BERT for Evidence Retrieval and Claim Verification

Amir Soleimani(✉), Christof Monz, and Marcel Worring

Informatics Institute, University of Amsterdam, Amsterdam, The Netherlands
{a.soleimani,c.monz,m.worring}@uva.nl

Abstract. We investigate BERT in an evidence retrieval and claim verification pipeline for the task of evidence-based claim verification. To this end, we propose to use two BERT models, one for retrieving evidence sentences supporting or rejecting claims, and another for verifying claims based on the retrieved evidence sentences. To train the BERT retrieval system, we use pointwise and pairwise loss functions and examine the effect of hard negative mining. Our system achieves a new state of the art recall of 87.1 for retrieving evidence sentences out of the FEVER dataset 50K Wikipedia pages, and scores second in the leaderboard with the FEVER score of 69.7.

Keywords: Evidence retrieval · Claim verification · BERT

1 Introduction

The constantly growing online textual information has been accompanied by an increasing spread of false claims. Therefore, there is a need for automatic claim verification and fact-checking. The Fact Extraction and VERification (FEVER) shared task [14] introduces a benchmark for evidence-based claim verification, making it possible to integrate information retrieval and natural language inference components. FEVER consists of 185 K claims labelled as ‘Supported’, ‘Refuted’ or ‘NotEnoughInfo’ (‘NEI’) based on a 50K Wikipedia pages dump. The task is to classify the claims and extract the corresponding evidence sentences (see Fig. 1). To evaluate the retrieval and verification performance together, FEVER score is defined as label accuracy conditioned on providing evidence sentence(s) unless the label is ‘NEI’.

Verifying a claim based on 50K pages is a computational challenge and can be alleviated by a multi-step pipeline. Most work [7–9,16] on FEVER has adopted a three-step pipeline. (1) Document Retrieval: a set of documents, which possibly contain relevant information to support or reject a claim, are retrieved; (2) Sentence Retrieval: five sentences are extracted out of the retrieved documents; (3) Claim Verification: the claim is verified against the retrieved sentences.

Pre-trained language models, particularly Bidirectional Encoder Representations from Transformers (BERT) [6] has significantly advanced the performance

This research was partly supported by VIVAT.

© Springer Nature Switzerland AG 2020

J. M. Jose et al. (Eds.): ECIR 2020, LNCS 12036, pp. 359–366, 2020.
https://doi.org/10.1007/978-3-030-45442-5_45

Claim: Roman Atwood is a content creator. (**Supported**)
Evidence: `[wiki/Roman_Atwood]` He is best known for his vlogs, where he posts updates about his life on a daily basis.

Claim: Furia is adapted from a short story by Anna Politkovskaya. (**Refuted**)
Evidence: `[wiki/Furia_(film)]` Furia is a 1999 French romantic drama film directed by Alexandre Aja, ..., adapted from the science fiction short story Graffiti by Julio Cortázar.

Claim: Afghanistan is the source of the Kushan dynasty. (**NotEnoughInfo**)

Fig. 1. Three examples from the FEVER dataset [14].

in a wide variety of information retrieval and natural language processing tasks including passage re-ranking [2], question answering [1,6], and question retrieval [12].

In this paper, we examine BERT for evidence-based claim verification in a three-step pipeline. A first BERT model is trained to retrieve evidence sentences. We compare pointwise cross entropy loss and pairwise Hinge loss and Ranknet loss [3] for the BERT evidence retrieval. We also investigate the effect of Hard Negative Mining (HNM), which means training on harder negative samples. Next, we train another BERT model to verify claims against the retrieved evidence. The code is available online[1].

In summary, our contributions are as follows: (1) To the best of our knowledge, we are the first to use BERT for evidence retrieval and claim verification; (2) We compare pointwise and pairwise loss functions for training the BERT sentence retrieval; (3) We investigate and employ HNM to improve the retrieval performance; (4) We achieve second rank in the FEVER leaderboard.

2 Related Work

Thorne et al. [14] shortlists the k-nearest documents based on TF-IDF features similar to DrQA [4]. UCL [16] detects the pages titles in the claims and rank pages by logistic regression. UKP-Athene [7], the highest document retrieval scoring system, uses MediaWiki API[2] to search Wikipedia for the claims noun phrases.

To extract evidence sentences, Thorne et al. [14] use a TF-IDF approach similar to their document retrieval. UCL [16] trains a logistic regression model on a heuristically set of features. Enhanced Sequential Inference Model (ESIM) [5] has been used in [7,9]. ESIM uses two BiLSTMs and the co-attention mechanism to classify a hypothesis based on a premise.

Decomposable attention [10] is used by Thorne et al. [14] for claim verification, which compares and aggregates soft-aligned words in sentences. In [8], transformer networks pre-trained on language generation [11] are employed.

[1] http://github.com/asoleimanib/BERT_FEVER.

[2] mediawiki.org/wiki/api:main_page.

ESIM has been widely used for this step [7,9,16]. UNC [9], proposes a modified ESIM that takes the concatenation of the retrieved sentences and claim along with ELMo embedding. Very recently, Dream [17] published the state of the art FEVER score using a graph reasoning module. It uses pre-trained XLNet [15] to just calculate contextual word embedding without fine-tuning. However, we show that using BERT as retrieval and verification components without any additional modules achieves comparable results.

3 Methods

FEVER provides N_D Wikipedia documents $D = \{d_i\}_{i=1}^{N_D}$. The document d_i consists of sentences $S^{d_i} = \{s_j^i\}_{j=1}^{N_{S^{d_i}}}$. The goal is to classify the claim c_l for $l = 1, \ldots, N_C = 145K$ as 'Supported', 'Refuted', or 'NEI'. For a prediction to be considered correct, a complete set of ground-truth evidence must be retrieved for the claim c_l. The 'NEI' labels do not have an evidence set. We explain the proposed system in the three-step pipeline: document retrieval, sentence retrieval, and claim verification. Figure 2 demonstrates the proposed BERT architectures for the second and third steps.

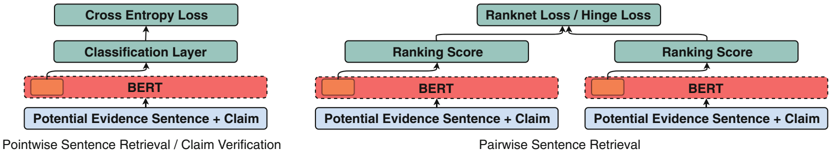
Cross Entropy Loss
Classification Layer
BERT
Potential Evidence Sentence + Claim
Pointwise Sentence Retrieval / Claim Verification
Ranknet Loss / Hinge Loss
Ranking Score
Ranking Score
BERT
BERT
Potential Evidence Sentence + Claim
Potential Evidence Sentence + Claim
Pairwise Sentence Retrieval

Fig. 2. Pointwise sentence retrieval and claim verification (left), Pairwise sentence retrieval (right). Orange boxes indicate the last hidden state of the [CLS] token. (Color figure online)

Following the UKP-Athene promising document retrieval component [7] (MediaWiki API), which results in more than 93% document recall, we use their method to collect a set of top documents $D_{top}^{c_l}$ for the claim c_l. We use all the retrieved documents as $D_{top}^{c_l}$.

The sentence retrieval step extracts the top five evidence sentences $S_{top}^{c_l}$. The training set consists of claims and the sentences from $D_{top}^{c_l}$ corresponding to c_l ($S_{all}^{c_l} = \{S^{d_i} | d_i \in D_{top}^{c_l}\}$).

BERT is a multi-layer transformer pre-trained on next sentence prediction and masked word prediction using extremely large datasets. BERT takes the input with a special classification embedding ([CLS]) followed by the tokens representations of the first and second sentences separated by another specific token ([SEP]). To use BERT for classification, a softmax is added on the last hidden state of the classification token ([CLS]) and trained together with the pre-trained layers.

We adopt the pre-trained BERT model and fine-tune using two different pointwise and pairwise approaches. By default, we use the BERT base (12 layers) in all the experiments. In order to compensate for the missed co-reference pronouns in the sentences [8], we add the page titles at the beginning of sentences. We use a batch size of 32, a learning rate of $2e-5$, and one epoch of training.

In the pointwise approach, we use cross entropy loss, and every single input is classified as evidence or non-evidence. At testing time, $S_{top}^{c_l}$ sentences are selected by their probability values. Alternatively, a threshold can also be used on the values to filter out uncertain results and trade-off the recall against the precision.

In the pairwise approach, a pair of positive and negative samples are compared against each other (Fig. 2 (right)) using the Ranknet loss function [3]. We do not force the positive and negative samples to be selected from the same claims because the number of sentences per claim is significantly different and this results in oversampling sentences from the claims with limited sentences. In addition, we experiment with the modified Hinge loss functions like [7]. At testing time, for both pairwise loss functions, the top five sentences $S_{top}^{c_l}$ are selected based on their output probability values.

The ratio of negative to positive sentences is high, thus it is not reasonable to train on all the negative samples. Random sampling limits the number of negative samples, however, this leads to training on trivial samples. Similar to [13], we focus on online HNM. We fix the positive samples batch size of 16 but heuristically increase negative batch from 16 to 64 and train on the positive samples and only the 16 negative samples with the highest loss values. In the case of pairwise retrieval, HNM selects the 32 hardest pairs out of 128 pairs. Loss values are computed in the no-gradient mode, and thus there is no need for more GPUs than normal training without HNM.

Table 1. Development set sentence retrieval performance. For * we calculated the scores using the official code, and for ** we used the F1 formula to calculate the score.

Model	Precision (%)	Recall@5 (%)	F1 (%)
UNC [9]	36.39	86.79	51.38
UCL [16]	22.74**	84.54	35.84
UKP-Athene [7]	23.67*	85.81*	37.11*
Pointwise	25.14	88.25	39.13
Pointwise+Threshold	**38.18**	88.00	**53.25**
Pointwise+HNM	25.13	88.29	39.13
Pairwise Ranknet	24.97	88.20	38.93
Pairwise Ranknet+HNM	24.97	**88.32**	38.93
Pairwise Hinge	24.94	88.07	38.88
Pairwise Hinge+HNM	25.01	88.28	38.98

In the claim verification step, each claim c_l is compared against $S_{top}^{c_l}$ and the final claim classification label is determined by aggregating the five individual decisions. Like [8], the default label is 'NEI' unless there is any supporting evidence to predict the claim label as 'Supported'. If there is at least one rejecting evidence and no supporting fact, the label is 'Refuted'. We train a pre-trained BERT as a three-class classifier (Fig. 2(left)). We train the model on 722K evidence-claim pairs provided by the first two steps. We use the batch size of 32, the learning rate of $2e-5$, and two epochs of training.

4 Results

Table 1 compares the development set performance of different retrieval variants. It indicates that both pointwise and pairwise BERT sentence retrieval improve the recall. Note that although precision and F1 are of value, as discussed by UNC [9], recall is the most important factor because the retrieval predictions are the samples that the verification system is trained on, and low recall leaves many claims with no probable evidence. Additionally, recall is weighted by the FEVER score requiring evidence for 'Supported' and 'Refuted' claims. A threshold can also regulate the recall and precision and shows that our method can achieve the best precision and F1 too. We opt to focus more on recall and train the claim verification model on the predictions with the maximum recall.

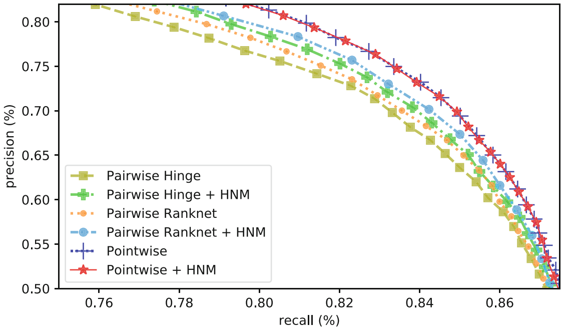

Fig. 3. Recall and precision results on the development set.

Although the pairwise Ranknet with HNM marginally has the best recall, we cannot conclude that it is necessarily better for this task. This is more clear by a trade-off between the precision and recall displayed in Fig. 3. The pointwise methods surpass the pairwise methods in terms of recall-precision performance. It also shows that HNM enhances both Ranknet and Hinge pairwise and preserves the pointwise performance.

Table 2 compares the development set results of the previous methods with the BERT models. The BERT claim verification even if it is trained on the

Table 2. Development set verification scores.

Model	FEVER score (%)	Label Acc. (%)
UNC [9]	66.14	69.60
UCL [16]	65.41	69.66
UKP-Athene [7]	64.74	–
BERT & UKP-Athene	69.79	71.70
BERT Large & UKP-Athene	70.64	72.72
BERT & BERT (Pointwise)	71.38	73.51
BERT & BERT (Pointwise+HNM)	71.33	73.54
BERT (Large) & BERT (Pointwise)	**72.42**	74.58
BERT (Large) & BERT (Pointwise+HNM)	**72.42**	**74.59**
BERT & BERT (Pairwise Ranknet)	71.02	73.22
BERT & BERT (Pairwise Ranknet+HNM)	70.99	73.02
BERT & BERT (Pairwise Hinge)	71.60	72.74
BERT & BERT (Pairwise Hinge+HNM)	70.70	72.76

Table 3. Results on the test set (October 2019).

Model	FEVER score (%)	Label Acc. (%)
DREAM [17]	**70.60**	**76.85**
BERT (Large) & BERT (Pointwise+HNM)	69.66	71.86
abcd_zh (unpublished)	69.40	72.81
BERT (Large) & BERT (Pointwise)	69.35	71.48
cunlp (unpublished)	68.80	72.47
BERT & BERT (Pointwise)	68.50	70.67
BERT (Large) & UKP-Athene	68.36	70.41
BERT & FEVER UKP-Athene	67.49	69.40
UNC [9]	64.21	68.21
UCL [16]	62.52	67.62
UKP-Athene [7]	61.58	65.46

UKP-Athene sentence retrieval predictions, the previous method with the highest recall, improves both label accuracy and FEVER score. Training based on the BERT retrieval predictions significantly enhances the verification because while it improves the FEVER score by providing more correct evidence, it provides a better training set for the verification system. It also shows that pointwise retrieval leads to more accurate claim verification. The large BERTs are only trained on the best retrieval systems, and as expected significantly improve the performance. Finally, we report the blind test set results in Table 3 using the FEVER leaderboard[3]. Our best model ranks at the second place that indicates the importance of using pre-trained language models for both sentence retrieval and claim verification.

[3] https://competitions.codalab.org/competitions/18814#results.

5 Conclusion

We demonstrated the BERT promising performance for the sentence retrieval and claim verification pipeline. In the retrieval step, we compared the pointwise and pairwise approaches and concluded that although the pairwise Ranknet approach achieved the highest recall, pairwise approaches are not necessarily superior to the pointwise approach particularly if precision is taken into account. Our results showed that training BERT on the pointwise retrieved sentences results in a better performance. We also examined HNM for training the retrieval systems and showed that it improves the retrieval and verification performance. We suspect that HNM can also make the training faster and leave the investigation for future work. Furthermore, using BERT as an end-to-end framework for the entire evidence-based claim verification pipeline can be investigated in the future.

References

1. Alberti, C., Lee, K., Collins, M.: A BERT baseline for the natural questions. arXiv preprint arXiv:1901.08634 (2019)
2. Bajaj, P., et al.: MS MARCO: a human generated machine reading comprehension dataset. arXiv preprint arXiv:1611.09268 (2016)
3. Burges, C., et al.: Learning to rank using gradient descent. In: Proceedings of the 22nd International Conference on Machine learning, ICML 2005, pp. 89–96 (2005)
4. Chen, D., Fisch, A., Weston, J., Bordes, A.: Reading Wikipedia to answer open-domain questions. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1870–1879 (2017)
5. Chen, Q., Zhu, X., Ling, Z., Wei, S., Jiang, H.: Enhancing and combining sequential and tree LSTM for natural language inference. arXiv preprint arXiv:1609.06038 (2016)
6. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
7. Hanselowski, A., et al.: UKP-Athene: multi-sentence textual entailment for claim verification. In: Proceedings of the First Workshop on Fact Extraction and VERification (FEVER), pp. 103–108. Association for Computational Linguistics, Brussels, November 2018. https://doi.org/10.18653/v1/W18-5516
8. Malon, C.: Team Papelo: transformer networks at FEVER. In: Proceedings of the First Workshop on Fact Extraction and VERification (FEVER), pp. 109–113. Association for Computational Linguistics, Brussels, November 2018. https://doi.org/10.18653/v1/W18-5517
9. Nie, Y., Chen, H., Bansal, M.: Combining fact extraction and verification with neural semantic matching networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, no. 01, pp. 6859–6866 (2019). https://doi.org/10.1609/aaai.v33i01.33016859
10. Parikh, A.P., Täckström, O., Das, D., Uszkoreit, J.: A decomposable attention model for natural language inference. arXiv preprint arXiv:1606.01933 (2016)
11. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving language understanding by generative pre-training. Technical report, OpenAI (2018)

12. Sakata, W., Shibata, T., Tanaka, R., Kurohashi, S.: FAQ retrieval using query-question similarity and BERT-based query-answer relevance. In: SIGIR 2019: 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1113–1116 (2019)
13. Schroff, F., Kalenichenko, D., Philbin, J.: FaceNet: a unified embedding for face recognition and clustering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 815–823 (2015)
14. Thorne, J., Vlachos, A., Christodoulopoulos, C., Mittal, A.: FEVER: a large-scale dataset for fact extraction and verification. arXiv preprint arXiv:1803.05355 (2018)
15. Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., Le, Q.V.: XLNet: generalized autoregressive pretraining for language understanding. arXiv preprint arXiv:1906.08237 (2019)
16. 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. Association for Computational Linguistics, Brussels, November 2018. https://doi.org/10.18653/v1/W18-5515
17. Zhong, W., et al.: Reasoning over semantic-level graph for fact checking. arXiv preprint arXiv:1909.03745 (2019)