Toward Automatic Misinformation Detection Utilizing Fact-checked Information

Anonymous ACL submission

Abstract

We proposed a new task FCCKB: Fact-checking by Claim Knowledge Base. The goal was to fact-check a sentence utilizing verified claims stored in the database. To retrieve relevant claims from the large database, we proposed applying Semantic Role Labeling(SRL) on the input sentence having rich semantics and then encoding the results to get fine-grained sentence embeddings. That improved semantic matching between the input sentence and the relevant claims. We used three sentence encoders for sentence encoding. In FEVER dataset, precision and recall was improved by more than 5 percent after SRL was applied.

1 Introduction

Misinformation has been raging on social media, which indicates the urgency of fact-checking. Traditional fact-checking relies on professionals. Before fact-checking a rumor, they have first to find out unverified rumors from thousands of ones. To avoid wasting human resources, it is valuable to explore a way to detect misinformation automatically using information verified in the past.

The shared task FEVER(Thorne et al., 2018) aims to verify a claim by finding evidence from Wikipedia. Recent work(Nie et al., 2019; Yoneda et al., 2018; Hanselowski et al., 2018; Malon, 2018; Liu et al., 2020) that engaged in fact-checking used FEVER as a benchmark. However, we found that FEVER's problem is not suitable to deal with rumor verification. In FEVER, a sentence to be verified is typically a single claim, and the truth value of evidence is always true. However, a sentence in a rumor usually has multiple claims. Much of rumors contains misinformation, which let them be evidence with a truth value of falsity. On the other hand, evidence in FEVER is typically a complicated sentence, which makes an ambiguity of fact-checking since a sentence may not be totally wrong. If we labeled an entire sentence as a false evidence, a sentence might be classified as misinformation even when it makes a true claim in the evidence. Hence, we proposed a new task FCCKB: Fact-checking with Claim Knowledge Base. The problem definition of FCCKB is a better fit for rumor verification. To solve the problem, we build a fact-checking system, and use Semantic Role Labeling(SRL) to improve the part of semantic matching in our system.

2 Related Work

As early as 2014, Vlachos et al.(Vlachos and Riedel, 2014) have formulated fact-checking as a stance classification task in which statements from journalists are provided to verify the authenticity of a claim. Ferreira et al. (Ferreira and Vlachos, 2016) proposed Emergent, a dataset consisting of suspicious claims. Each claim has relevant articles and their stances toward the claim. Though they rigorously defined the stance of an article, the data they released did not follow the definition they made. Kotonya et al.(Kotonya and Toni, 2020) focused on news in the healthcare domain. Given a claim, they generate an explanation from a fact-checking report by abstractive summarization. However, any clues which can support or deny the claim might be hidden in the context of the original fact-checking reports. The summarization model they used ignored information in the claim so that the summary might be a sentence irrelevant to the given claim.

We use the term *evidence* to denote a statement or an article that can be used to infer the authenticity of information to be verified. The above work only considered the stance of evidence but neglected how to get the evidence. The first step of automatic fact-checking is to retrieve evidence from a trusty knowledge base. FEVER(Thorne et al., 2018) is a shared task that is most relevant to our work. The goal is to retrieve evidence from Wikipedia to verify the authenticity of a given claim. Related work(Nie et al., 2019; Yoneda et al.,

2018; Hanselowski et al., 2018; Malon, 2018) followed the procedures: (1) document retrieval, (2) sentence selection for evidence, (3) Natural Inference Language (NLI). In the first stage, documents with keywords in the claims were retrieved. As not all the sentences in the documents were related to the claim, relevant sentences were further picked out as evidence. Finally, an NLI model was used to determine whether there exists a contradiction between the claim and evidence.

As data in FEVER was collected from Wikipedia, Qifei Li et al.(Li and Zhou, 2020) tried to bridge the gap between FEVER and fake news detection. They applied similar procedures on fake news verification. First, the news to be verified was summarized into a claim using a summarization model, and the claim was viewed as a query for document retrieval via Google Search Engine. They used Sentence-BERT(Reimers and Gurevych, 2019) to encode each sentence in the retrieved documents. Finally, sentences most similar to the claim were selected as evidence for authenticity prediction. As mentioned previously, summarizing an article might lose information in the article. Neither did they consider the source's credibility.

2.1 Semantic Textual Similarity

Semantic Textual Similarity (STS) is a task to evaluate the level of similarity between two texts. It has benefited from the success of BERT(Devlin et al., 2018), which concatenated two sentences as input and achieved state-of-the-art. However, this way is infeasible when many sentences should be considered. Sentence-BERT(Reimers and Gurevych, 2019), is a siamese BERT model which encodes each sentence into a vector. Compared to BERT, it reduces time spent on the forward process, which usually consists of several matrix multiplications. Suppose there are N sentences, and we want to find out the most semantically similar pair of sentences. Simply concatenating two sentences as input into BERT to get their similarity takes $N(N-1)/2$ times on forwarding. On the contrary, encoding each sentence into a vector takes N times. We only need to search the closest vectors with a minimum angle in the vector space.

2.2 Semantic Role Labeling

An event typically can be described with a predicate and several arguments. Those Arguments might be an *Agent* (the causer), a *Theme* (the patient), a *Location* or *Time* that the event occurred at and so on. The goal of *Semantic Role Labeling* (SRL) [1]is to find out the predicate-arguments relations in a sentence.

Most of the benchmarks used in related work are built on PropBank(Kingsbury and Palmer, 2002). In PropBank, a verb might have several senses and the corresponding arguments with semantic roles. Arguments are labeled with numbers. Table 1 shows examples provided by PropBank[2]. Additionally, arguments were labeled at the constituent level. Tasks(Carreras and Màrquez, 2005) following this way are called span-based SRL. Another annotation policy is annotating only the head of argument constituent. Tasks(Surdeanu et al., 2008; Hajič et al., 2009) are called dependency-based SRL if they follow this policy.

The application of SRL has been explored since ten years ago. Tsai et al.(Tsai et al., 2007) built a SRL system to improve information extraction in the biomedical domain. Lai et al.(Lai et al., 2016) used SRL to extract subject-verb–object (SVO) triplet and mapped it to biological expression language (BEL).

In (He et al., 2018; Li et al., 2019), the end-to-end approach is used to predict all predicate-arguments relations. Subsequently, Shi et al.(Shi and Lin, 2019) utilized the BERT-based model to achieve the state-of-the-art performance.

3 Problem Definition

We are going to point out the inadequacy of FEVER for rumor verification. After that, we will define our problem which is a better fit for the scenario of rumor verification.

3.1 Inadequacy of FEVER

In FEVER, a claim is a sentence to be verified. However, here we redefine *claim* as an atomic statement with a single predicate since we found claims in FEVER with such characteristics.

We observed that FEVER did not fit the scenario of rumor verification in three aspects. Firstly, a sentence to be verified is typically a single claim in FEVER. However, a sentence in a rumor usually has multiple claims. The following sentence consists of three claims at least.

[1]a great material for SRL https://web.stanford.edu/~jurafsky/slp3/old_oct19/20.pdf

[2]https://github.com/propbank/propbank-documentation/raw/master/annotation-guidelines/Propbank-Annotation-Guidelines.pdf

ARG0	agent	ARG3	starting point, benefactive, attribute
ARG1	patient	ARG4	ending point
ARG2	instrument, benefactive, attribute	ARGM	modifier

Table 1: Arguments in PropBank

Sentence According to a report, Foodwatch, a German food inspection organization, spot-checked more than 20 brands of local snacks and found that 3 of them contained carcinogens. Among them, Kinder Reigel has the highest content of mineral oil aromatic hydrocarbons, reaching 1.2mg/kg.)

Claim 1 Foodwatch spot-checked more than 20 brands of local snacks.

Claim 2 Foodwatch found that three brands of food contained carcinogen.

Claim 3 Kinder Reigel has the highest content of mineral oil aromatic hydrocarbons.

It is more difficult to fact-check a sentence with multiple claims than a single one since we need to first parse out all the claims in the sentence and verify them.

Secondly, we assume that the truth value of evidence is always true in FEVER. However, it is usually false in the scenario of rumor verification. For example, the sentence mentioned above would be evidence with a false truth value after verification, as the third claim is false.

Thirdly, evidence is usually a complicated sentence with multiple claims in FEVER. Labeling such a sentence as a false statement makes ambiguity. Though the entire sentence mentioned above is false, the first two claims are true. If we adopt the sentence as evidence, a rumor with only the first two sub-claims might be viewed as misinformation. This problem can be avoided by only labeling the third claim as a false statement. In other words, evidence should be a single claim.

3.2 Problem Definition of FCCKB

For the above reasons, we present a new task: FCCKB: Fact-checking by Claim Knowledge Base. Assuming that $C = \{c_1, ..., c_n\}$ is a claim set with size n in the database. Each claim has been labeled with its truth value. Given a sentence s in a rumor, to verify whether s is true or false, the goal is to know whether s supports or denies c_i for some $i \in \{1, ..., n\}$. Figure 1 shows an example

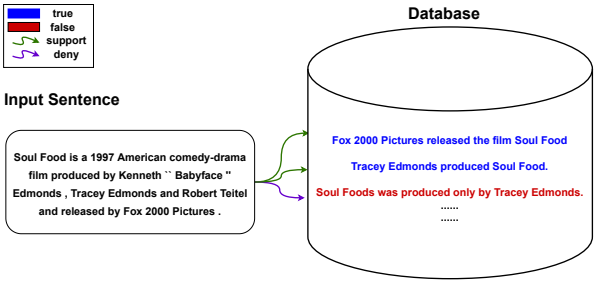

Figure 1: Task: FCCKB

under our problem definition. The left-hand side is the input sentence, and the right-hand side is the database composed of many verified claims.

We say s supports c if some part of s has the same meaning with c and s denies c if some part of s has the reverse meaning with c. The following examples show how to fact-check a sentence using the stance(support or deny) of s toward c:

Claim 1 "Lemon belongs to the citrus category."

Claim 2 "Lemon can prevent cancer."

Sentence 1 "Lemon is a kind of citrus fruit and has been proved to be effective in cancer prevention."

Sentence 2 "Rumor has it that lemon is a kind of citrus fruit and can prevent cancer. It has been verified to be misinformation."

Here we use $c1$, $c2$, $s1$, $s2$ to denote claim 1, 2 and sentence 1, 2 respectively. Assuming that we know $c1$ is true and $c2$ is false. The first half of $s1$ has the same meaning with $c1$ and the second half has the same meaning with $c2$. Since $c1$ is true, it is fine that $s1$ supports $c1$. However, $c2$ is false and $s1$ supports it which means $s1$ is also false. In contrast, $s2$ has the reverse meaning with $c1$ and $c2$, i.e. $s2$ denies $c1$ and $c2$. As $c2$ is false, it is fine that $s2$ denies a false claim. However, $c1$ is true and $s2$ denies it which means $s2$ is false.

Note that in the real world, there might be several sentences in a rumor support or deny a claim, but none of the sentences supports or denies the claim individually

Sentence 3 "Recent research shows that lemon has the following effects:"

Sentence 4 "Cancer prevention"

It is an example in which Sentence 3 and Sentence 4 support $c2$ but neither of them supports $c2$. In FCCKB, we ignore cases like this and only consider fact-checking at the sentence level.

4 Methodology

4.1 Basic Model

Our basic model is a pipeline with two stages - (1) Claim Retrieval and (2) Authenticity Inference. Figure 2 shows how the pipeline works. Before running the pipeline, each claim c_j in the database is encoded to an embedding $\mathbf{e_{c_j}}$. In the first stage, the input sentence s was used to retrieve relevant claims by the similarity strategy. After claims were retrieved, the authenticity of s was decided by an authenticity predictor.

4.2 Similarity Strategy

Algorithm 1: Similarity Strategy

Input : a sentence s
claim embeddings
$\mathbf{E_c} = \begin{pmatrix} \mathbf{e_{c_1}} & \cdots \mathbf{e_{c_n}} \end{pmatrix}$
Output : a set of claims indexes I in which the corresponding claims are most similar to s
Option : K = number of claims to be retrieved

Function `BasicSimilarityStrategy`$(s, \mathbf{E_c}, K)$**:**
  $\mathbf{e_s}$ = Encoder(s);
  scores = $\mathbf{e_s}^T \mathbf{E_c}$;
  return argpartition(**scores**, K);

Function `SRLSimilarityStrategy`$(s, \mathbf{E_c}, K)$**:**
  $frames$ = SRL(s);
  $\mathbf{E_q}$ = Encoder($(s; frames)$);
  scores = column-wise-max($\mathbf{E_q}^T \mathbf{E_c}$);
  return argpartition(**scores**, K);

Algorithm 1 shows details about the similarity strategy. BasicSimilarityStrategy computes similarity scores between two texts by the dot product of their embeddings. SRLSimilarityStrategy is our proposed solution. Note that *argpartition(scores, K)* returns the array indices with the highest K values in *scores*.

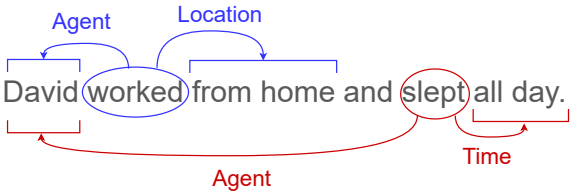

Figure 3: An Example of Semantic Roles

Let us explain the reason why we use SRL to improve the basic similarity strategy. The sentence in Figure 3 is composed of two events - "David worked from home." and "David slept all day." Each of them has a predicate and the corresponding arguments. As a sentence might have rich semantics, we hypothesize that encoding the entire sentence will result in the ambiguity of the embedding in semantics. Hence, we need to parse out all the events in the sentence.

Let *frame* denotes a subsentence composed of a predicate and its associated arguments. The goal of SRL is to find out all frames in a sentence. Here is an example generated by the SRL model from AllenNLP[3].

frame1 [ARG0: David] [V: worked] [ARGM-LOC: from home] and slept all day

frame2 [ARG0: David] worked from home and [V: slept] [ARGM-TMP: all day]

We can regard a frame as the basic unit (as part of a sentence) with semantics, and SRL can help to find out all the frames in a sentence. Therefore, we proposed using SRL to first find all frames in the input sentence and consider those frames when computing the similarity with claims in the database.

Suppose that the sentence has m frames, $E_q \in \mathbb{R}^{(m+1)\times d}$ is their embedding matrix, and $E_c \in \mathbb{R}^{n\times d}$ is the embedding matrix of claims in the database.

[3] https://demo.allennlp.org/semantic-role-labeling

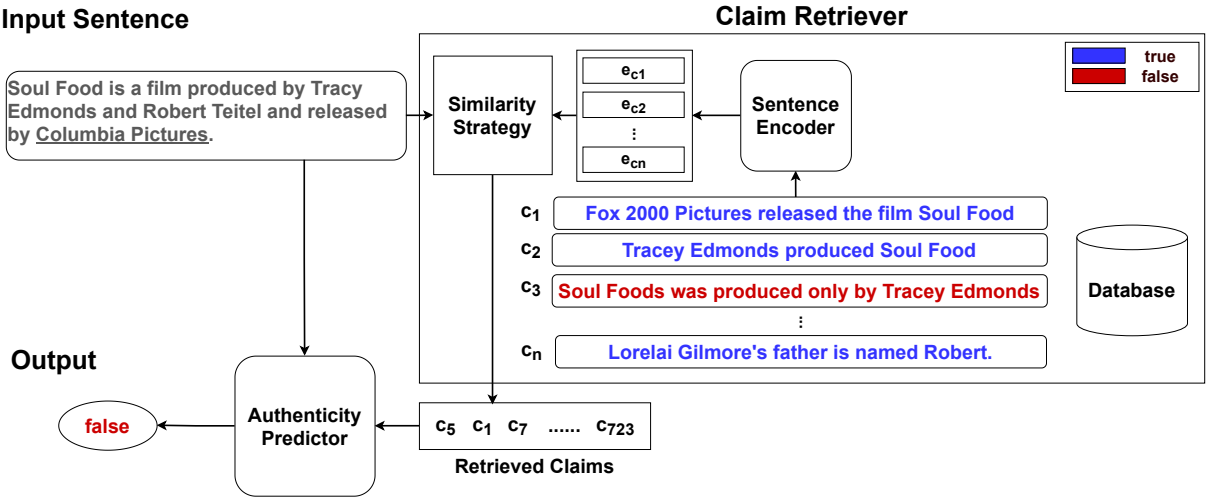

Figure 2: Fact-checking Pipeline

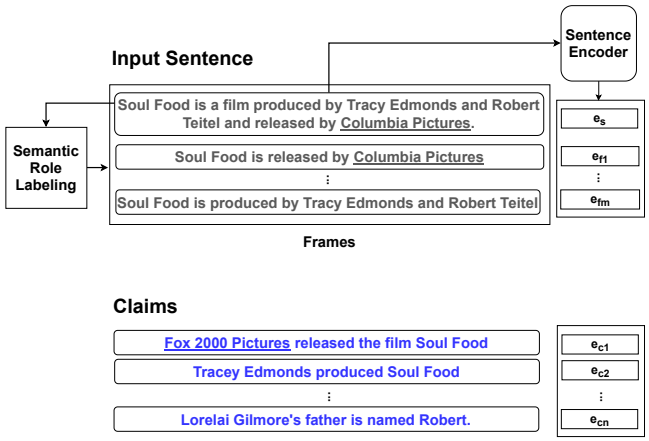

Figure 4: SRLSimilarityStrategy

Algorithm 2: Authenticity Predictor

Input : a sentence s
a list of retrieved claims and their truth value
$L = [(c_1^s, t_1^s), ..., (c_K^s, t_K^s))$
Output : the authenticity of s

```
Function Predict(s, L):
    for c, t in L do
        r = NLI(s, c);
        if (r is Support and t is False) ||
         (r is Deny and t is True) then
            return False;
        end if
    end for
    return True;
```

The similarity between the sentence and all the claims will be:

$$scores(s, c_1, ..., c_n) = \text{column-wise-max}(E_q^T E_c)$$

where

$$E_q \cdot E_c^\intercal = \begin{pmatrix} v_{c_1}^s & v_{c_2}^s & \cdots & v_{c_n}^s \\ v_{c_1}^{f_1} & v_{c_2}^{f_1} & \cdots & v_{c_n}^{f_1} \\ \vdots & \vdots & \ddots & \vdots \\ v_{c_1}^{f_m} & v_{c_2}^{f_m} & \cdots & v_{c_n}^{f_m} \end{pmatrix}$$

That means the similarity between the sentence and a claim is decided by the most similar part of them. In Figure 4, the first frame f_1 is most similar to the first claim c_1. Hence, the similarity between the sentence and the claim should be $v_{c_1}^{f_1}$ rather than $v_{c_1}^s$.

4.3 Authenticity Predictor

Algorithm 2 shows how to determine the authenticity of the input sentence. An NLI model was used to predict whether the sentence supports or denies each claim. If the sentence supports a false claim or denies a true claim, the authenticity of the sentence should be false. Otherwise, it should be true.

5 Experiment

In FEVER, each claim is mapped to several evidence sets. Each evidence set may consist of multiple sentences supporting or denying the claim. Each sentence comes from an introduction section of a document in Wikipedia. In our problem, we ignore the context of each sentence. Hence, only sentences that individually support or deny a claim

were considered. Finally, there were 165447 claims in the database, 19964 input sentences in the training set, and 4440 input sentences in the development set.

In the first stage, each sentence was viewed as a query to retrieve relevant claims. We used the pre-trained model provided by AllenNLP(Shi and Lin, 2019) for SRL parsing. One reason is that it was trained on OntoNotes 5.0,(Weischedel et al., 2013) which followed the span-based annotation policy. Frames in span-based SRL are closer to natural language than frames in dependency-based SRL and should be more suitable for semantic matching using sentence embedding.

After parsing frames from the input sentence, we prepended the document title to the input sentence and all the frames. Three different sentence encoders(Cer et al., 2018; Gao et al., 2021; Reimers and Gurevych, 2019) were used to evaluate the effectiveness of our method. Then in the second stage, we used a pre-trained NLI model(Nie et al., 2020) for stance prediction.

5.1 Evaluation

Setting	Accuracy
SBERT	0.790
SBERT + SRL	0.812

Table 2: Accuracy in the Development Set

Figure 5 shows precision and recall of claim retrieval in each setting when K = 1 to 100. After SRL was applied, precision and recall got a significant improvement no matter which encoder was used. Without SRL, the mean average precision is 0.226, 0.094, and 0.105 for SBERT, SimCSE, and USE respectively. With SRL, it is 0.260, 0.151, and 0.166. The results indicate that information will be lost after a complicated sentence is encoded and SRL can help to alleviate the problem.

Since we found F1 reached the highest value at K=5, we used the top five retriteved claims for authenticity prediction. Table 2 shows the accuracy of the entire fact-checking system in the development set when SBERT was used for encoding.

5.2 Case Study

Table 3 shows an example in which **sentence** is the query and **claim** is one of the relevant claims in our database to be retrieved. **Frame 1** to **Frame 3** are SRL results after parsing the **sentence**. There are many redundant words in the input **sentence**, however, only a few words (in highlight) are related to the **claim**. In contrast, **Frame 1** and **Frame 2** almost express the same meaning with **claim** without redundant words.

Claim: Vedam was written and directed solely by Stephen King.
Sentence: Vedam -LRB- English : Chant -RRB- is a 2010 Telugu language Indian drama film written and directed by Radhakrishna Jagarlamudi , starring Allu Arjun , Manoj Manchu , Anushka Shetty , Manoj Bajpayee , Saranya Ponvannan , Deeksha Seth , Lekha Washington , and Siya Gautham . Radhakrishna Jagarlamudi Radhakrishna Jagarlamudi Anushka Shetty Anushka Shetty Allu Arjun Allu Arjun Manoj Manchu Manoj Manchu Deeksha Seth Deeksha Seth Manoj Bajpayee Manoj Bajpayee Saranya Ponvannan Saranya Ponvannan Lekha Washington Lekha Washington Telugu language Telugu language drama film drama film. **Frame 1**: Vedam -LRB-film-RRB.a 2010 Telugu language Indian drama film written by Radhakrishna Jagarlamudi. **Frame 2**: Vedam -LRB-film-RRB.a 2010 Telugu language Indian drama film directed by Radhakrishna Jagarlamudi. **Frame 3**: Vedam -LRB-film-RRB.a 2010 Telugu language Indian drama film written and directed by Radhakrishna Jagarlamudi Allu Arjun , Manoj Manchu , Anushka Shetty , Manoj Bajpayee , Saranya Ponvannan , Deeksha Seth , Lekha Washington , and Siya Gautham . Radhakrishna Jagarlamudi Radhakrishna Jagarlamudi Anushka Shetty Anushka Shetty Allu Arjun Allu Arjun Manoj Manchu Manoj Manchu Deeksha Seth Deeksha Seth Manoj Bajpayee Manoj Bajpayee Saranya Ponvannan Saranya Ponvannan Lekha Washington Lekha Washington Telugu language Telugu language drama film drama film.

Table 3: Case Study

Given the input **sentence**, when top K claims were retrieved, the baseline model could not retrieve the **claim** even at K=100 but our method did it at K=74.

The similarity between the input sentence and the claim was 0.35, which was the lowest one among the similarity scores between the sentence

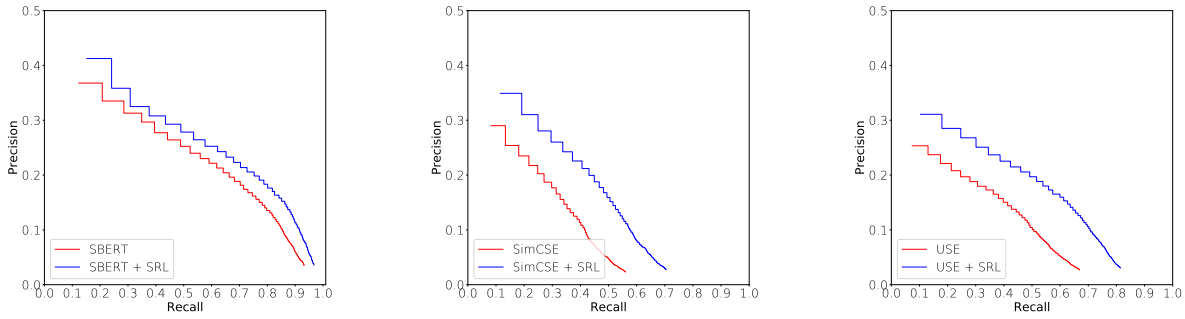

Figure 5: Precision and Recall of Each Setting in the Development Set When K=1 to 100

and frames and the claim. That was the reason why the BasicSimilarityStrategy failed to retrieve the claim. The similarity between the first frame and the claim is 0.44, which was the highest one and increased the possibility that the claim could be retrieved. This example indicates that SRL can help to get a fine-grained embedding for the original sentence.

5.3 Limitation

Our method is limited by the performance of SRL model. In addition, we found that coreference would be a problem even the model performed well. The following frame was parsed out by the SRL model we used.

> **Frame** Soul Food is [ARG 1: a film] [V: produced] [ARG 0: by Tracy Edmonds and Robert Teitel] and released by Columbia Pictures.

The model labeled "a film" rather than "Soul Food" as ARG1. However, "a film" is ambiguous, and "Soul Food" clearly denotes the entity, which implies that coreference problem should further be solved.

6 Conclusion

We proposed a new task FCCKB in which the sentence to be verified is more complicated and closer to a sentence in a rumor in the real world. On the other hand, we use claims, namely, atomic statements, for sentence verification, which will not make the label ambiguous. We proposed to use SRL to improve sentence embeddings for semantic matching. After SRL was applied, the precision and recall increased by more than 5 percent in the FEVER dataset, when three different sentence encoders were used for sentence encoding.

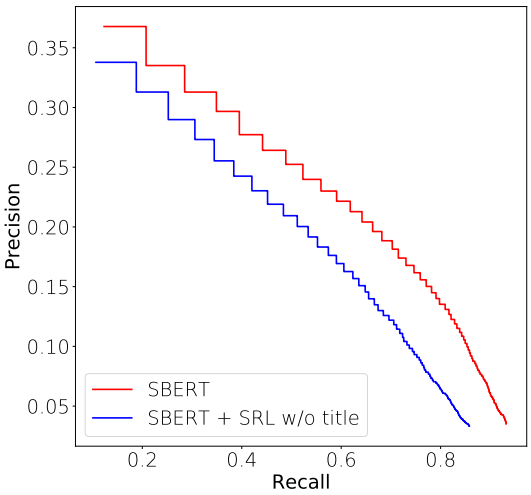

Figure 6: Performence of SBERT and SBERT+SRL w o title

References

Xavier Carreras and Lluís Màrquez. 2005. Introduction to the CoNLL-2005 shared task: Semantic role labeling. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, pages 152–164, Ann Arbor, Michigan. Association for Computational Linguistics.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder. *CoRR*, abs/1803.11175.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

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, pages 1163–1168, San Diego, California. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *CoRR*, abs/2104.08821.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, Pavel Straňák, Mihai Surdeanu, Nianwen Xue, and Yi Zhang. 2009. The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task*, pages 1–18, Boulder, Colorado. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Luheng He, Kenton Lee, Omer Levy, and Luke Zettlemoyer. 2018. Jointly predicting predicates and arguments in neural semantic role labeling. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 364–369, Melbourne, Australia. Association for Computational Linguistics.

Paul Kingsbury and Martha Palmer. 2002. From TreeBank to PropBank. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*, Las Palmas, Canary Islands - Spain. European Language Resources Association (ELRA).

Neema Kotonya and Francesca Toni. 2020. Explainable automated fact-checking for public health claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7740–7754, Online. Association for Computational Linguistics.

Po-Ting Lai, Yu-Yan Lo, Ming-Siang Huang, Yu-Cheng Hsiao, and Richard Tzong-Han Tsai. 2016. BelSmile: a biomedical semantic role labeling approach for extracting biological expression language from text. *Database*, 2016. Baw064.

Qifei Li and Wangchunshu Zhou. 2020. Connecting the dots between fact verification and fake news detection. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1820–1825, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Zuchao Li, Shexia He, Hai Zhao, Yiqing Zhang, Zhuosheng Zhang, Xi Zhou, and Xiang Zhou. 2019. Dependency or span, end-to-end uniform semantic role labeling. *CoRR*, abs/1901.05280.

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, Online. Association for Computational Linguistics.

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

Yixin Nie, Haonan Chen, and Mohit Bansal. 2019. Combining fact extraction and verification with neural semantic matching networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6859–6866.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *CoRR*, abs/1908.10084.

Peng Shi and Jimmy J. Lin. 2019. Simple bert models for relation extraction and semantic role labeling. *ArXiv*, abs/1904.05255.

Mihai Surdeanu, Richard Johansson, Adam Meyers, Lluís Màrquez, and Joakim Nivre. 2008. The CoNLL 2008 shared task on joint parsing of syntactic and semantic dependencies. In *CoNLL 2008: Proceedings of the Twelfth Conference on Computational Natural Language Learning*, pages 159–177, Manchester, England. Coling 2008 Organizing Committee.

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

Richard Tzong-Han Tsai, Wen-Chi Chou, Ying-Shan Su, Yu-Chun Lin, Cheng-Lung Sung, Hong-Jie Dai, Irene Tzu-Hsuan Yeh, Wei Ku, Ting-Yi Sung, and Wen-Lian Hsu. 2007. Biosmile: A semantic role labeling system for biomedical verbs using a maximum-entropy model with automatically generated template features. *BMC Bioinformatics*, 8(1):325.

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, Baltimore, MD, USA. Association for Computational Linguistics.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, Mohammed El-Bachouti, Robert Belvin, and Ann Houston. 2013. OntoNotes Release 5.0.

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. Association for Computational Linguistics.