Integrating Entity Linking and Evidence Ranking for Fact Extraction and Verification

Motoki Taniguchi *
motoki.taniguchi@fujixerox.co.jp

Tomoki Taniguchi *
taniguchi.tomoki@fujixerox.co.jp

Takumi Takahashi
takahashi.takumi@fujixerox.co.jp

Yasuhide Miura
yasuhide.miura@fujixerox.co.jp

Tomoko Ohkuma
ohkuma.tomoko@fujixerox.co.jp

Fuji Xerox Co., Ltd.

Abstract

We describe here our system and results on the FEVER shared task. We prepared a pipeline system which composes of a document selection, a sentence retrieval, and a recognizing textual entailment (RTE) components. A simple entity linking approach with text match is used as the document selection component, this component identifies relevant documents for a given claim by using mentioned entities as clues. The sentence retrieval component selects relevant sentences as candidate evidence from the documents based on TF-IDF. Finally, the RTE component selects evidence sentences by ranking the sentences and classifies the claim simultaneously. The experimental results show that our system achieved the FEVER score of 0.4016 and outperformed the official baseline system.

1 Introduction

The increasing amounts of textual information on the Web have brought demands to develop techniques to extract and verify a fact. The Fact Extraction and VERification (FEVER) task (Thorne et al., 2018) focuses on verification of textual claims against evidence. In the FEVER shared task, a given claim is classified as SUPPORTED, REFUTED, or NOTENOUGHINFO (NEI). Evidence to justify a given claim is required for SUPPORTED or REFUTED claims. The evidence is not given and must be retrieved from Wikipedia.

This paper describes our participating system in the FEVER shared task. The architecture of our system is designed by following the official baseline system (Thorne et al., 2018). There are two main differences between our system and the baseline system. The first one is identifying documents that contain evidence by using text match between mentioned entities in a given claim and Wikipedia page title. The details are described in Section 2.1. The next one is a neural network based model, details of which are described in Section 2.3, for selecting evidence sentences as ranking task and classifying a claim simultaneously.

2 System

We propose a pipeline system which composes of a document selection, a sentence retrieval, and a recognizing textual entailment (RTE) components. A simple entity linking approach with text match is used as the document selection component. This component identifies relevant documents for a given claim by using mentioned entities as clues. The sentence retrieval component selects relevant sentences as candidate evidence from the documents based on Term Frequency-Inverse Document Frequency (TF-IDF). Finally, the RTE component selects evidence sentences by ranking the candidate sentences and classifies the claim as SUPPORTED, REFUTED, or NOTENOUGHINFO simultaneously. Details of the components are described in the following Section.

2.1 Document selection

Wikipedia pages of entities mentioned in a claim can be good candidate documents containing the SUPPORTED/REFUTED evidence. Therefore, we use a simple but efficient entity linking approach as a document selection component. In our entity linking approach, relevant documents are retrieved by using exact match between page titles of Wikipedia and words in a claim. We expect

* Authors contributed equally

Proceedings of the First Workshop on Fact Extraction and VERification (FEVER), pages 124–126
Brussels, Belgium, November 1, 2018.

this component to select only surely correct documents. In other words, we decided to prefer precision of evidence rather than recall. In fact, our preliminary experiment indicates that 68% of claims excluding NEI in a development set can be fully supported or refuted by the retrieved documents with our approach. This corresponds roughly to the accuracy of 10 nearest documents retrieved by the DrQA (Chen et al., 2017) based retrieval approach used in the baseline system. The average number of selected documents in our approach is 3.7, and thus our approach is more efficient than the baseline system.

2.2 Sentence retrieval

Following the baseline system, we use a sentence retrieval component which returns K nearest sentences for a claim using cosine similarity between unigram and bigram TF-IDF vectors. The K nearest sentences are retrieved from the documents selected by the document selection component. We selected optimal K using grid search over $\{5, 10, 15, 20, 50\}$ in terms of the performance of the full pipeline system on a development set. The optimal values was $K = 15$.

2.3 Recognizing textual entailment

As RTE component, we adopt DEISTE (Deep Explorations of Inter-Sentence interactions for Textual Entailment) model that is the state-of-the-art in RTE tasks (Yin et al., 2018). RTE component is trained on labeled claims paired with sentence-level evidence. To build the model, we utilize the NEARESTP dataset described in Thorne et al. (2018). In a case where multiple sentences are required as evidence, the texts of the sentences are concatenated. We use Adam (Kingma and Ba, 2014) as an optimizer and utilize 300 dimensional GloVe vector which is adapted by the baseline system. The other model parameters are the same as the parameters described in Yin et al. (2018).

Claims labelled as NEI are easier to predict correctly than SUPPORTED and REFUTED because unlike SUPPORTED and REFUTED, NEI dose not need evidence. Therefore, our RTE component are designed to predict the claims as NEI if the model can not predict claims as SUPPORTED or REFUTED with high confidence. RTE prediction process is composed of three steps. Firstly, we calculate the probability score of each label for pairs of a claim and candidate sentence using DEISTE model. Secondly, we decide a prediction label using the following equations.

$$SR = \arg\max_{s \in S, a \in A} P_{s,a}$$

$$P_{max} = \max_{s \in S, a \in A} P_{s,a}$$

$$Label_{pred} = \begin{cases} SR & (P_{max} > \mathrm{P_t}) \\ \mathrm{NEI} & (\text{otherwise}) \end{cases}$$

where S is a set of pairs of a claim and candidate sentence; $A = \{\textit{SUPPORTED}, \textit{REFUTED}\}$; $P_{s,a}$ is a probability score of a pair for label a; $\mathrm{P_t}$ is a threshold value; $Label_{pred}$ is prediction label for a claim. Finally, we sort candidate sentences in descending order of scores and select at most 5 evidence sentences with the same label as predicted label. We also apply grid search to find the best threshold P_t and set it to 0.93.

3 Evaluation

3.1 Dataset

We used official training dataset for training RTE component. For parameter tuning and performance evaluation, we used a development and test datasets used in (Thorne et al., 2018). Table 1 shows statistics of each dataset.

	SUPPORTED	REFUTED	NEI
Training	80,035	29,775	35,639
Development	3,333	3,333	3,333
Test	3,333	3,333	3,333

Table 1: The number of claims in each datasets.

3.2 In-house Experiment

We evaluated our system and baseline system on the test dataset with FEVER score, label accuracy, evidence precision, evidence recall and evidence F1. FEVER score is classification accuracy of claims if the correct evidence is selected. Label accuracy is classification accuracy of claims if the requirement for correct evidence is ignored. Table 2 shows the evaluation results on the test dataset. Our system achieved FEVER score of 0.4016 and outperformed the baseline system. As expected, our system produced a significant improvement of 59 points in evidence precision against the baseline system. Though evidence recall decreased, evidence F1 increased by 17 points compared to the baseline system.

Table 3 shows the confusion matrix on the development dataset. Even though our model

	FEVER Score	Label Accuracy	Evidence Precision	Evidence Recall	Evidence F1
Baseline	0.2807	0.5060	0.1084	0.4599	0.1755
Ours	0.4016	0.4851	0.6986	0.2265	0.3421

Table 2: Evaluation results on the test dataset.

Actual class ＼ Predicted class	SUPPORTED	REFUTED	NEI	Total
SUPPORTED	929	181	2223	3333
REFUTED	104	1331	1898	3333
NEI	363	300	2670	3333
Total	1396	1812	6791	9999

Table 3: Confusion matrix on the development dataset.

	FEVER Score	Label Accuracy	Evidence F1
Ours	0.3881	0.4713	0.1649

Table 4: Final results of our submissions.

tends to predict claims as NEI, the precisions of SUPPORTED (929/1396 = 0.67) and REFUTED (1331/1812 = 0.73) are higher than the precision of NEI (2670/6791 = 0.39).

3.3 Submission run

Table 4 presents the evaluation results of our submissions. The models showed similar behavior as in the in-house experiment excepting evidence F1. Our submission were ranked in 9th place.

4 Conclusion

We developed a pipeline system which composes of a document selection, a sentence retrieval, and an RTE components for the FEVER shared task. Evaluation results of in-house experiment show that our system achieved improvement of 12% in FEVER score against the baseline system.

Even though document selection component of our system has contributed to find more correct evidence document, the component was too strict, and thus degraded evidence recall. Therefore, as a future work, we plan to explore more sophisticated entity linking approach.

References

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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. Association for Computational Linguistics.

Wenpeng Yin, Dan Roth, and Hinrich Schütze. 2018. End-task oriented textual entailment via deep explorations of inter-sentence interactions. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 540–545. Association for Computational Linguistics.