GEAR: Graph-based evidence aggregating and reasoning for fact verification. In *Proceedings of ACL*, pages 892–901.

A Additional results on different pre-trained models

Table 9 shows the results of different pre-trained models in detail. All the pre-trained models used in our experiments also come from HuggingFace.[8] We conducted each experiment on a single NVIDIA Tesla A100 GPU with 40 GB RAM. We used a batch size of 256 with gradient accumulation to control memory.

B Additional sentence-selection results

Table 10 shows the results of various sentence-selection models on the test set. Not all published models report precision and recall. Our precision, recall@5, and F1 scores are slightly better than those of Liu et al. (2020). Our sentence-selection model took 1 hour and 10 minutes to train. We find that getting high recall in evidence sentence selection is necessary to achieve good results in veracity relation prediction.

C Additional error analysis

Here, we provide additional examples of errors, including complex reasoning errors (Table 11), possible annotation errors (Table 12), half-true claims (Table 13), and questionable claims (Table 14).

[8] https://huggingface.co/transformers/pretrained_models.html.

Pre-trained model	# Params	Learning rate	Epochs	Time	LA	FEVER
BERT-base	117M	5e-5	2	46m	76.92	73.78
BERT-large	349M	2e-5	3	2h 50m	77.27	74.10
ALBERT-base	20M	5e-5	2	57m	76.58	73.83
ALBERT-large	33M	2e-5	3	3h 35m	76.94	74.24
RoBERTa-base	132M	5e-5	2	45m	77.54	74.41
RoBERTa-large	370M	2e-5	3	2h 49m	**79.31**	**75.96**

Table 9: Additional results of MLA on the dev set using different pre-trained models.

Model	Loss	Pre-trained model	Prec	Rec@5	F1
Hanselowski et al. (2018)	Pairwise	–	–	–	36.97
Yoneda et al. (2018)	Pointwise	–	–	–	34.97
Nie et al. (2019a)	Pointwise	–	–	–	52.96
Zhou et al. (2019)	Pairwise & filtering	–	–	–	36.87
Nie et al. (2019b)	Pointwise	BERT-base	–	–	74.62
Soleimani et al. (2019)	Pointwise & HNM	BERT-base	–	–	38.61
Liu et al. (2020)	Pairwise	BERT-base	25.21	87.47	39.14
Zhong et al. (2020)	Pointwise	RoBERTa	25.63	85.57	39.45
Subramanian and Lee (2020)	Pointwise & multi-hop	ALBERT-base	–	–	52.78
Ye et al. (2020)	(adopting Liu et al. (2020)'s results)	–	–	–	39.14
This work	Pointwise	BERT-base	25.33	87.58	39.29

Table 10: Sentence-selection results on the blind test set. The F1 results can be found on the FEVER leaderboard: https://competitions.codalab.org/competitions/18814#results.

ID: 112396
Claim: Aristotle spent the majority of his life in Athens.
Evidence: [*Aristotle*] At seventeen or eighteen years of age, he joined Plato's Academy in Athens and remained there until the age of thirty-seven (c. 347 BC).[2]
Annotated label: SUPPORTED
Predicted label: REFUTED

ID: 3111
Claim: Luis Fonsi was born in the eighties.
Evidence: [*Luis_Fonsi*] Luis Alfonso Rodríguez López-Cepero, more commonly known by his stage name Luis Fonsi, (born April 15, 1978) is a Puerto Rican singer, songwriter and actor.[0]
Annotated label: REFUTED
Predicted label: SUPPORTED

ID: 64685
Claim: The Bassoon King is the full title a book.
Evidence: [*The_Bassoon_King*] The Bassoon King: My Life in Art, Faith, and Idiocy is a non-fiction book authored by American actor Rainn Wilson.[0]
Annotated label: REFUTED
Predicted label: SUPPORTED

ID: 102001
Claim: Jens Stoltenberg was Prime Minister of Norway once.
Evidence: [*Jens_Stoltenberg*] Stoltenberg served as Prime Minister of Norway from 2000 to 2001 and from 2005 to 2013.[4]
Annotated label: REFUTED
Predicted label: SUPPORTED

Table 11: Examples of prediction errors requiring complex reasoning.

ID: 117520
Claim: The host of The Joy of Painting was Bob Ross.
Evidence: [*Bob_Ross*] He was the creator and host of The Joy of Painting, an instructional television program that aired from 1983 to 1994 ...[1]
Annotated label: REFUTED
Predicted label: SUPPORTED

ID: 114640
Claim: IMDb is not user-edited.
Evidence: [*IMDb*] The site enables registered users to submit new material and edits to existing entries.[10]
Annotated label: SUPPORTED
Predicted label: REFUTED

ID: 137678
Claim: Food Network is available to approximately 96,931,000 pay television citizens.
Evidence: [*Food_Network*] As of February 2015, Food Network is available to approximately 96,931,000 pay television households ...[8]
Annotated label: REFUTED
Predicted label: SUPPORTED

ID: 34195
Claim: Annie Lennox was named "The Greatest White Soul Singer Alive" by VH1.
Evidence: [*Annie_Lennox*] Lennox has been named "The Greatest White Soul Singer Alive" by VH1 ...[19]
Annotated label: REFUTED
Predicted label: SUPPORTED

Table 12: Example of possible annotation errors.

ID: 174029
Claim: The Endless River came out in 1995 and is Pink Floyd's fifteenth studio album.
Evidence: [*The_Endless_River*] The Endless River is the fifteenth and final studio album by the English rock band Pink Floyd.[0]
Annotated label: REFUTED
Predicted label: SUPPORTED

ID: 161094
Claim: French Indochina was a grouping of territories.
Evidence: [*French_Indochina*] French Indochina (previously spelled as French Indo-China) ... was a grouping of French colonial territories in Southeast Asia.[0]
Annotated label: REFUTED
Predicted label: SUPPORTED

ID: 48148
Claim: On Monday August 19, 1945, Ian Gillan was born.
Evidence: [*Ian_Gillan*] Ian Gillan (born 19 August 1945) is an English singer and songwriter.[0]
Annotated label: SUPPORTED
Predicted label: NOTENOUGHINFO
Note: August 19, 1945 is Sunday, not Monday.

ID: 85350
Claim: Andrew Kevin Walker was born on Monday August 14, 1964.
Evidence: [*Andrew_Kevin_Walker*] Andrew Kevin Walker (born August 14, 1964) is an American BAFTA-nominated screenwriter .[0]
Annotated label: SUPPORTED
Predicted label: NOTENOUGHINFO
Note: August 19, 1945 is Friday, not Monday.

Table 13: Examples of half-true claims.

ID: 92900
Claim: The Indian Institute of Management Bangalore offers a business executive training program.
Evidence: [*Indian_Institute_of_Management_Bangalore*] It offers Post Graduate, Doctoral and executive training programmes.[5]
Annotated label: SUPPORTED
Predicted label: NOTENOUGHINFO
Note: The evidence does not specify that the institute offers a *business* executive training program.

ID: 46271
Claim: Prescott, Arizona is in northern Yavapai County.
Evidence: [*Prescott,_Arizona*] Prescott ... is a city in Yavapai County, Arizona, United States.[0]
Annotated label: SUPPORTED
Predicted label: NOTENOUGHINFO
Note: The evidence does not specify that Prescott is in the *northern* part of Yavapai County.

ID: 227779
Claim: Lyon is a city in Southwest France.
Evidence: [*Lyon*] Lyon had a population of 506,615 in 2014 and is France's third-largest city after Paris and Marseille.[4]
Annotated label: SUPPORTED
Predicted label: REFUTED
Note: The evidence does not directly support the claim.

Table 14: Examples of questionable claims.