# Evidence Selection as a Token-Level Prediction Task

**Dominik Stammbach**
Center for Law & Economics
ETH Zurich
dominik.stammbach@gess.ethz.ch

## Abstract

In Automated Claim Verification, we retrieve evidence from a knowledge base to determine the veracity of a claim. Intuitively, the retrieval of the correct evidence plays a crucial role in this process. Often, evidence selection is tackled as a pairwise sentence classification task, i.e., we train a model to predict for each sentence individually whether it is evidence for a claim. In this work, we fine-tune document level transformers to extract all evidence from a Wikipedia document at once. We show that this approach performs better than a comparable model classifying sentences individually on all relevant evidence selection metrics in FEVER. Our complete pipeline building on this evidence selection procedure produces a new state-of-the-art result on FEVER, a popular claim verification benchmark.

## 1 Introduction

Automated Claim Verification is the task of determining the veracity of a claim given evidence retrieved from a knowledge base. A popular and large-scale benchmark for this task is FEVER, consisting of almost 200K synthetically generated claims derived from the introduction sections of Wikipedia sentences (Thorne et al., 2018). Input to such a system is a claim. We then search relevant documents from Wikipedia, and select the evidence sentences from the retrieved documents. Finally, we determine the veracity of the claim given the retrieved evidence, i.e., we predict one of the labels SUPPORTS, REFUTES or NOT ENOUGH INFO.

In this paper, we propose a new angle to think about evidence selection. Unlike the sentence-level approaches taken in previous work, we extract all evidence sentences from a Wikipedia document at once. This procedure outperforms a comparable sentence-level baseline in FEVER. The code and models described in this paper are publicly available.[1]

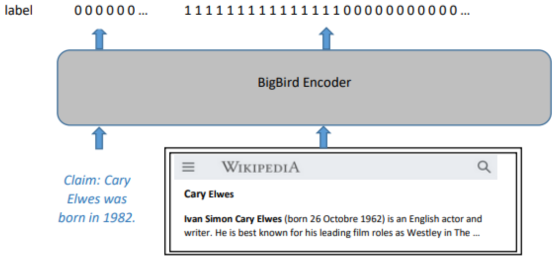
label
000000...
11111111111111100000000000...
BigBird Encoder
Claim: Cary Elwes was born in 1982.
WIKIPEDIA
Cary Elwes
Ivan Simon Cary Elwes (born 26 Octobre 1962) is an English actor and writer. He is best known for his leading film roles as Westley in The ...

Figure 1: Selecting evidence from <claim, article> pairs, while predicting a score for each token.

Consider the claim *Cary Elwes was born in 1982*. The FEVER task consists of extracting the annotated evidence from Wikipedia and to predict the right label. For this claim, we would have to retrieve the page about *Cary Elwes* from which we have to select the following evidence sentence: *Cary Elwes, born 26 October 1962, is an English actor and writer*. Because the evidence contradicts the claim, the claim is REFUTED. FEVER Score, the official metric, is a combination of returning the correct evidence and predicting the correct label.

Past work considered evidence selection to be a pairwise sentence classification task. For example, the page about *Cary Elwes* contains four sentences. A model would predict each <claim, sentence> pair individually. Our intuition is that more context helps to decide whether a given sentence is evidence – where more context is having access to the complete article. Hence, we predict a score for each token in a document and aggregate token scores on a sentence level. We show a visualization of our approach in Figure 1. Input to our system are <claim, article> pairs. We then fine-tune a transformer to predict 1 for each token belonging to annotated evidence for a claim, and to predict 0 for all other tokens. At test time, we average scores for

[1] https://github.com/dominiksinsaarland/document-level-FEVER

all tokens on a sentence level. If the resulting average is higher than 0.5, we consider the sentence to be evidence. Because Wikipedia articles are often longer than the maximum sequence length allowed in standard Transformers (e.g. Devlin et al., 2018; Liu et al., 2019), we use BigBird, which uses sparse attention to accommodate long sequences (Zaheer et al., 2020).

This perspective on evidence selection is more related to, e.g., answer span extraction in question answering (Rajpurkar et al., 2016; Kwiatkowski et al., 2019). Beltagy et al. (2020) and Zaheer et al. (2020) have both shown that long document encoders significantly improve model performance in question answering tasks with lots of context. We explore whether this holds for claim verification in FEVER as well. For example, consider the claim *St. Anger is the first studio album by Metallica.* One evidence sentence for this claim is *It was released in June 2003 as the lead single from their eighth studio album of the same name* . which can be found in the Wikipedia page *St. Anger (Song)*. Without additional context, it is not possible to resolve *their eight studio album*, and therefore it is difficult to select this sentence as evidence. However, the first sentence of this article states *[...] is a song by American heavy metal group Metallica* ., and thus resolves *their*. Having access to the whole document makes it straightforward to correctly predict this example.

To summarize: We propose a new perspective on evidence retrieval in automated claim verification.[2] We treat evidence selection as a token-level prediction task and extract all evidence from a Wikipedia article at once. This approach improves all relevant evidence retrieval metrics in FEVER compared to a RoBERTa baseline. Finally, we fine-tune a claim verification model using evidence retrieved by the proposed method which results in a new state-of-the-art result in FEVER.

## 2 Methods and Related Work

In this section, we describe our methods and we place our modeling decisions in the context of previous FEVER systems. To begin with, there exists a plethora of related work on FEVER – The most notable arguably is the FEVER baseline system itself. Thorne et al. (2018) perform (i) coarse-grained TF-IDF-based document retrieval, followed by (ii) more fine-grained evidence retrieval and eventually (iii) a model prediction of whether the extracted evidence entails the claim. Most subsequent systems broadly follow this pipeline while improving the individual building blocks with more tailor-made solutions. And so do we – in this work, we specifically provide a new perspective on evidence selection.

**Document Retrieval:** Many FEVER systems query the MediaWiki API to retrieve relevant evidence documents (introduced by Hanselowski et al., 2019). Given the importance of having retrieved the relevant candidate documents, more recent systems started to combine the MediaWiki API with TF-IDF or bm25 (Stammbach and Ash, 2020; Jiang et al., 2021).

We also take the union of documents retrieved via MediaWiki API and TF-IDF scores. These documents are the starting point of our system.

**Evidence Selection:** After having retrieved documents, the relevant sentences have to be selected from the documents. This step is usually perceived to be a pairwise sentence classification task. For each <claim, sentence> pair, a model predicts whether the sentence is evidence for the claim. Alternatively, this stage can be considered re-ranking of the evidence (Jiang et al., 2021). Most systems use Transformer-based architecture for sentence retrieval (e.g. Malon, 2018; Soleimani et al., 2020; Jiang et al., 2021).

The novelty of our work lies in the evidence selection stage. As visualized in Figure 1, we approach evidence selection as a token-level prediction task. We fine-tune BigBird to predict an evidence score for each token in a document individually and average[3] these scores on sentence-level. Hence, inputs to our system are <claim, article> pairs and output is a score for each sentence.

Document-level context has already been introduced in earlier FEVER systems, e.g. Malon (2018) prepends page titles to sentences to help resolve co-references. Additionally, document-level context improves evidence selection for scientific claim verification (Li et al., 2021) – our approach models both implicitly.

[2] In fact, the authors in (Wadden et al., 2020) also explored paragraph-level evidence selection using Longformer (Beltagy et al., 2020) for scientific claim verification. They add a [CLS] token (with global attention) to each sentence and the model predicts whether this token is evidence. Their system *LongChecker* ranks on top of the SciFact leaderboard at the time of submission of this work.

[3] The median and max lead to almost identical results.

Furthermore, our approach is computationally more efficient. Assuming BigBird is as efficient for a <claim, article> pair as RoBERTa is for a <claim, sentence> pair (which is not quite the case), we reduce training and inference time by the factor of the mean sentence length of Wikipedia articles[4].

**Multi-hop Retrieval:** Some systems follow hyperlinks present in a Wikipedia article to explore additional documents (see e.g. Nie et al., 2018; Stammbach and Neumann, 2019). We do the same and model hyperlink documents in the same way as described in (Stammbach and Neumann, 2019).

**Sparse Attention:** Vanilla Transformer-based architectures (introduced in Vaswani et al., 2017) calculate the full attention matrix $A$ which includes all pairwise interactions between tokens in a sequence. Because $A$ is quadratic in time and memory complexity, these systems only work on modestly long input, i.e., the cutoff is usually made at 512 subwords. A substantial amount of introduction sections in Wikipedia are longer, so we fall back to BigBird which implements sparse attention patterns, allowing for sequence lengths up to 4096 tokens.

BigBird only computes neighboring interactions between tokens, global interactions for a subset of tokens (by default the [CLS] and [SEP] token), and random interactions between a token and randomly sampled other tokens in the sequence. Thus, the full attention matrix is not computed anymore, which resolves the quadratic time and memory complexity. Zaheer et al. (2020) have shown that BigBird's attention pattern is equivalent in performance to e.g. BERT on short sequence lengths and outperforms BERT on all tasks requiring longer sequence lengths.

**Claim verification:** This is the last step, where most related work takes a textual entailment approach. To account for all retrieved evidence in the previous stage, the selected sentences are concatenated (see e.g. Thorne et al., 2018).

There is no annotated evidence for non-verifiable claims. Hence, early systems extracted evidence for these claims using the Sentence Selection module (Thorne et al., 2018). More recent systems create a noisy claim verification dataset by predicting evidence for all claims in the training set. The verification model is then trained on this noisy dataset (Stammbach and Ash, 2020; Jiang et al., 2021). This arguably minimizes distribution mismatch during training and inference and can lead to substantially higher label accuracy.

We take this approach as well and extract evidence for all claims in the training set. We then concatenate the five highest scoring evidence sentences for each claim and sort them in descending order according to the assigned sentence score. This constitutes our noisy training set, on which we fine-tune DeBERTa-V2-XL pre-trained on MNLI[5] (He et al., 2021).

## 3 Results

### 3.1 Sentence Selection Results

We present evidence selection results on the FEVER development set in Tables 1 and 2. We compare results of our sentence selection method to a RoBERTa checkpoint making a decision for each sentence individually.

We show evidence precision, recall, F1 and FEVER Score assuming access to oracle labels. In the first half of the tables, we show metrics for evidence scoring $> 0.5$, i.e. our model is confident that this is an evidence sentence. In Table 1, we see that BigBird achieves remarkable gains in both precision and recall, resulting in a 2% point increase in F1 compared to a RoBERTa checkpoint with the same amount of parameters. Assuming oracle claim verification labels for both selection methods, our approach still results in a 1% higher FEVER Score. BigBird-large maintains recall while gaining another 5% points in precision.

In the second half of the tables, we show results assuming re-ranking.[6] Comparing RoBERTa and BigBird in this setting, we again observe an increase of 1% in recall and FEVER Score assuming oracle labels. BigBird-large again improves recall over BigBird-base. We also present an oracle upper bound (last row) to explore what is possible given our retrieved Wikipedia pages: The oracle assigns 1 to each annotated evidence sentence and 0 to all other sentences. Surprisingly, if we measure FEVER Scores assuming oracle labels, the FEVER Score of the oracle is less than 1% point higher than our BigBird-large model.

[4] if an article contains N sentences, pairwise sentence classification requires N computation steps, whereas our approach takes only one step.

[5] larger models perform better – DeBERTa has been shown to work well on MNLI and is the largest model we could fit on a single 32GB GPU

[6] This is standard in FEVER – only recall matters and the five highest scoring evidence sentences are submitted.

| Experiment | Pr (%) | Rc (%) | F1 (%) | FEVER Score (%) |
|---|---|---|---|---|
| Roberta-base | 71.25 | 83.21 | 76.72 | 88.74 |
| Bigbird-base | 73.57 | 84.57 | 78.69 | 89.71 |
| Bigbird-large | **79.16** | 84.52 | **81.75** | 89.68 |
| Roberta-base top5 | 25.06 | 89.30 | 39.14 | 92.86 |
| Bigbird-base top5 | 24.86 | 90.33 | 38.99 | 93.55 |
| Bigbird-large top5 | 25.49 | **90.79** | 39.81 | **93.86** |
| Oracle | 98.39 | 97.58 | 92.19 | 94.81 |

Table 1: Evidence Retrieval

| Experiment | Pr (%) | Rc (%) | F1 (%) | FEVER Score (%) |
|---|---|---|---|---|
| Roberta-base | 53.17 | 86.81 | 65.95 | 91.20 |
| Bigbird-base | 56.9 | 88.15 | 69.13 | 92.1 |
| Bigbird-large | **69.98** | 88.22 | **78.05** | 92.15 |
| Roberta-base top5 | 24.51 | 91.94 | 38.7 | 94.62 |
| Bigbird-base top5 | 26.87 | 92.55 | 41.6 | 95.03 |
| Bigbird-large top5 | 26.03 | **93.62** | 40.73 | **95.74** |
| Oracle | 98.50 | 97.76 | 96.02 | 96.88 |

Table 2: Multi-hop Evidence Retrieval

In Table 2, we show analogous results for additionally incorporating retrieved multi-hop evidence sentences. Consequently, precision drops while recall and FEVER Score assuming oracle labels increase for all settings. We observe the same trends where BigBird-base outperforms an equivalent RoBERTa model in precision and recall, leading to a 3% point increase in F1 and a 0.8% point increase in FEVER Score. Again, the large model achieves similar recall than the base model, but makes remarkable gains in precision. In the re-ranking setting, BigBird (both base and large) achieve higher recall and FEVER Score than RoBERTa, with more striking differences between the two BigBird checkpoints. Lastly, if we again assume an oracle selecting evidence, this oracle would only achieve a FEVER Score 1% higher than our BigBird-large model. Thus, we conclude our approach achieves close to what is possible in evidence selection given the retrieved documents.

### 3.2 Leaderboard Results

Our final submission consists of the evidence retrieved by BigBird-large, including multi-hop evidence. We submit the five highest scoring evidence sentences for each claim. Our claim verification model is also trained on this input and assigns a veracity label to each claim. In Appendix A, we compare our submission to other high ranking systems on the leaderboard of the FEVER test set.

## 4 Conclusion

We present a novel approach to evidence retrieval. Instead of considering sentence selection as a pairwise sentence task, we extract all evidence sentences from a Wikipedia article at once by predicting for each token whether it is part of an evidence sentence or not. Our method outperforms a comparable RoBERTa checkpoint trained on <claim, sentence> pairs in every metric regarding evidence retrieval in FEVER, while being less computationally expensive. Hence, we consider our method for evidence selection to be an unambiguous improvement over previous approaches. We believe most FEVER systems (and possibly other claim verification tasks) would benefit from the method proposed in this work. We have shown that our system performs only 1% point worse than an oracle in terms of FEVER Score assuming gold veracity labels. Hence, we reason that our evidence selection method reflects strong performance given our initial set of retrieved documents. Finally, our full pipeline leads to a new state-of-the-art on the blind FEVER test set at the time of submission.

We consider our work only as the starting point in the incorporation of document-level transformers to automated claim verification. Avenues for future work could include e.g. taking a more SQuAD-like approach where only start and end tokens of evidence spans are predicted, or where we predict a special [CLS] token for each sentence, while having the whole article as input. Given re-ranked documents (such that the combined sequence length is $< 4096$), it might also be interesting to concatenate all of them and extract evidence at once from all documents for a given claim. Or predict the veracity directly from these documents, without having to extract evidence at all.

## References

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer.

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. 2019. Ukp-athene: Multi-sentence textual entailment for claim verification.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention.

Kelvin Jiang, Ronak Pradeep, and Jimmy Lin. 2021. Exploring listwise evidence reasoning with t5 for fact verification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 402–410, Online. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*.

Xiangci Li, Gully Burns, and Nanyun Peng. 2021. A paragraph-level multi-task learning model for scientific fact-verification.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

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.

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. Association for Computational Linguistics.

Amir Soleimani, Christof Monz, and Marcel Worring. 2020. Bert for evidence retrieval and claim verification. In *Advances in Information Retrieval*, pages 359–366, Cham. Springer International Publishing.

Dominik Stammbach and Elliott Ash. 2020. e-fever: Explanations and summaries forautomated fact checking. In *Proceedings of the 2020 Truth and Trust Online Conference (TTO 2020), Virtual, October 15-17, 2020*, pages 32–43. Hacks Hackers.

Dominik Stammbach and Guenter Neumann. 2019. Team DOMLIN: Exploiting evidence enhancement for the FEVER shared task. In *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*, pages 105–109, Hong Kong, China. Association for Computational Linguistics.

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, New Orleans, Louisiana. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550, Online. Association for Computational Linguistics.

Manzil Zaheer, Guru Prashanth Guruganesh, Avi Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Minh Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Mahmoud El Houssieny Ahmed. 2020. Big bird: Transformers for longer sequences.

## A Leaderboard at Time of Submission

In Table 3, we show results of the leaderboard for the blind FEVER test set at time of submission of this work. Even though our submission does not achieve the highest label accuracy, it still improves by 0.9% points the FEVER score compared to the second best entry. FEVER Score by default is upper-bounded by the retrieved evidence, and we attribute the mis-match in FEVER Score and label accuracy to our method of selecting evidence sentences. Compared to the third best entry, our system scores higher in every metric.

## B Hyperparameters

In Table 4, we list the hyper-parameters used for training the models used in this paper. We did not tune them for the Sentence Selection model, but used the same training data and hyper-parameters as in (Stammbach and Ash, 2020), from where we also have the Roberta-base model against which we compare our results in Table 1 and 2. We experimented with batchsizes 8, 16 and 32 for the RTE model, where batchsize 16 yielded best development set results.

## C Additional Training and Evaluation Tricks

**Excluding Long Wikipedia Pages:** We exclude Wikipedia articles during training and evaluation which contain more than 1500 tokens (split by whitespace). Manual inspection of the training and development set yields that these lengthy articles are mostly verbose lists and do not contain annotated evidence. Including these articles would imply that we could not train with batchsize 4 (and 8 gradient accumulation steps), but would need to train on batchsize 2 and 16 gradient accumulation steps (due to memory requirements).

**Masking:** We mask the predictions for tokens in the claim during training (and ignore these predictions during fine-tuning). We also set the label of the [CLS] token to 1 if a Wikipedia article contains at least 1 evidence sentence, else the label of the [CLS] token is 0. The [CLS] token is not masked during computation of the loss.

**Down-sampling Negative Examples** We downsample articles not containing any annotated evidence. We downsample lengthier articles more frequently, i.e. we keep 10% lengthy articles and 25% *short* articles. This speeds up training time.

**Model Input:** We add a special <EOS> token at the end of each sentence. We add the [CLS] token at the beginning of each <claim, article> pair, a [SEP] token between the claim and the article and a [SEP] token at the end of each Wikipedia article, i.e. model input for sentence selection consists of

[CLS] claim [SEP] sentence_1 [EOS] sentence_2 [EOS] ... [SEP]

Figure 2: Model Input for Sentence Selection Model

**Down-weighting Hyperlink Evidence:** We follow (Stammbach and Neumann, 2019) and retrieve hyperlink pages for each evidence sentence $e_i$ selected by our model in a first pass. We then predict for each such retrieved page whether it contains evidence, by additionally prompting $e_i$ next to the claim in the model input. We down-weight scores for examples retrieved in this manner to be $0.0001$ (barely above confidence threshold) if the score is > 0. If the score is negative, it remains unchanged. We find that our model would otherwise rank these sentences higher than *more important* evidence retrieved in the first pass.

| author | evidence F1 | LA | FEVER Score |
|---|---|---|---|
| `nudt_nlp` | 38.9 | 77.4 | 74.4 |
| (Jiang et al., 2021) | 39.6 | **79.4** | 75.9 |
| `ours` | **40.0** | 79.2 | **76.8** |

Table 3: Leaderboard Entries at Time of Submission

| parameter | Sentence Selection | RTE |
|---|---|---|
| checkpoint | bigbird-roberta-large | deberta-v2-xlarge-mnli |
| learning rate | 2e-5 | 3e-6 |
| (effective) batch size | 32 | 16 |
| number of epochs | 2 | 2 |
| max_length | 1536 | 256 |
| max_grad_norm | 1.0 | 1.0 |
| weight_decay | 0.0 | 0.0 |
| warmup_steps | 0 | 0 |
| FP16_training | True | True |
| adam_epsilon | 1e-8 | 1e-8 |
| dropout | 0.1 | 0.1 |

Table 4: Hyper-parameters Used during Training