Evidence Distilling for Fact Extraction and Verification

Yang Lin[1], Pengyu Huang[2], Yuxuan Lai[1], Yansong Feng[1], and Dongyan Zhao[1]

[1] Institute of Computer Science and Technology, Peking University, China
[2] Beijing University of Posts and Telecommunications, China
{strawberry,erutan,fengyansong,zhaodongyan}@pku.edu.cn, hpy@bupt.edn.cn

Abstract. There has been an increasing attention to the task of fact checking. Among others, FEVER is a recently popular fact verification task in which a system is supposed to extract information from given Wikipedia documents and verify the given claim. In this paper, we present a four-stage model for this task including document retrieval, sentence selection, evidence sufficiency judgement and claim verification. Different from most existing models, we design a new evidence sufficiency judgement model to judge the sufficiency of the evidences for each claim and control the number of evidences dynamically. Experiments on FEVER show that our model is effective in judging the sufficiency of the evidence set and can get a better evidence F1 score with a comparable claim verification performance.

Keywords: Claim verification · Fact checking · Natural lauguage inference.

1 Introduction

With the development of online social media, the amount of information is increasing fast and information sharing is more convenient. However, the correctence of such a huge amount of information can be hard to check manually. Based on this situation, more and more attention has been paid to the automatic fact checking problem.

The Fact Extraction and VERification (FEVER) dataset introduced a benchmark fact extraction and verification task in which a system is asked to extract sentences as evidences for a claim in about 5 million Wikipedia documents and label the claim as "SUPPORTS", "REFUTES", or "NOT ENOUGH INFO" if the evidences can support, refute, or not be found for the claim. Fig. 1 shows an example. For the claim" Damon Albarn's debut album was released in 2011", we need to find the Wikipedia document and extract the sentences: "His debut solo studio album Everyday Robots – co-produced by XL Recordings CEO Richard Russell – was released on 28 April 2014 " . Then the claim can be labeled as "REFUTES" and this sentence is the evidence. Different from the traditional fact checking task, fact extraction and verification requires not only checking whether the claim is true, but also extracting relevant information which can support the

verification result from huge amounts of information. In the FEVER shared task, both the F1 score of the evidence and the label accuracy is evaluated as well as FEVER score which evaluate the integrated result of the whole system.

Claim: Damon Albarn's debut album was released in 2011.
Predicted document: [wiki/Damon_Albarn]
Selected sentences:
[1] His debut solo studio album Everyday Robots -- co-produced by XL Recordings CEO Richard Russell -- was released on 28 April 2014.
[2] Drawing influences from alternative rock , trip hop , hip hop , electronica , dub , reggae and pop music , the band released their self-titled debut album in 2001 to worldwide success .
[3] Raised in Leytonstone , East London and around Colchester , Essex , Albarn attended the Stanway School , where he met Graham Coxon.
[4] Damon Albarn , born on 23 March 1968 , is an English musician , singer , songwriter, multi-instrumentalist and record producer .
[5] Subsequent albums such as Blur, Think and The Magic contained influences from lo-fi ,electronic and hip hop music .
Standard evidence:
[1] His debut solo studio album Everyday Robots -- co-produced by XL Recordings CEO
Richard Russell -- was released on 28 April 2014
Label : REFUTES

Fig. 1. An example of FEVER. Given a claim, the system is supposed to retrieve evidence sentences from the entire Wikipedia and label it as "SUPPORTS", "REFUTES" or "NOT ENOUGH INFO"

Most of the previous systems [6, 14, 3] use all the five sentences retrieved from the former step to do the claim verification subtask. However, 87.8% of the claims in the dataset can be verified by only one sentence according to oracle evidences [3]. Obviously, using all five evidences is not a good method, so we would like to use evidence distilling to control the number of evidences and to improve the accurancy of claim verification.

In this paper, we present a system consisting of four stages that conduct document retrieval, sentence selection, evidence sufficiency judgement and claim verification. In the document retrieval phase, we use entity linking to find candidate entities in the claim and select documents from the entire Wikipedia corpus by keyword matching. In the sentence selection phase, we use modified ESIM[2] model to select evidential sentences by conducting semantic matching between each sentence from the retrieved pages in the former step and the claim and to reserve the top-5 sentences as candidate evidences. In the evidence sufficiency judgement phase, we judge whether the evidence set is sufficient enough to verify the claim so that we can control the number of evidences for each claim dynamically. Finally, we train two claim verification models, one on the full five retrieved evidences, and the other on manually annotated golden evidence and do weighted average over them to infer whether the claim is supported, refuted or can not be decidede due to the lack of evidences.

Our main contributions are as follows. We propose a evidence distilling method for fact verification and extraction. And we construct a model to realize evidence distilling on the FEVER shared task and achieved the state-of-the-art

[3] the evidences provided in the FEVER dataset

performance on the evidence F1 score and comparable performance on claim verification.

2 Our Model

In this section, we will introduce our model in details. Our model aims to extract possible evidences for a given claim in 5 million most-accessed Wikipedia pages and judge whether these evidences support or refute the claim, or state that these evidence are not enough to decide the correctness. We first retrieve documents corresponding to the claim from all Wikipedia pages, and then select most relevant sentences as candidate evidences from these documents. After judging the sufficiency of evidences, we can distill the evidence set. Finally, we judge if the evidence set can support, refute, or not be found for the claim and label the claim as "SUPPORTS", "REFUTES", or "NOT ENOUGH INFO".

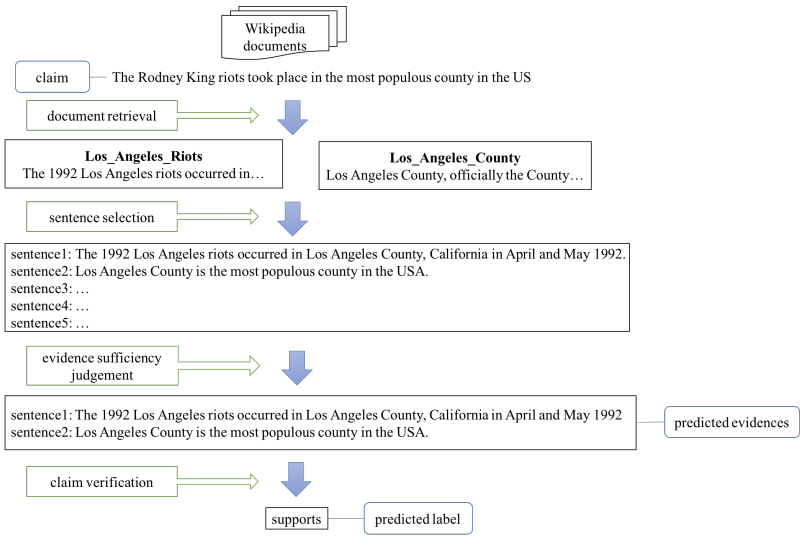

Fig. 2. Our system overview: document retrieval, sentence selection, evidence sufficiency judgement and claim verification

Formally, given a set of Wikipedia documents D=$\{d_1, d_2, d_3, \ldots, d_m\}$, each document d_i is also an array of sentences, namely $d_i = \{s_1^i, s_2^i, s_3^i \ldots s_n^i\}$ with each s_j^i denoting the j-th sentence in the i-th document and a claim c_i, the model is supposed to give a predicition tuple $(\hat{E}_i, \hat{y}_i)$ satisfying the $\hat{E}_i = \{s^{e_0}, s^{e_0}, \ldots\} \subset \cup d_i$, representing the set of evidences for the given claim, and $\hat{y}_i \in$\{ SUPPORTS, REFUTES, NOT ENOUGH INFO\}. As illustrated in Fig.2, our model contains four parts: document retrieval, sentence selection, evidence sufficiency judgement and claim verification.

2.1 Document Retrieval and Sentence Selection

Document retrieval is the selection of Wikipedia documents related to the given claim. This phase handles the task as the following function:

$$f(c_i, D) = D_{c_i} \tag{1}$$

c_i is the given claim and D is the collection of Wikipedia documents. $\hat{D_{c_i}}$ is a subset of D that consists of retrieved documents relevant to the given claim.

In this step, we first extract candidate entities from the claim and then retrieve the documents by the MediaWiki API [4] with these entities. The retrieved articles whose titles are longer than the entity mentioned and with no other overlap with the claim except for the entity will be discarded.

In the sentence selection phase, we rank all sentences in the documents we selected previously and select the most relevant sentences. In other words, our task in this phase is to choose candidate evidences for the given claim and we only consider the correlation between each single sentence and the claim without combining evidence sentences. This module handles the task as the following function:

$$g(c_i, D_{c_i}) = E_{c_i} \tag{2}$$

which takes a claim and a set of documents as inputs and outputs a subset of sentences from all sentences in the documents of D_{c_i}. This problem is treated as semantic matching between each sentence and the claim c_i to select the most possible candidate evidence set. And $E(c_i) = \{e_1, e_2, e_3, e_4, e_5\}$ represents the candidate evidence set selected.

As the sentence selection phase, we adopt the same method as the Hanselowski et al. (2018) [3]. To get a relevant score, the last hidden state of ESIM [2]is fed into a hidden layer connected to a single neuron. After getting the score, we rank all sentences and select the top five sentences as candidate evidences because each claim in FEVER has at most five evidences.

2.2 Evidence Sufficiency Judgement

We find 87.8% claims have only one sentence as evidence while in previous work, sentences selected by sentence selection are all treated as evidences. However, there may be several non-evidential sentences that could interfere with our verification for the claim. For example in Fig.1, for the claim "Damon Albarn's debut album was released in 2011.", the first sentence we selected from the sentence selection model has already covered the standard evidence set and the other four sentences can not help to verify the claim.

To alleviate this problem, We incorporate an evidence sufficiency judge model to control the number of evidences. Because the candidate evidence sentences have been sorted according to their relevance to the claim in the sentence selection phase, we first judge whether the first sentence is enough to classify the

[4] https://www.mediawiki.org/wiki/API: Mainpage

claim, if not, we would add the next sentence until the sentences are enough. And for the "NOT ENOUGH INFO" claims, because we have not enough information to verify, we keep all five candidate sentences . Consequently, we can control the number of evidences for each claim dynamically formalized as the following function:

$$h(c_i, E'_{c_i}, y_i) = l_{c_i} \tag{3}$$

$E'_{c_i}$is a subset of $E(c_i)$, E'_{c_i}can be $\{e_1\}, \{e_1, e_2\}$,$\{e_1, e_2, e_3\}$, $\{e_1, e_2, e_3, e_4\}$ or $\{e_1, e_2, e_3, e_4, e_5\}$, $l_{c_i} \in \{0, 1\}$ indicates that whether $E'_{c_i}$is enough to judge c_i in which 0 indicates not enough and 1 indicates enough. We regard it as a classification problem and construct an evidence sufficiency judge model as illustrated in Fig.3 to solve it. First, we concatenate all the evidence subsets. Then we put the concatenated evidences E and the claim C into a bidirectional LSTM layer respectively and get the encoded vectors $\hat{E}$ and $\hat{C}$.

$$\hat{E} = BiLSTM(E), \quad \hat{C} = BiLSTM(C) \tag{4}$$

Then, a bidirectional attention mechanism is adopted. After computing the alignment matrix of $\hat{E}$ and $\hat{C}$ as A, we can get aligned representation of E from $\hat{C}$ as $\widetilde{E}$ and same on C as $\widetilde{C}$ with softmax over the rows and columns.

$$A = \hat{C}^\top \hat{E} \tag{5}$$

$$\widetilde{E} = \hat{C} \cdot softmax_{col}(A^\top), \quad \widetilde{C} = \hat{E} \cdot softmax_{col}(A) \tag{6}$$

We then integrate $\hat{E}$ and $\widetilde{E}$ as well as $\hat{C}$ and $\widetilde{C}$ by the following method as EE and EC respectively.

$$EE = [\hat{E}; \widetilde{E}; \hat{E} - \widetilde{E}; \hat{E} \circ \widetilde{E}] \tag{7}$$

$$EC = [\hat{C}; \widetilde{C}; \hat{C} - \widetilde{C}; \hat{C} \circ \widetilde{C}] \tag{8}$$

Then EE and EC are put in two bidirectional LSTM respectively and after that we do max pooling and average pooling on $\hat{EE}$ and $\hat{EC}$.

$$\hat{EE} = BiLSTM(EE), \quad \hat{EC} = BiLSTM(EC) \tag{9}$$

$$e_{max} = MaxPool_{row}(\hat{EE}), \quad e_{ave} = AvePool_{row}(\hat{EE}) \tag{10}$$

$$c_{max} = MaxPool_{row}(\hat{EC}), \quad c_{ave} = AvePool_{row}(\hat{EC}) \tag{11}$$

The pooled vectors are then concatenated and put in an multi-layer percetron and the label l is produced finally.

$$MLP([e_{max}; e_{ave}; c_{max}; c_{ave}]) = l \tag{12}$$

And if the label is 1, we regard the current evidence set as the final evidence set. For example, h(c_i,$\{e_1, e_2\}$)=1, the evidence set for c_i is $\{e_1, e_2\}$ rather than $\{e_1, e_2, e_3, e_4, e_5\}$. In this way, we can control the number of evidences.

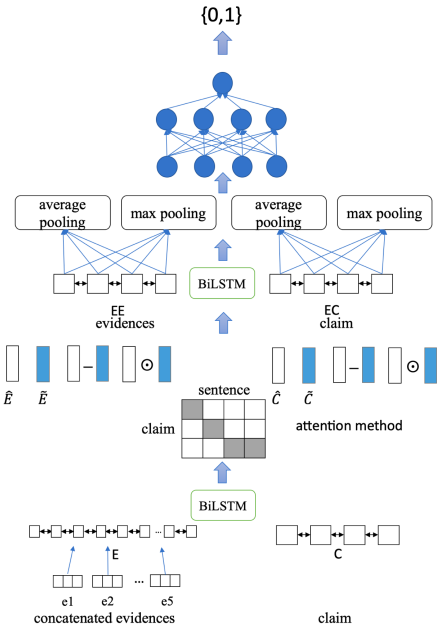

Fig. 3. The model structure for evidence sufficiency judgement phase.

2.3 Claim Verification

In this phase, we use the final evidence set selected in the evidence sufficiency judgement sub-module to classify the claim as SUPPORTS, REFUTES or NOT ENOUGH INFO. This task is defined as follows:

$$h(c_i, \hat{E_{c_i}}) = y_{c_i} \tag{13}$$

where $\hat{E_{c_i}}$ is the evidences selected by last phase for c_i and $y_{c_i} \in \{S, R, NEI\}$.

Our model in this section is modified on the basis of ESIM. The major difference is that we add a self-attention layer while the original model only use coattention. This model takes a concatenated evidence sentence and the given claim as input and outputs the label of the claim. Firstly, We compute the coattention between the concatenated evidence and the claim which is a codependent encoding of them. And then it is summarized via self-attention to produce a fine-grain representation.

We trained two claim verification models in total, one on the full data from sentence selection part with all five retrieved evidences called five-sentence model, the other on the evidence we manually annotated by gold evidences contained in the retrieved evidence set called judged-sentence model. Then we put all five of the evidences and the evidences from the evidence sufficiency judgement in the two models respectively and get the output of the two models.

Finally, we do weighted average on the two outputs to get the final label of the claim.

3 Experiment & Analysis

3.1 Dataset and Evaluation

We evaluate our model on FEVER dataset which consists of 185445 claims and 5416537 Wikipedia documents. Given a Wikipedia document set, we need to verify an arbitrary claim and extract potential evidence or state that the claim is non-verifiable. For a given claim, the system should predict its label and produce an evidence set $\hat{E_{c_i}}$, satisfying $\hat{E_{c_i}} \subseteq E_i$, where E_i is the standard evidence set provided by the dataset. For more information about the dataset please refer to Thorne et al. (2018)[10].

Besides the main track on FEVER, we construct a auxiliary dataset to help training a evidence sufficiency judge model. Specifically, for each claim-evidence pair $< c_i, E_i >$ in fever, a series of triples in the form of $< c_i, E_i', l_i >$ are constructed in our auxiliary dataset, where E_i' is a continuous subset of the whole potential evidence set E_i, and l_i is a handcrafted indicator indicates whether the subset is enough for claim verification. Considered that the evidence in E_i' is ordered by the confidence given by the sentence selection module, the continuous subset E_i' can also be seen as top m potiential evidenves in E_i. For example, E_i=$< s_i^1, s_i^2, s_i^4 >$, we can construct four triples as following: $< c_i, [s_i^1], 0 >$, $< ci, [s_i^1, s_i^2], 0 >$,$< ci, [s_i^1, s_i^2, s_i^3], 0 >$,$< ci, [s_i^1, s_i^2, s_i^3, s_i^4], 1 >$. Especially, for "NOT ENOUGH INFO" claims, we construct only one triple where E_i' contains five random sentences and l_i=0. Finally, we can get our auxiliary dataset which has 367k triples in training set and 57k in dev set. And the distribution is shown in Table. 1."evinum=i" means the first i evidences ranked by sentence selection model can cover all golden evidences. And evinum"not covered" means all five evidences can not cover golden evidences. With this dataset, our evidenve sufficiency judgement module can be trained in a supervised fasion.

Table 1. Statistics of the number of golden evidences on train and dev set respectively. "evinum=i" means that the first i evidences ranked by sentence selection model can cover all golden evidences, evinum="not covered" means that all five evidences selected by sentence selection model can not cover all golden evidences.

evinum	1	2	3	4	5	not covered
Train	85341	6381	2037	959	557	49575
Dev	9363	1210	455	255	180	8492

3.2 Baselines

we choose three models as our baselines. FEVER baseline[10] use tf-idf to select doucuments and evidences and then use MLP/SNLI to make the final prediction; UNC[6] propose a neural semantic matching network(NSMN) and use the model jointly to solve all three subtasks. They also incorporate additional information such as pageview frequency and WordNet features. And this system has the best performance in the FEVER shared task; Papelo[5] use tf-idf to select sentences and transformer network for entailment. And this system has the best f1-score of the evidence in the shared task.

3.3 Training details

In sentence selection phase, the model takes a claim and a concatenation of all evidence sentences as input and outputs a relevance score. And we hope the golden evidence set can get a high score while the plausible one gets a low score. For training, we concatenate each sentence in oracle set as positive input and concatenate five random sentences as negative input and then try to minimize the marginal loss between positive and negative samples. As word representation for both claim and sentences, we use the Glove[7] embeddings.

In evidence sufficiency judgement section, we use our auxiliary dataset to train the model. And in the claim verification section, for the five-sentence model, we use all the five sentences retrieved by our sentence selection model for training. While for the judged-evidence model, we use the golden evidences in our auxiliary dataset for training. For a given claim, we concatenate all evidence sentences as input and train our model to output the right label for the claim. We manually choose a weight (based on the performance on dev set) and use the weighted average of the two models outputs as final claim verification prediction.

3.4 Results

Overall Results In Table.2, we compare the overall performance of different methods on dev set. Our final model outperforms the Papelo which had the best evidence f1-score in the FEVER shared task by 1.8% on evidence f1-score which means our evidence distilling model has a better abillty choose evidence. Meanwhile, our label accurancy is comparable to UNC which is the best submitted system in the shared task.

Document Retriveal and Sentence Selection First, we test the performance of our model for document retrieval on the dev set. We find that for 89.94% of claims (excluding NOT ENOUGH INFO), we can find out all the documents containing standard evidences and for only 0.21% claims, we cannot find any document which consists two parts: 1) We cannot find related Wikipedia page based on the candidate entity (26 claims). 2) We cannot find the page we found in the Wikipedia online in the provided Wikipedia text source (2 claims).

Table 2. Performance of different models on FEVER.Evidence f1 is the f1 score of evidence selection where the oracle evidences are marked as correct evidences. LabelAcc is the accuracy of the predicted labels. The five-sentence model uses all five sentences selected by sentence selection model. The judged-evidence model uses evidences selected by evidence sufficiency judgement model. And the combined one is the combination of these two model. FEVER baseline is the baseline model described in [10].UNC[6] is the best submitted system during the FEVER shared task and Papelo[5] had the best f1-score of the evidence in the task.

	Evidence f1	LabelAcc
FEVER baseline[10]	18.66	48.92
UNC[6]	53.22	**67.98**
Papelo[5]	64.71	60.74
five-sentence model	35.14	65.98
judged-evidence model	**66.54**	59.47
combined	**66.54**	**67.00**

And for the other 10% claims, we can find some of the documents which contain some of the evidences but not all of them.

Then, for the sentence selection model, we extract the top 5 most similar sentences from the documents. And for 85.98% claims, the 5 sentences we selected can fully cover the oracle evidence set, and we called it fully-supported and 6.95% has at least one evidence. And hit@1 is 76.35% which means the rank-1 sentence is in the oracle evidence set.

Table 3. Performance of evidence sufficiency judge model. The first line represents the number of evidences for each claim. num_right is the number of evidence set we seleted which is exactly match with the gold evidence set on dev set

evidence_num	1	2	3	4	5
num_after_control	9367	542	166	118	9762
num_right	6429	171	65	71	6071

Evidence sufficiency Judgement Table. 3 shows the results of the evidence sufficiency judge model. Before this model, each claim has five evidences. After the dynamic control, 9367 pieces of claims has only one evidence which means our model does well in controlling the amount of evidences. And the num_right is the number of evidence set we seleted which is exactly match with the gold evidence set on dev set which we made in the same manner as we made the eivdence set for training this model.

Claim Verification As shown in Table. 4, totally, the evidence set selected by our model is exactly match with the golden evidence set for 64% data. And we do claim verfication use the judged-evidence model on this part of data and the label accurancy can reach 81.09% which means that the judged-evidence model can get a good performance when the evidence selected by evidence sufficiency judge model is right.

Table 4. Performance of judged-evidence model on the results of evidence sufficiency judge model

	completely right	not completely right
num	12807	7191
label acc	81.09%	20.84%

The results on the not completely right set is not good. This is because that the judged-evidence model has two disadvantages: first, as mentioned before, for about 14% claims we can not select all needed evidences in the sentence selection model and for these data our evidence sufficiency judge model will reserve all five sentences as evidence. But actually most data of five sentences is labeled as "NOT ENOUGH INFO". This part may produce error propagation, since in the training phase, the claim with five evidences are mostly in the label "NOT ENOUGH INFO" which will be long after the concatenation. However, in the test phase, the claim with five evidences may also be claims whose evidences are not fully found in the first two phase, causing the evidence sufficiency judgement model regard them as not sufficiency and they will have all the five evidences reserved to the claim verification phase and finally be labeled as "NOT ENOUGH INFO" which is actually wrong. Besides, for the judged-evidence model, the length of evidence ranges widely, the max length is more than 400 tokens while the min length is just about 20 tokens. The results of judged-evidence model may be influenced by the length of the input evidence. For these two problems, the five-sentence model can handle it better. So we combine these two model and get a better performance. To be more specific, after the evidence sufficiency judgement step, the judged-evidence model can regard the label "NOT ENOUGH INFO" better with more information of evidence sufficiency, while the five-sentence model are trained with more noisy evidences and can hace better performance on 14% of the claims whose oracle evidences are not be fully retrieved in the first two phase of the system. Thus, the weighted average result of the two results performs improves 7.7% of label acc. And we compare the label accurancy with different weights(the weight for judged-evidence model) for combining judged-evidence model and five-sentence model on dev set , as show in Table. 5. We find the model with weight 0.3 achieves the highest label accurancy.

Table 5. Claim verification evaluation with different weights for combining judged-evidence model and five-sentence model on dev set .

weight	0.1	0.2	0.3	0.4	0.5	0.6
label acc	66.25%	66.68%	66.98%	66.35%	64.21%	62.15%

4 Related Works

Our model focus on evidence distilling in the retrieved evidences while doing claim verification. In that circumstance, there are many works that are related to ours, and we will introduce them in this section to illustrate our model more properly.

Natural Language Inference is basically a classification task in which a pair of premise and hypothesis is supposed to be classified as entailment, contradiction or neutral which is quite same as the third step – Recognizing Textual Entailment in the FEVER Pipelined System described in (Throne et al., 2018) [10]. Recently, the emergence of Stanford Natural Language Inference(SNLI) [1]and the Multi-Genre Natural Language Inference(Multi-NLI) [13] with as much as 570,000 human-annotated pairs have enabled the use of deep neural networks and attention mechanism on NLI, and some of them have achieved fairly promising results [2, 9, 4] . However, unlike the vanilla NLI task, the third step in the FEVER Pipelined System described in (Throne et al., 2018) [10] presents rather challenging features, as the number of premises retrieved in the former steps is five instead of one in most situations. While the NLI models are mostly constructed to do one-to-one natural language inference between premise and hypothesis, there has to be a way to compose the premises or the results inferred from each of the premises with the certain hypothesis.

Fact Checking Task: After the definition of Fact Checking given by Vlachos and Riedel [11], there are many fact checking datasets apart from FEVER. Wang [12] provides a dataset for fake news detection with 12.8K manually labeled claims as well as the context and the justification for the label but not machine-readable evidence available to verify the claim. The Fake News challenge[8] provides pairs of headline and body text of News and participants are supposed to classify a given pair of a headline and a body text. However, compared with FEVER, the systems do classification by given resources rather than retrieved in the former step of the system. The FEVER shared task ,on which we did our experiments, describes a task in which we should not only verify the given claim, but also do the verification based on the evidences we retrieved ourselves in the collection of the Wikipedia text resources and provides 185,445 claims associated with manually labeled evidences.

5 Conclusions and Future Work

In this paper, we present a new four-stage fact checking framework, where we design a novel evidence sufficiency judgement model to dynamically control the

number of evidences to be considered for later verification. We show that precise control of evidence is helpful for evaluating the quality of evidence and also further claim verification. In future, we plan to improve our model by leveraging context-dependent pre-trained representations to better deal with more complex sentences. We may also try to use graph networks to incorporate inner structure among multiple evidences instead of direct concatenation.

Acknowledgment

This work is supported in part by the NSFC (Grant No.61672057,61672058,61872294), the National Hi-Tech R&D Program of China(No. 2018YFC0831900). For any correspondence, please contact Yansong Feng.

References

1. Bowman, S.R., Angeli, G., Potts, C., Manning, C.D.: A large annotated corpus for learning natural language inference. arXiv:1508.05326 (2015)
2. Chen, Q., Zhu, X., Ling, Z., Wei, S., Jiang, H., Inkpen, D.: Enhanced lstm for natural language inference. arXiv:1609.06038 (2016)
3. Hanselowski, A., Zhang, H., Li, Z., Sorokin, D., Gurevych, I.: Ukp-athene: Multi-sentence textual entailment for claim verification (2018)
4. Kim, S., Hong, J.H., Kang, I., Kwak, N.: Semantic sentence matching with densely-connected recurrent and co-attentive information. arXiv:1805.11360 (2018)
5. Malon, C.: Team papelo: Transformer networks at fever (2019)
6. Nie, Y., Chen, H., Bansal, M.: Combining fact extraction and verification with neural semantic matching networks. arXiv:1811.07039 (2018)
7. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
8. Pomerleau, D., Rao., D.: Fake news challenge. http://www.fakenewschallenge.org/, 2017
9. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving language understanding by generative pre-training. OpenAI (2018)
10. Thorne, J., Vlachos, A., Christodoulopoulos, C., Mittal, A.: Fever: a large-scale dataset for fact extraction and verification (2018)
11. Vlachos, A., Riedel, S.: Fact checking: Task definition and dataset construction. In: Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science. pp. 18–22 (2014)
12. Wang, W.Y.: " liar, liar pants on fire": A new benchmark dataset for fake news detection. arXiv:1705.00648 (2017)
13. Williams, A., Nangia, N., Bowman, S.R.: A broad-coverage challenge corpus for sentence understanding through inference. arXiv:1704.05426 (2017)
14. Yoneda, T., Mitchell, J., Welbl, J., Stenetorp, P., Riedel, S.: Ucl machine reading group: Four factor framework for fact finding (hexaf). In: Proceedings of the First Workshop on Fact Extraction and VERification (FEVER). pp. 97–102 (2018)