Modeling Entity Knowledge for Fact Verification

Yang Liu, Chenguang Zhu, Michael Zeng
Microsoft Cognitive Services Research
1 Microsoft Way, Redmond, WA, USA
{yaliu10,chezhu,nzeng}@microsoft.com

Abstract

Fact verification is a challenging task of identifying the truthfulness of given claims based on the retrieval of relevant evidence texts. Many claims require understanding and reasoning over external entity information for precise verification. In this paper, we propose a novel fact verification model using entity knowledge to enhance its performance. We retrieve descriptive text from Wikipedia for each entity, and then encode these descriptions by a smaller lightweight network to be fed into the main verification model. Furthermore, we boost model performance by adopting and predicting the relatedness between the claim and each evidence as additional signals. We demonstrate experimentally on a large-scale benchmark dataset FEVER that our framework achieves competitive results with a FEVER score of 72.89% on the test set.

1 Introduction

The rapid development of online applications provides open and efficient platforms for spreading information. However, false information, including fake news and online rumors, have also been growing and spreading widely over the past several years. Vosoughi et al. (2018) shows that false news travels even faster, deeper and broader than the truth. To prevent harm from this false information, automatically verifying the truthfulness of textual contents is becoming an urgent need for our society. In this work, we study fact verification with the goal of automatically assessing the veracity of a textual claim given supporting evidence.

Most existing methods consider fact verification as a natural language inference task (Angeli and Manning, 2014). Usually, these systems concatenate claim and its supporting evidence sentences, and then feed them into a classification model (Nie et al., 2019). Alternatively, previous studies construct graph structures based on claim and evidence, and reason over this graph with graph neural networks (Zhou et al., 2019; Liu et al., 2020) or Transformer models (Zhong et al., 2020), which are used in top systems in the FEVER challenge (Thorne et al., 2018). While these studies focus on reasoning based on claim and evidence text, we believe entity knowledge is also important for precise fact verification. For example, given the first claim from FEVER dataset in Table 1, making the correct verification requires a model to understand what is "*Wii U*" and "*OS X*" and know the fact that they are not Microsoft and Sony platforms. Similarly, for the second claim, the knowledge that "*New York City*" is in United States can also be potentially useful for verifying the claim. This information is not included in the gold evidence provided by the dataset.

In this work, we present a fact verification model that can effectively incorporate external entity information. Given a claim and its evidence sentences, we first recognize named entities from them, linking them with Wikipedia articles, and then retrieve the lead sections of these articles as the entity descriptions. To make the most of this entity knowledge while not introducing noisy information, we propose a lightweight entity knowledge encoder module for representing external entity knowledge. Our large fact verification network then accesses this knowledge by a unidirectional attention mechanism at each encoding layer. Meanwhile, since the input evidence sentences are obtained by an upstream retrieval module, some evidence may be irrelevant to the claim. Thus, we predict and adopt this relatedness between each evidence and the claim as an auxiliary signal to train our model.

We experiment with our approach on FEVER (Thorne et al., 2018), one influential benchmark dataset for fact verification. FEVER contains over 185k labeled claims and each verifiable claim is paired with several natural language sentences from Wikipedia as their

Claim #1:
Assassin's Creed has only ever been released on a **Microsoft** and **Sony** platform.
Gold Evidence:
The main video game series consists of nine games , developed by **Ubisoft**, released on **PlayStation 3**, **PlayStation 4**, **Xbox 360**, **Xbox One**, **Wii U**, **Microsoft Windows**, and **OS X** platforms .
Entity Knowledge:
Wii U: The Wii U is a home video game console developed by Nintendo as the successor to the Wii.
OS X: macOS (previously Mac OS X and later OS X) is a series of proprietary graphical operating systems developed and marketed by Apple Inc.
Verdict: REFUTED
Claim #2:
Beastie Boys was formed in **Australia**.
Gold Evidence:
The Beastie Boys were an American hip hop group from **New York City**, formed in 1981.
Entity Knowledge:
New York City: New York City (NYC), often called simply New York, is the most populous city in the United States.
Verdict: REFUTED

Table 1: Two motivating examples for fact checking and the FEVER task. Identifying the truthfulness of claims requires understanding and reasoning of entity knowledge within the claim and the evidence sentences. The bold phrases are named entities. Underlined entities are linked to their Wikipedia descriptions, which can potentially provide useful knowledge for verifying the claim.

supporting evidence. Our system achieves the state-of-the-art result on label accuracy and competitive result on FEVER score. Ablation study shows that the integration of entity knowledge and auxiliary relatedness signal can effectively improve performance. We then provide a detailed error analysis for our system. In summary, we list our contributions as follows.

- We propose to enhance fact verification models with external entity knowledge.
- We design an entity knowledge encoder module and employ unidirectional attention to effectively incorporate entity descriptions.
- Empirical results show that our approach achieves competitive performance on the FEVER dataset, and ablation study shows that incorporating entity knowledge is useful for fact verification.

2 Related Work

2.1 Fact Verification

Fact checking is a challenging task aiming to automatically verify the truthfulness of claims. A claim can be a plain text or a triple of (subject, predicate, object) (Nakashole and Mitchell, 2014), and different fact checking datasets usually provide different evidence sources. Vlachos and Riedel (2014) propose a fact verification dataset by collecting 106 labeled political claims and providing the journalists' analysis material as the evidence. Ferreira and Vlachos (2016) construct the Emergent dataset containing 300 labeled rumors and 2,595 associated news articles, collected and labelled by journalists. LIAR (Wang, 2017) is a dataset for fake news detection. It contains 2.8K labeled short statements from the web, with detailed analysis and source documents as evidence. Chen et al. (2020c) build TABFACT, a dataset collecting Wikipedia tables as the evidence for human-labeled statements.

Recently, the FEVER shared task 1.0 (Thorne et al., 2018) attracts attention from the research community. It is a challenge that requires participants to develop automatic fact verification systems to check the truthfulness of human-generated claims by extracted evidence from Wikipedia. Many systems were proposed for this challenging task. Nie et al. (2019) design a Neural Semantic Matching Network that takes the concatenation of all evidence sentences as input. They also propose a two-hop evidence enhancement process where they apply sentence selection twice to retrieve more related evidence sentences. Stammbach and Neumann (2019) propose a two-staged selection process with two different retrieval models for selecting evidence sentences. Yoneda et al. (2018) infer the veracity of each claim-evidence pair and make final prediction by aggregating multiple predicted labels. Hanselowski et al. (2018) encode each claim-evidence pair separately, and use a pooling function to aggregate features for prediction. Zhou et al. (2019) formulates claim verification as a graph reasoning task and propose a new model with graph neural networks. Liu et al. (2020) regards sentences as the nodes of a graph and uses Kernel Graph Attention Network (KGAT) to aggregate information. Zhong et al. (2020) further constructs a semantic-level graph for input claim and evidence and perform reasoning over this graph with pretrained XLNet model (Yang et al., 2019).

Similar to our work, some previous systems also

focus on using entity information for fact verification. Taniguchi et al. (2018) first extract entities from the claim and propose to use a simple entity-linking system based on text match to retrieve evidence documents. Nooralahzadeh and Øvrelid (2018) select evidence documents by finding article titles which contain the entities and noun phrases of the claim.

2.2 Modeling External Knowledge in NLP

The usage of external knowledge, like WordNet, Wikipedia and knowledge graph, has benefited many natural language processing tasks including natural language inference and fact verification. Jijkoun et al. (2005) uses WordNet to measure word similarity to obtain a better textual entailment recognizer. Chen et al. (2018) proposes a neural network model for natural language inference equipped with several external knowledge. Wang et al. (2019) finds that utilizing ConceptNet as an external knowledge source can benefit entailment model in scientific domain. Chen et al. (2020b) proposes WIKINLI, a large-scale naturally annotated dataset constructed from Wikipedia category graph. And they show that model pretrained on this dataset can achieve better performance on downstream natural language entailment tasks.

3 FEVER Challenge

In this paper, we tackle the large-scale challenge for fact extraction and verification: the FEVER Challenge (Thorne et al., 2018). It contains 185,445 claims generated by altering sentences extracted from Wikipedia.

To verify a claim in FEVER, a model typically follows a three-step pipeline framework, i.e. document retrieval, sentence selection and claim verification. In document retrieval, a system matches the claim to Wikipedia articles by extracted named entities and phrases using a search engine built on Wikipedia. In sentence selection, a system ranks the sentences from retrieved articles by their similarity scores against the claim. The similarity score can be calculated by a trainable regression model, like Enhanced LSTM (Chen et al., 2017), or pretrained language models like BERT and RoBERTa (Devlin et al., 2019; Liu et al., 2019). In claim verification, a system classifies the truthfulness of the claim based on top-ranked sentences from the previous step, also known as the evidence sentences. Like most participants in this challenge, we adopt existing approaches for document retrieval and sentence selection, while mainly focusing on the claim verification model.

4 Method

In this section, we first formalize the fact verification problem in Section 4.1 and then introduce our model for incorporating entity knowledge in Section 4.2. Finally, we present our complete solution to the FEVER Challenge including document retrieval, evidence selection and entity description collection in Section 4.3.

4.1 Problem Formulation

Given the input claim and its retrieved evidence sentences, our approach predicts the truthfulness of the claim. As defined in FEVER dataset, we frame the prediction as a three-way classification, i.e. the prediction is '*SUPPORTED*', '*REFUTED*' or '*NOT ENOUGH INFO (NEI)*'. Furthermore, we require the model to predict the relatedness of the evidence sentences as an auxiliary task.

Formally, the input to our model is $[C, E_1, E_2, E_3, \cdots, E_n]$, where C is the claim and E_i is the i-th evidence. The evidence sentences are obtained by an upstream retrieval module. The claim and each evidence are composed of a list of tokens: $C = [w_{c_1}, w_{c_2}, \cdots, w_{c_{|C|}}]$, $E_i = [w_{e_{i_1}}, w_{e_{i_2}}, \cdots, w_{e_{i_{|E_i|}}}]$. The target output is the claim truthfulness label y_c. Also, the FEVER dataset provides a relatedness label for each evidence sentence as auxiliary targets, i.e. $y_{e_i} \in \{$'RELATED', 'NOT RELATED'$\}$.

4.2 Model Architecture

The general architecture of our fact verification model is shown in Figure 1. It is a classification neural network based on RoBERTa (Liu et al., 2019), a Transformer-based model (Vaswani et al., 2017) pretrained on large corpora with a masked language modeling objective.

We concatenate the claim with evidences as input to the model. Following the default configuration of RoBERTa, we insert a [CLS] token at the start of the input; the output representation of this token is used to aggregate information from the whole sequence. And we insert a token [SEP] before each evidence sentence as an indicator of sentence boundaries. We use the output vectors of these [SEP] tokens as features for the evidence sentence after it.

The modified text is then represented as a sequence of tokens $X = [w_1, w_2, \cdots, w_n]$. Each token w_i is assigned three types of embeddings: *token embeddings* indicate the meaning of each token, *position embeddings* indicate the position of each token within the text sequence, and *segmentation embeddings* are used to discriminate between the claim and the evidence sentences[1]. These three embeddings are summed into a single input vector x_i and fed to a bidirectional Transformer with multiple layers:

$$\tilde{H}^l = \mathrm{LN}(H^{l-1} + \mathrm{Att}(H^{l-1}, H^{l-1})) \quad (1)$$

$$H^l = \mathrm{LN}(\tilde{H}^l + \mathrm{FFN}(\tilde{H}^l)) \quad (2)$$

where $H^0 = x$ are the input vectors; LN is the layer normalization operation (Ba et al., 2016); Att is the multi-head self-attention operation (Vaswani et al., 2017); the superscript l indicates the depth of the stacked layers. On the top layer, RoBERTa generates an output vector for each token with rich contextual information for fact verification.

As shown in the motivating examples in Table 1, making the correct prediction needs good understanding and reasoning of the entities in claim and evidence. Thus, we collect entity knowledge from Wikipedia and encode them by a decomposable entity encoder. The result is attended by the previous fact verification module. In the follows, we will first introduce the main fact verification module and then the entity knowledge encoder.

4.2.1 Fact Verification Module

Suppose that in the output contextual embeddings from RoBERTa, $\boldsymbol{c}$ is the vector for the [CLS] token and $\boldsymbol{e}_i$ is the vector for the i-th [SEP] token. To predict the truthfulness of the claim, we apply a three-way softmax classification layer over $\boldsymbol{c}$:

$$\hat{y}_c = \mathrm{softmax}(\boldsymbol{c}\boldsymbol{W}_c + \boldsymbol{b}_c) \quad (3)$$

where $\boldsymbol{W}_c$ and $\boldsymbol{b}_c$ are weight and bias. We adopt the cross entropy loss for claim truthfulness classification against the ground-truth label y_c.

Since the input evidence sentences are obtained by an upstream retrieval module, some of them may be irrelevant to the claim. Therefore, as an auxiliary training task, we also predict the relatedness of each evidence sentence, which has been shown to be effective in Yin and Roth (2018).

[1]RoBERTa does not use segmentation embeddings in pretraining, but we found it is useful in finetuning.

To do that, we apply a sigmoid classification layer over each $\boldsymbol{e}_i$:

$$\hat{y}_{e_i} = \sigma(\boldsymbol{e}_i\boldsymbol{W}_e + \boldsymbol{b}_e) \quad (4)$$

where $\boldsymbol{W}_e$ and $\boldsymbol{b}_e$ are weight and bias, and σ is the sigmoid function. Likewise, we adopt cross entropy loss for this binary classification of evidence relatedness against the ground-truth label y_{e_i}.

The final loss L for our fact verification module is the weighted summation of the claim loss L_c and the evidence loss L_e:

$$L = \lambda L_c + (1 - \lambda) L_e \quad (5)$$

where λ is searched from $\{0.1, 0.3, 0.5, 0.7, 0.9\}$ based on model performance on the development set. It is set to 0.5 to achieve the best performance.

4.2.2 Decomposable Entity Knowledge Encoder

To augment fact verification model with external entity knowledge, we first identify all named entities in the claim and evidence sentences with an external named entity recognizer. We then link these entities to Wikipedia articles with a trained entity linker (more details in Section 4.3). The lead section of the corresponding Wikipedia article is used as the description of an entity.

While a straightforward approach is to append these descriptions to the input claim and evidence, it may lead to two potential issues. First, since entity descriptions are retrieved from Wikipedia articles, they could contain irrelevant noisy information and degrade the model performance. Second, many descriptions are very long and can reduce our model's efficiency in both training and inference. Therefore, we propose a decomposable entity knowledge encoder module to represent this external entity information in a compact semantic space.

We denote the fact verification module in Section 4.2.1 as T_m. We co-train a lightweight entity knowledge encoder module T_e initialized with the distilled RoBERTa-base (Sanh et al., 2019). Thus, T_e has less parameters and fewer layers than T_m and the hidden state dimension of T_e, i.e. d_e, is smaller than that of T_m, i.e. d_m.

We concatenate descriptions of entities in the claim and evidence sentences and feed the concatenated text into T_e. We denote the input hidden states to the l-th layer in T_m as H_m^{l-1} and the input hidden states to the l-th layer in T_e as H_e^{l-1}.

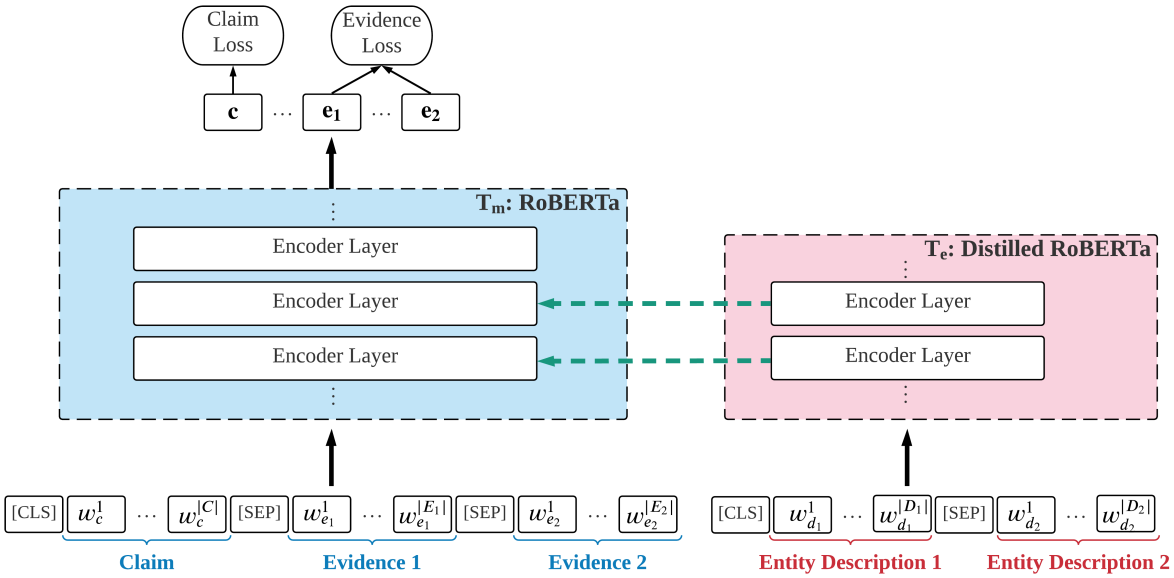

Figure 1: Architecture of our fact verification system enhanced with entity knowledge. The left part is the fact verification module based on RoBERTa and the right part is the entity encoder based on distilled RoBERTa. The dotted green arrows indicate unidirectional attention mechanism which the fact verification module uses to access outputs from the entity encoder. The final loss of our model is the combination of claim loss and auxiliary evidence loss.

Then, the fact verification module T_m employs a unidirectional attention to access outputs from T_e to adopt entity knowledge for fact verification. Since T_m and T_e have different hidden sizes, we first apply a linear transformation to the outputs of T_e:

$$\hat{h}_{e_i}^{l-1} = h_{e_i}^{l-1}\boldsymbol{W}_e^{l-1} + \boldsymbol{b}_e^{l-1} \tag{6}$$

where $h_{e_i}^{l-1}$ is the i-th output of the $(l-1)$-th layer of T_e, and $\boldsymbol{W}_e^{l-1} \in \mathbb{R}^{d_e \times d_m}$, $\boldsymbol{b}_e^{l-1} \in \mathbb{R}^{d_m}$ are weight and bias.

Then the fact verification module T_m conducts unidirectional attention to $\hat{H}_e^{l-1} = \{\hat{h}_{e_i}^{l-1}\}$, along with its self-attention, to produce the output H_m^l.

$$\begin{aligned}\tilde{H}_m^l =& \mathrm{LN}(H_m^{l-1} \\ &+ \mathrm{Att}(H_m^{l-1}, [H_m^{l-1}, \hat{H}_e^{l-1}])) \qquad (7)\\ H_m^l =& \mathrm{LN}(\tilde{H}_m^l + \mathrm{FFN}(\tilde{H}_m^l)), \qquad (8)\end{aligned}$$

where $[*, *]$ indicates the element-wise concatenation of two lists of vectors.

And the entity knowledge encoder T_e carries out its self-attention as in standard Transformer models.

$$\begin{aligned}\tilde{H}_e^l =& \mathrm{LN}(H_e^{l-1} + \mathrm{Att}(H_e^{l-1}, H_e^{l-1})) \qquad (9)\\ H_e^l =& \mathrm{LN}(\tilde{H}_e^l + \mathrm{FFN}(\tilde{H}_e^l)) \qquad (10)\end{aligned}$$

Since T_m has more encoding layers than T_e, the unidirectional attention only works on the lower layers of T_m where it has a corresponding layer in T_e.

In this way, the fact verification module can efficiently reason about the truthfulness of the claim with the compact representations of rich entity information from the entity knowledge encoder.

4.3 Complete Solution

In this section, we introduce our complete solution to the FEVER Challenge of fact verification.

Document Retrieval We adopt the same document retrieval module as in (Hanselowski et al., 2018; Liu et al., 2020). For a given claim, it first utilizes the constituency parser in AllenNLP (Gardner et al., 2018) to extract all phrases which potentially indicate entities. Then it uses these phrases as queries to find relevant Wikipedia pages through the online MediaWiki API. Then the highest-ranked results are retrieved and further filtered by a set of rules.

Evidence Selection We use the evidence selection module from Liu et al. (2020) to select related sentences from the retrieved Wikipedia pages. The module consists of a regression model based on BERT to score the claim and evidence sentence

pair. For each claim, we use the top 5 ranked sentences as evidence.

Entity Descriptions We first use Flair (Akbik et al., 2019) as the NER tool to extract entities from input claim and evidence sentences. We then use the entity linking system REL (van Hulst et al., 2020) to link entities to Wikipedia articles, and take the first section of the linked article as the entity description. We limit the length of any entity description to 100 tokens, and the total length of all descriptions for one instance to 512 tokens.

Claim Verification Finally, the claim, evidence sentences and entity descriptions are fed into our model in Section 4.2 to verify the claim's truthfulness.

5 Experiments

5.1 Dataset and Evaluation Metrics

We evaluate our model on FEVER 1.0 (Thorne et al., 2018), a large-scale benchmark dataset for fact extraction and verification. Detailed statistics of FEVER are shown in Table 2. Each instance in FEVER 1.0 consists of a human-written claim, a set of ground-truth evidence sentences from Wikipedia and a label (i.e., 'SUPPORTED', 'REFUTED' or 'NOT ENOUGH INFO'), indicating the truthfulness of claim. FEVER also provides a Wikipedia dump containing 5,416,537 pre-processed articles for machine learning models to select evidence sentences.

Models are evaluated by two metrics: label accuracy and FEVER score. Label accuracy measures the accuracy of model's prediction for claim truthfulness. FEVER score considers whether both the predicted claim truthfulness and the selected evidence sentences are correct.

5.2 Implementation details

We implement our model with Huggingface Transformers (Wolf et al., 2020). The training batch size is set to 32. We use the Adam optimizer (Kingma and Ba, 2014) with a learning rate of 1e-5 and a warm-up proportion of 0.1. The main encoder is initialized with RoBERTa-large. It has 355M parameters with 24 layers and a Transformer hidden size of 1,024. The entity encoder is initialized with distilled RoBERTa-base with 82M parameters, 6 layers and a Transformer hidden size of 768. We train our models for 10 epochs and the model achieving the highest label accuracy on development set of FEVER is selected. All source codes of this work are available at https://github.com/nlpyang/FeverEntity.

Data Split	SUPPORTED	REFUTED	NEI
Train	80,035	29,775	35,639
Dev	6,666	6,666	6,666
Test	6,666	6,666	6,666

Table 2: Statistics of the FEVER dataset.

5.3 Baselines

We compare our system to the following top-performing systems on the FEVER shared task.

- Athene (Hanselowski et al., 2018) models each claim-evidence pair separately and applies a pooling operation for feature aggregation.
- UCL MRG (Yoneda et al., 2018) uses Convolutional Neural Network as the encoder for claim and evidence. Label aggregation is used for final prediction.
- UNC NLP (Nie et al., 2019) designs a semantic matching neural model for both sentence selection and claim verification.
- GEAR (Zhou et al., 2019) constructs a graph with each evidence sentence as a node and uses a graph neural network over this graph for prediction.
- DREAM (Zhong et al., 2020) is built upon a graph derived from semantic role labeling and embeds a graph-based module into the pretrained XLNet (Yang et al., 2019) model.
- KGAT (Liu et al., 2020) uses Kernel Graph Attention Network over a graph with evidence sentences as nodes. The model is based on the pretrained RoBERTa-large model.
- LOREN (Chen et al., 2020a) uses design a neural network that can aggregate probabilistic version of the logic rules for fact verification.

5.4 Results

Table 3 shows the results on both the development set and blind test set of FEVER. The first block in the table includes the results of systems without using pretrained models. The second block in

Method	Dev		Test	
	Label Acc.	FEVER score	Label Acc.	FEVER score
Non-pretrained Systems				
Athene	68.49	64.74	65.46	61.58
UCL MRG	69.66	65.41	67.62	62.52
UNC NLP	69.72	66.49	68.21	64.21
Pretrained Systems				
GEAR (*BERT-Base)*	74.84	70.69	71.60	67.10
DREAM (*XLNet)*	79.16	-	76.85	70.60
KGAT				
⌞ *BERT-Large*	77.91	75.86	73.61	70.24
⌞ *RoBERTa-Large*	78.29	76.11	74.07	70.38
⌞ *CorefRoBERTa-Large*	-	-	75.96	72.30
LOREN (*RoBERTa-Large)*	81.12	78.94	73.43	74.84
Our Approach				
⌞ *BERT-Large*	78.17	75.44	75.06	71.19
⌞ *RoBERTa-Large*	81.43	78.65	77.29	72.89

Table 3: Evaluation of fact verification systems on FEVER dataset. Variants of a system using different pretrained models are listed.

Entity Description	Entity Knowledge Encoder	Evidence Loss	Label Accuracy	FEVER Score
✓	✓	✓	81.43	78.65
✓	×	✓	80.89	77.93
✓	✓	×	80.15	77.30
×	×	✓	80.44	77.65
×	×	×	79.87	76.99

Table 4: Ablation study on FEVER development set. Our model is based on RoBERTa-Large in this experiment.

the table includes systems using pretrained models. The last block includes results from our framework. For a fair comparison with previous systems, we also implemented a version of our model based on BERT-Large.

As shown, our approach achieves the best performance on label accuracy and competitive results on FEVER score in both development and test set, proving the effectiveness of our entity knowledge-based approach. We also show that this improvement is consistent across different underlying pretrained models. For instance, our approach of knowledge integration outperforms KGAT when the language understanding model is initialized with either BERT-Large or RoBERTa-Large.

Ablation Study To further investigate how our proposed system improves fact verification, we conduct ablation study of different model components. Table 4 presents the result on the development set after removing different components in our model based on RoBERTa-Large.

As shown, when the entity knowledge encoder is removed and the entity description is concatenated with the claim and evidence sentences as input, the label accuracy drops 0.5%. This proves the necessity of using a separate module to represent external knowledge. When evidence loss is removed, the label accuracy drops almost 1.3%. When entity description is not used at all, the FEVER score drops 1%. When evidence loss is further removed, the FEVER score drop increases 1.6%.

These results show that our proposed entity descriptions, entity knowledge encoder and evidence loss all contribute to the effectiveness of our model.

5.5 Error Analysis

To take a deeper investigation into current fact verification systems, we manually analyze 100 randomly selected cases that are incorrectly predicted by our model. We summarize several primary error types in this section.

The first error type is the failure of inference over multiple sentences. About 44% claims in

FEVER development set have more than one gold evidence sentences. Identifying the truthfulness of these claims sometimes requires multi-sentence inference. For example, to verify the claim *"Hourglass was released 6 years after New Moon Shine."*, we need to infer over two evidence sentences: *"It built upon the success of his previous effort, New Moon Shine."* and *"Taylor's first studio album in six years was released in 1997 to glowing notices."*

Another primary error type is mistakes in semantic matching. Given the claim *"Valencia is in a country."*, although model successfully retrieved the gold evidence *"Valencia is the capital of the autonomous community of Valencia and the third largest city in Spain ..."*, it still fails to predict it correctly. We believe one possible reason is that model doesn't realize being a city of a country is synonymous with being in a country. This suggests we need more powerful language representation models to tackle fact verification.

The third error type is caused by ambiguity of concepts in the claim. For example, the claim describes *"Bones is a movie."*, and our model predicts its to be true based on retrieved evidence *"Bones is a 2001 American horror film directed ..."*. However, there are different definitions of *"Bones"* in Wikipedia and the human annotator was referring to the TV series also named *"Bones"*.

6 Conclusion

In this paper, we present a novel framework for fact verification. When assessing the truthfulness of a claim, we first identify the entities within the claim and evidence, and then retrieve external entity descriptions from Wikipedia. We design a decomposable entity knowledge encoder with unidirectional attention for effectively incorporating entity knowledge. Furthermore, we propose to use the prediction of input evidence sentences' relatedness as an auxiliary task. Experimental results show that our model achieves competitive results on the large-scale fact verification dataset FEVER. And we conduct ablation studies to showcase the effectiveness of our proposed components.

References

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, Minneapolis, Minnesota.

Gabor Angeli and Christopher D. Manning. 2014. NaturalLI: Natural logic inference for common sense reasoning. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 534–545, Doha, Qatar.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Jiangjie Chen, Qiaoben Bao, Jiaze Chen, Changzhi Sun, Hao Zhou, Yanghua Xiao, and Lei Li. 2020a. Loren: Logic enhanced neural reasoning for fact verification. *arXiv preprint arXiv:2012.13577*.

Mingda Chen, Zewei Chu, Karl Stratos, and Kevin Gimpel. 2020b. Mining knowledge for natural language inference from Wikipedia categories. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, Online.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Diana Inkpen, and Si Wei. 2018. Neural natural language inference models enhanced with external knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Melbourne, Australia.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced LSTM for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1657–1668, Vancouver, Canada.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2020c. Tabfact: A large-scale dataset for table-based fact verification. In *Proceedings of the ICLR Conference*.

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.

William Ferreira and Andreas Vlachos. 2016. Emergent: a novel data-set for stance classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, San Diego, California.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, Melbourne, Australia.

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, Brussels, Belgium.

Valentin Jijkoun, Maarten de Rijke, et al. 2005. Recognizing textual entailment using lexical similarity. In *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*, pages 73–76. Citeseer.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *Proceedings of the ICLR Conference*.

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 the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7342–7351.

Ndapandula Nakashole and Tom M. Mitchell. 2014. Language-aware truth assessment of fact candidates. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1009–1019, Baltimore, Maryland.

Yixin Nie, Haonan Chen, and Mohit Bansal. 2019. Combining fact extraction and verification with neural semantic matching networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6859–6866.

Farhad Nooralahzadeh and Lilja Øvrelid. 2018. Siriusltg: An entity linking approach to fact extraction and verification. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 119–123.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

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.

Motoki Taniguchi, Tomoki Taniguchi, Takumi Takahashi, Yasuhide Miura, and Tomoko Ohkuma. 2018. Integrating entity linking and evidence ranking for fact extraction and verification. In *Proceedings of the First Workshop on Fact Extraction and Verification (FEVER)*, pages 124–126.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana.

Johannes M van Hulst, Faegheh Hasibi, Koen Dercksen, Krisztian Balog, and Arjen P de Vries. 2020. Rel: An entity linker standing on the shoulders of giants. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2197–2200.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30.

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*, Baltimore, MD, USA.

Soroush Vosoughi, Deb Roy, and Sinan Aral. 2018. The spread of true and false news online. *Science*, 359(6380):1146–1151.

William Yang Wang. 2017. "liar, liar pants on fire": A new benchmark dataset for fake news detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 422–426, Vancouver, Canada. Association for Computational Linguistics.

Xiaoyan Wang, Pavan Kapanipathi, Ryan Musa, Mo Yu, Kartik Talamadupula, Ibrahim Abdelaziz, Maria Chang, Achille Fokoue, Bassem Makni, Nicholas Mattei, et al. 2019. Improving natural language inference using external knowledge in the science questions domain. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33.

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 the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763.

Wenpeng Yin and Dan Roth. 2018. TwoWingOS: A two-wing optimization strategy for evidential claim verification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 105–114, Brussels, Belgium.

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, Brussels, Belgium.

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 the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6170–6180, Online.

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