Team DOMLIN: Exploiting Evidence Enhancement for the FEVER Shared Task

Dominik Stammbach
DFKI, Saarbrücken, Germany
dominik.stammbach@dfki.de

Günter Neumann
DFKI, Saarbrücken, Germany
neumann@dfki.de

Abstract

This paper contains our system description for the second Fact Extraction and VERification (FEVER) challenge. We propose a two-staged sentence selection strategy to account for examples in the dataset where evidence is not only conditioned on the claim, but also on previously retrieved evidence. We use a publicly available document retrieval module and have fine-tuned BERT checkpoints for sentence selection and as the entailment classifier. We report a FEVER score of 68.46% on the blind test set.

1 Introduction

The nowadays vast amounts of textual information, its ease of sharing and its error pronesses call for automatic means of fact checking (Thorne et al., 2018a). Automated Fact checking is the assignment of a truth value to a given (factual) statement, also referred to as a claim. Such an assignment by itself lacks interpretability, thus it is desirable to have access to the evidence used to reach an assignment (Vlachos and Riedel, 2014). This has led to the Fact Extraction and VERification (FEVER) challenge, i.e. the task is to classify a claim into 'SUPPORTS', 'REFUTES' or 'NOT ENOUGH INFORMATION' and to also retrieve the relevant evidence sentences from Wikipedia (Thorne et al., 2018a). An example claim is *'Cary Elwes was born in 1982.'* and we have to retrieve the evidence sentence *'Cary Elwes, born 26 October 1962, is an English actor and writer.'* from the Wikipedia page about Cary Elwes. Because the evidence contradicts the claim, the claim is refuted. In this paper, we present our system descriptionfor the *builder* phase of the second Version of this challenge (FEVER 2.0).

The *builder* phase in FEVER 2.0 is equivalent to the first FEVER shared task and participants try to beat the top performing systems of the first FEVER challenge which act as a baseline, i.e. beat 64.21% FEVER score (Thorne et al., 2018c). Some of the systems from the first FEVER challenge are publicly available and can be used by participants in FEVER 2.0[1].

In a preliminary experiment, we have fine-tuned a BERT checkpoint (Devlin et al., 2018) as the textual entailment classifier and have achieved 92.8% label accuracy on the supported/refuted examples of the development set using oracle evidence. Thus, we have focused on the evidence retrieval part of the challenge.

In our hand-in, we have used the document retrieval module developed by UKP-ATHENE in the first fever challenge (Hanselowski et al., 2018). We have built on the 'two-hop' evidence enhancement strategy proposed in (Nie et al., 2018) and propose a two-staged sentence selection strategy. We used BERT for sentence selection and for recognizing textual entailment[2] (RTE) between a claim and retrieved evidence for that claim.

2 Related Work

Most work on the FEVER dataset is based on the baseline system proposed in the dataset description (Thorne et al., 2018a), using a pipeline consisting of document retrieval, sentence selection and RTE. We implemented such a pipeline as well and have built on several ideas found in the first FEVER challenge.

We have used the document retrieval module developed by (Hanselowski et al., 2018) which achieved the highest evidence recall in the first fever challenge (Thorne et al., 2018c). They use the MediaWiki API[3] which queries the Wikipedia

[1] http://fever.ai/resources.html

[2] https://en.wikipedia.org/wiki/Textual_entailment

[3] https://www.mediawiki.org/wiki/API:Main_page

Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER), pages 105–109
Hong Kong, November 3, 2019. ©2019 Association for Computational Linguistics

search engine. Every *noun phrase* is considered to be a possible entity mention and is fed into the MediWiki API, yielding up to seven Wikipedia pages per claim.

Nie et al. (2018) propose a 'two-hop' evidence enhancement process, that is they gather all hyperlinks in their already retrieved evidence sentences and apply their sentence selection module on all sentences found in these documents retrieved by following the hyperlinks. A 0.8% increase in FEVER score (using oracle lables) is reported by using this strategy.

Malon (2018) use the open-GPT model (Radford et al., 2018) for sentence selection and entailment classification. We have trained similar models, but used BERT instead. BERT is a noisy autoencoder pre-trained on masked language modeling tasks and was the state of the art on a number of natural language understanding (NLU) tasks (Devlin et al., 2018) during the *builder* phase of FEVER 2.0, e.g. the NLU benchmark GLUE (Wang et al., 2018) and on SQuAD (Rajpurkar et al., 2016), a question answering dataset. Classification in BERT is achieved by training a special '[CLS]' token which is prepended to every sequence (or sequence pair), gather the '[CLS]' token's hidden representation and perform classification on top of that. We used the cased English version of $\text{BERT}_{\text{BASE}}$ for all our experiments.

Hanselowski et al. (2018) use the hinge loss function[4] to maximize the margin between positive and (sampled) negative evidence sentences. Thus, we adapted BERT for sentence selection to be trained with the hinge loss as well.

3 Our Model

We have submitted a pipeline appraoch consisting of document retrieval, a two-staged sentence selection strategy followed by an RTE module. In this section, we describe the different modules of our pipeline in more detail.

3.1 Document Retrieval

We have re-used the document retrieval developed by (Hanselowski et al., 2018). We have experimented with using the union of the retrieved documents of the three best performing systems in the first fever challenge, but found that document recall of using such an ensemble only slightly increases while precision drops massively (Table 1).

In Table 1, we report precision, recall and F1 for the relevant documents retrieved by the union of different retrieval modules in the development set.

System	Pr (%)	Rc (%)	F1 (%)
Athene UKP TU Darmstadt	28.3	78.6	41.6
Athene + UCL Machine Reading Group	7.8	80.1	14.0
Athene + UCL + UNC-NLP	6.2	80.2	11.0

Table 1: Results for different Document Retrieval strategies

Because of the only slight increase in recall, but the big drop in precision and the increase in computation, we restricted ourselves to only use the document retrieval system developed by (Hanselowski et al., 2018).

3.2 Sentence Selection

In 16.82% of cases in the FEVER dataset, a claim requires the combination of more than one sentence to be able to support or refute that claim (Thorne et al., 2018c). While inspecting such cases, we have found that sometimes, evidence is not only conditioned on the claim, but also on already retrieved evidence. Two examples of such cases can be found in Table 2.

Claim	Evidence 1	Evidence 2
Ryan Gosling has been to a country in Africa.	He [...] has traveled to Chad , Uganda and eastern Congo [...].	Chad [...] is a landlocked country in Central Africa
Stanley Tucci performed in an television series.	He won two Emmy Awards for his performances in Winchell and Monk	Monk is an American comedy-drama detective mystery television series created by Andy Breckman and starring Tony Shalhoub as the eponymous character, Adrian Monk.

Table 2: Examples where evidence sentences are not only conditioned on the claim

Thus, we propose a two-staged sentence selection process building on top of the 'two-hop' evidence enhancement process in (Nie et al., 2018). We believe that the relevant document for the second evidence (in Table 2) can only be retrieved by gathering the hyperlinks in *Evidence 1*, and adopt that 'two-hop' strategy. Because *Evidence 2* is not only conditioned on the claim, but also the first evidence sentence, we find it impossible (as humans) to correctly classify the second evidence without having information about the first evidence. Thus, we want to model this fact accordingly and describe our sentence selection strategy in the following.

We fine-tune two different BERT checkpoints with different training examples. For the first model, we select only the first sentence in every

[4] https://en.wikipedia.org/wiki/Hinge_loss

evidence set as a positive example. This covers the 83.18% of cases where the example only requires one evidence sentence. If an evidence set consists of more than one sentence, we only use the first one and ignore the other evidence sentences. Negative examples are sampled from the same document a positive example appears in (as long as it is not contained in the evidence set of an example) and from non-relevant documents returned by the document retrieval module. Following (Malon, 2018), we add the page title for co-reference resolution to the evidence sentence. An input example consists of *”[CLS]” + claim + ”[SEP]” + page_title + ”:” + evidence_sentence + ”[SEP]”*. We assign BERT segment embeddings *A* to the claim and segment embeddings *B* to the page title and the evidence sentence. Following (Hanselowski et al., 2018), we use the hinge loss function for sentence selection to maximize the margin between positive and negative examples.

We fine-tune a second BERT checkpoint (using hinge loss as well) to account for the examples in Table 2. We consider as positive examples all instances in the training set where the evidence set consists of exactly two sentences. Negative examples are sampled from hyperlinked documents in the first evidence sentence and from the same document as the second evidence, as long as a sampled sentence does not appear in any evidence set of the claim. Input to the model consists of *”[CLS]” + claim + page_title_1 + evidence_1” + [SEP]” + page_title_2 + ”:” + evidence + ”[SEP]”*. BERT segment embeddings *A* are assigned to the claim and the first evidence sentence, segment embeddings *B* are assigned to the second sentence.

During test time, we let the first model classify all sentences in all retrieved documents for a given claim. If a sentence receives a score bigger than 0, we apply the ‘two-hop’ strategy, i.e. we retrieve all hyperlinks in the document this sentence occurs in. We then collect all sentences in the documents found via hyperlinks and let the second model predict all these additionally retrieved sentences conditioned on the claim and the previously retrieved first evidence sentence. Finally, we rank all sentences with respect to their score and return the five highest scoring sentences as evidence for a claim.

We report results for the two-staged sentence selection process on the development set in Table 3 (assuming oracle labels for the FEVER score).

Model	Pr (%)	Rc (%)	F1 (%)	FEVER score (%)
First sentence selection module	24.9	87.4	38.7	91.6
Both retrieval modules	25.1	89.8	39.3	**93.2**

Table 3: Results for the two-staged Sentence Selection Module

We observe an increase of 1.6% in FEVER score (assuming oracle labels) by using the proposed two-staged sentence selection approach, twice as high as the 0.8% increase for evidence enhancement reported in (Nie et al., 2018), supporting the assumption that cases shown in Table 2 should be modelled accordingly. More importantly, we believe this strategy enables us, in theory, to retrieve most of the relevant evidence in the FEVER dataset. We think this was not possible before with the different sentence selection modules used in the first FEVER challenges.

3.3 Claim Verification

The last part of our pipeline is the claim verification (RTE) module. We adopt two strategies used in the FEVER baseline (Thorne et al., 2018b), namely how we retrieve evidence for the ‘NOT ENOUGH INFORMATION’ (NEI) examples and how we handle multiple evidence sentences for a claim.

For ‘NEI’ examples, we let the document retrieval module predict relevant pages and use our two-staged sentence selection module to select relevant evidence for these examples.

If we have multiple evidence sentences for a claim, we prepend the Wikipedia page title to each of them (for co-reference resolution) and concatenate all the evidence sentences. We only consider sentences receiving a score > 0 by the sentence selection module, but return the five highest scoring sentences for an increased FEVER score.

In Table 4, we report results for an RTE experiment using the five best scored evidence sentences (trained with five best scored evidence sentences for ‘NEI’ examples) and for an experiment using only evidence sentences with a score greater than 0 (trained ‘NEI’ examples accordingly).

It follows from Table 4 that if we use noisy evidence in the RTE module, we get low precision/high recall for the ‘NEI’ class but low recall for the other two classes. In case we only use trustworthy evidence, we get high recall for the supports/refutes classes and low recall for the

	Noisy Evidence			Trustworthy Evidence		
Class	Pr (%)	Rc (%)	F1 (%)	Pr (%)	Rc (%)	F1 (%)
Supports	75.78	46.34	57.52	75.06	92.90	83.64
Refutes	74.76	41.48	53.35	78.27	77.42	77.84
‘NEI’	43.73	80.20	56.60	70.75	51.77	59.79
Overall Acc.	56.0%			**75.7%** (72.1%)		

Table 4: Experiments using noisy and only trustworthy evidence

‘NEI’ class. We think that in the noisy experiment, BERT has learned that if it is confronted with a long sequence (a claim and five evidence sentences), it most likely is a ‘NEI’ example, because 83.18% of the supportable/refutable examples only require one evidence sentence. During decoding, it would receive only long sequences and predicts most of them as the ‘NEI’ class. We report an overall label accuracy of 56% for that experiment.

However, if we only use trustworthy evidence, we get great scores for the supports/refutes classes but predict most of the ‘NEI’ examples as being verifiable. If we ignore the ‘NEI’ examples in evaluation, we achieve 85.3% label accuracy, getting close to the results in our preliminary experiment using oracle evidence for verifiable claims. We achieve an overall label accuracy of 75.7% for examples for which we find trustworthy evidence (17k examples). Otherwise, we classify a claim heuristically to belong to the ‘NEI’ class. Because this is not always correct, the overall label accuracy on the development set drops to 72.1%. This still clearly outperforms the 56% from the noisy experiment, hence we used this strategy in our submission.

The ‘NEI’ class seems to be the most problematic one to predict correctly. We tried to augment training examples for that class leveraging information found in the SQuAD 2.0 dataset (Rajpurkar et al., 2016). In SQuAD 2.0, a number of questions remain unanswerable given the information in the corresponding Wikipedia paragraph. We included these examples in the training set and have treated the question to be the claim and the corresponding paragraph to be the evidence. We hoped that this would give the model cues about when there is not enough information to answer a question and thus, the model would improve at handling examples from the ‘NEI’ class in a better way. However, this did not help and we report an overal label accuracy on the development set of 74.7% for examples we find evidence for and an overall label accuracy of 71.1% using our heuristic for claims for which we do not find any evidence.

Finally, We have not managed to find a suitable solution to handle the ‘NEI’ class convincingly in the *builder* phase of the shared task and leave this problem to future research.

4 Conclusion

In this paper, we described our system for the *builder* phase of FEVER 2.0. We use a publicly available document retrieval system and propose a new, two-staged sentence selection strategy. In a first stage, we classify all sentences in all retrieved documents. In a second stage, we follow all hyper-links in evidence retrieved in the first stage and use a second classifier to classify all sentences in these newly retrieved documents. We propose this strategy, because sometimes, further evidence for a claim is not only conditioned on the claim, but also on previously retrieved evidence. We think that this strategy enables us, in theory, to retrieve a large amount of the evidence in the FEVER dataset which has not been the case before.

Lastly, we use BERT as our RTE classifier and report 85.3% label accuracy for the supports/refutes classes and an overall label accuracy of 72.1% on the development set. On the blind test set, we achieve 71.5% label accuracy and an overall FEVER score of 68.46%. The most problematic class in the dataset remains the ‘NOT ENOUGH INFORMATION’ class. We tried to improve performance for that class by augmenting the training set with SQuAD data, but could not report positive results. We leave the problem of the ‘NEI’ class to future research.

5 Acknowledgements

This work was partially supported by the German Federal Ministry of Education and Research (BMBF) through the project DEEPLEE (01IW17001) and the project Precise4Q (777107).

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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 *EMNLP 2018*, EMNLP 2018, FEVER Workshop.

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. 2018. Combining fact extraction and verification with neural semantic matching networks. *CoRR*, abs/1811.07039.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018a. 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. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018b. FEVER: a large-scale dataset for fact extraction and verification. *CoRR*, abs/1803.05355.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018c. 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.

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. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.