Exploring Listwise Evidence Reasoning with T5 for Fact Verification

Kelvin Jiang, Ronak Pradeep, and **Jimmy Lin**
David R. Cheriton School of Computer Science
University of Waterloo
{kelvin.jiang, rpradeep, jimmylin}@uwaterloo.ca

Abstract

This work explores a framework for fact verification that leverages pretrained sequence-to-sequence transformer models for sentence selection and label prediction, two key sub-tasks in fact verification. Most notably, improving on previous pointwise aggregation approaches for label prediction, we take advantage of T5 using a listwise approach coupled with data augmentation. With this enhancement, we observe that our label prediction stage is more robust to noise and capable of verifying complex claims by jointly reasoning over multiple pieces of evidence. Experimental results on the FEVER task show that our system attains a FEVER score of 75.87% on the blind test set. This puts our approach atop the competitive FEVER leaderboard at the time of our work, scoring higher than the second place submission by almost two points in label accuracy and over one point in FEVER score.

1 Introduction

In recent years, the Internet has become an effective platform for creating and sharing content to large audiences. Unfortunately, there have been occurrences of bad actors taking advantage of this to propagate manipulative information for their benefit, often to the point of spreading misinformation. With the large amount of data being generated on the Internet each day, it is infeasible to manually verify it all, motivating recent research into automated fact verification.

In this work, we explore a fact verification framework built with the pretrained sequence-to-sequence transformer T5 (Raffel et al., 2020) as its backbone which we call LisT5. Within a standard three-stage architecture, we focus mostly on the label prediction problem. We adopt a "listwise approach", where all candidate sentences that form the evidence set of a claim are considered together. Our main contribution is a data augmentation technique that involves deliberately introducing noise into training data to combat data sparsity and produce a more robust model. At its introduction, a full pipeline using our techniques represents the state of the art, achieving the top scoring run on the FEVER leaderboard. An additional minor contribution exploits named entities during the sentence selection phase, which has a small but noticeable effect on generating a better candidate set for downstream label prediction. We believe that these techniques can be potentially valuable to a broader range of NLP tasks that also involve aggregation of information from upstream retrieval models.

2 Background and Related Work

As this work focuses on the Fact Extraction and VERification (FEVER) task (Thorne et al., 2018),[1] we begin by briefly describing the task setup. We are given a textual claim q, to be verified against a corpus comprised of a subset of Wikipedia. Each claim is associated with a three-way veracity label $v(q) \in \{\text{SUPPORTS}, \text{NOINFO}, \text{REFUTES}\}$ and a set of reference sentences $S(q)$ that provide support.[2] An example claim q, its label $v(q)$, and supporting evidence $S(q)$ are given in Figure 1.

The primary evaluation metric, FEVER score, is computed as the proportion of claims where the system has predicted the correct veracity label conditioned on also having retrieved a complete set of reference sentences. Most current systems adopt a three-stage approach to this task, comprising document retrieval, sentence selection, and label prediction. In this work, our contributions are focused on the second and third sub-tasks; for document retrieval, we simply augment current best practices with BM25 (Yang et al., 2017; Lin et al., 2021).

[1] Details of the FEVER sets are included in Appendix A.1.
[2] Each claim may have multiple different sets of reference sentences, any of which is sufficient as the support set.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 402–410
August 1–6, 2021.

Claim: The Rodney King riots took place in the most populous county in the USA.
Evidence 1 (`wiki/Los_Angeles_Riots`): The 1992 Los Angeles riots, also known as the Rodney King riots were a series of riots, lootings, arsons, and civil disturbances that occurred in Los Angeles County, California in April and May 1992.
Evidence 2 (`wiki/Los_Angeles_County`): Los Angeles County, officially the County of Los Angeles, is the most populous county in the USA.
Label: SUPPORTS

Figure 1: An example claim and its corresponding evidence and label from the FEVER dataset.

By construction, the veracity of each claim is determined by the (candidate) supporting sentences, taken together. One simple and popular approach to fact extraction and verification is to consider the veracity of the claim with respect to each candidate *independently* (i.e., classification), and then aggregate the evidence (Hanselowski et al., 2018; Zhou et al., 2019; Soleimani et al., 2019; Liu et al., 2020; Pradeep et al., 2021b). For convenience, we refer to these as "pointwise approaches", borrowing from the learning to rank literature (Li, 2011).

As an alternative, researchers have proposed approaches that consider multiple candidates at once to *jointly* arrive at a veracity prediction (Thorne et al., 2018; Nie et al., 2019a; Zhou et al., 2019; Stammbach and Neumann, 2019; Pradeep et al., 2021a). For convenience, we refer to these as "listwise approaches", also borrowing from the learning to rank literature (Li, 2011). Such listwise approaches have also been used for information aggregation in other NLP tasks such as question answering (Wang et al., 2018; Nie et al., 2019b). At a high level, this strategy suffers from a number of challenges, including data sparsity and a high level of sensitivity to noisy inputs. Following this thread of work, we adopt the listwise approach and improve it by training with a data augmentation technique that involves deliberately introducing noise into the training data to produce a more robust model.

3 Methods

Our work adopts a three-stage pipeline comprising document retrieval, sentence selection, and label prediction, which we detail in this section.

3.1 Document Retrieval

Given a claim q, our first step is to retrieve the top K most relevant documents $\hat{D}(q) = \{d_1, ..., d_K\}$. Since the corpus contains over 5.4M documents, we first perform document retrieval to narrow our search space. We leverage the Pyserini toolkit (Yang et al., 2017; Lin et al., 2021), which is based on the popular Lucene search engine, using the BM25 scoring function (Robertson and Zaragoza, 2009) to rank documents. Additional document retrieval details are described in Appendix A.3. We also incorporate document retrieval using the MediaWiki API, which has been shown in previous work to form a strong baseline (Hanselowski et al., 2018). We combine the results of the two methods by alternating through the two ranked lists of documents, skipping duplicates and keeping the top K unique documents.

3.2 Sentence Selection

Given a claim q and retrieved documents $\hat{D}(q)$, the next stage in the pipeline selects the top L most relevant evidence sentences $\hat{S}(q) = \{s_{k_1,i_1}, ..., s_{k_L,i_L}\}$, where $s_{k,i}$ is the i-th sentence from document d_k. Similar to how Soleimani et al. (2019) and Subramanian and Lee (2020) frame this stage as a semantic matching problem using BERT-based models, we use T5 to rank the similarities between the claim and the sentences in each document. Introduced by Nogueira et al. (2020), like Pradeep et al. (2021a), we use T5 (Raffel et al., 2020) as a pointwise reranker, which they dub monoT5. Empirically, T5 has been found to be more effective at ranking than BERT-based models across a wide variety of domains.

As a sequence-to-sequence model, ranking is performed using the following input template:

`Query:` q `Document:` $s_{k,i}$ `Relevant:`

where q and $s_{k,i}$ are the claim and evidence sentence, respectively. To provide a broader context and to resolve ambiguities, we prepend each sentence $s_{k,i}$ with the title of document d_k.

We fine-tune the model to generate the token "true" if $s_{k,i} \in S(q)$ and "false" otherwise. In terms of training data for fine-tuning, we use the gold evidence in the evidence sets in $S(q)$ for "true" samples, but for the "false" samples, we sample negatives from the sentences in $\hat{D}(q)$.

At inference time, we construct a candidate set comprised of sentences from each document in $\hat{D}(q)$ in its retrieved order. Using the same input

format, for each sentence, we probe the logits of the "true" and "false" tokens and apply the softmax function to produce a relevance probability score between 0 and 1; these scores are used to select the top L (= 5) sentences. For efficiency, instead of reranking all sentences in $\hat{D}(q)$, we take the first 200 sentences and only rerank this subset. Since there is an average of five non-empty sentences per document, we are roughly considering the top 40 documents from $\hat{D}(q)$.

On top of the basic reranking input template of Nogueira et al. (2020), we introduce a novel enhancement where we append any named entities found within the claim to the input of monoT5. The intuition here is to prompt monoT5 to promote sentences that come from documents with titles that are similar to those entities, which tend to contain information that is relevant to verifying the claims. During fine-tuning, we use the names of the documents that contain the gold evidence as entities, but during inference, we extract named entities from the claims using the named entity recognition (NER) module built into spaCy's `en_core_web_sm` model.[3] We append these entities, denoted as $e_1, ..., e_j$, to our monoT5 input template as follows:

`Query:` q `Document:` $s_{k,i}$ `Entity1:` e_1 $\cdots$ `Entity`j`:` e_j `Relevant:`

Additional details are described in Appendix A.3.

3.3 Label Prediction

Given claim q and evidence $\hat{S}(q)$, the final stage of the pipeline is to predict a veracity label $\hat{v}(q)$.

Pointwise Aggregation One common method in the literature for label prediction is to combine the claim and each evidence sentence individually as the inputs to some model and aggregate those model outputs to obtain a veracity prediction. With the sequence-to-sequence nature of T5, we achieve this by fine-tuning the model with samples of the following input sequence:

`query:` q `sentence:` $s_{k,i}$ `relevant:`

There are many different methods to aggregate the outputs: Soleimani et al. (2019) assumes NOINFO unless there are unanimous outputs of SUPPORTS or REFUTES, while Zhou et al. (2019) chooses the most frequently occurring label as well as attending over the outputs with the vector representations of each claim and evidence pair. Assume that the input sequence with evidence sentence $s_{k,i}$, after passing through T5 and applying the softmax function to the logits of the three classes, produces the probabilities $\Pr(\text{S} \mid q, s_{k,i})$ for SUPPORTS, $\Pr(\text{R} \mid q, s_{k,i})$ for REFUTES, and $\Pr(\text{N} \mid q, s_{k,i})$ for NOINFO. We experiment with two aggregation schemes that achieves the best results for us, which we denote by `sum` and `max`, as follows:

$$\texttt{sum}: \hat{v}(q) = \underset{l \in \{\text{S,R,N}\}}{\text{argmax}} \sum_{s_{k,i} \in \hat{S}(q)} \Pr(l \mid q, s_{k,i})$$

$$\texttt{max}: \hat{v}(q) = \underset{l \in \{\text{S,R,N}\}}{\text{argmax}} \max_{s_{k,i} \in \hat{S}(q)} \Pr(l \mid q, s_{k,i})$$

For fine-tuning, we use $S(q)$ as the evidence for SUPPORTS and REFUTES samples. Similar to sentence selection, for NOINFO samples, we sample negatives from the top predicted sentences from upstream, which in this case, is sentence selection, using the full reranked candidate list instead of just the top L sentences in $\hat{S}(q)$.

Listwise Concatenation Another common strategy for label prediction is to concatenate all L sentences into a single input to some model and have the model directly classify the claim and list of evidence $\hat{S}(q)$ as one of SUPPORTS, NOINFO, and REFUTES. Again, with T5, we use the following input sequence:

`query:` q `sentence1:` s_{k_1,i_1} $\cdots$ `sentence`L`:` s_{k_L,i_L} `relevant:`

To obtain fine-tuning training data, we use the same method as for pointwise aggregation.

Listwise Data Augmentation To make label prediction more tolerant to noisy evidence in the top L sentences, we fine-tune T5 with augmented, noisy evidence sets: this mimics the model during inference more closely as there usually exists some non-gold evidence in $\hat{S}(q)$. To accomplish this, instead of fine-tuning directly with the gold evidence sets $S(q)$, we fine-tune using $I(S(q))$, which "infuses" $S(q)$ with $\hat{S}(q)$. Specifically, we define the transformation I as:

- If $v(q) \in \{\text{SUPPORTS}, \text{REFUTES}\}$, we check if $S(q) \subseteq \hat{S}(q)$. For each $s \in S(q)$ such that $s \notin \hat{S}(q)$, we randomly select an index k of $\hat{S}(q)$ where $\hat{S}(q)[k] \notin S(q)$ and insert s at $\hat{S}(q)[k]$. This is repeated iteratively, and so $I(S(q))$ returns the resulting list of sentences $\hat{S}(q)$.
- If $v(q) = \text{NOINFO}$, $I(S(q)) = \hat{S}(q)$.

[3] https://spacy.io

Method		Dev		Test	
		LA (%)	FS (%)	LA (%)	FS (%)
(1a)	UNC (Nie et al., 2019a)	66.14	69.60	72.56	67.26
(1b)	Soleimani et al. (2019)	72.42	74.59	71.86	69.66
(1c)	HESM (Subramanian and Lee, 2020)	75.77	73.44	74.64	71.48
(1d)	CorefRoBERTa (Ye et al., 2020)	–	–	75.96	72.30
(1e)	GEAR (Zhou et al., 2019)	74.84	70.69	71.60	67.10
(1f)	DREAM (Zhong et al., 2020)	–	–	76.85	70.60
(1g)	`nudt_nlp`*	–	–	77.38	74.42
(1h)	`dominiks`*	–	–	76.60	74.27
(2a)	Oracle	–	94.74	–	–
(2b)	T5 w/ `sum` pointwise aggregation	63.19	59.45	–	–
(2c)	T5 w/ `max` pointwise aggregation	70.31	66.15	–	–
(2d)	T5 w/ listwise concatenation	70.66	67.18	–	–
(2e)	T5 w/ listwise data augmentation	**81.26**	**77.75**	**79.35**	**75.87**

Table 1: Label prediction results on the FEVER development set and blind test set. LA refers to label accuracy and FS refers to FEVER score. Other top submissions on the FEVER leaderboard at the time of our work are denoted with the symbol *.

Note that we use the same T5 input format as listwise concatenation. Training details for the label prediction stage can be found in Appendix A.3.

4 Results

We report the overall results of LisT5 on the FEVER development and blind test sets in Table 1, comparing the label prediction variations presented in Section 3.3. We also include the oracle FEVER score for our retrieved $\hat{S}(q)$ on line (2a). For reference, we compare LisT5 against several baselines and state-of-the-art techniques (drawn from the leaderboard) at the time of our work, shown in lines (1a)–(1h).

From the results in Table 1, it is clear that the different label prediction strategies lead to vastly different FEVER scores. The top-performing method, according to both label accuracy and FEVER score, is trained with augmented data in a listwise manner, found on line (2e). This run represents the state of the art atop the FEVER leaderboard at the time of our work. The other methods that fine-tune with only gold evidence data, found on lines (2b) to (2d), seem to trail by over 10 points. These results suggest the importance of training with augmented listwise evidence sets, which is presented in Section 3.3.

Contrary to the results reported in some papers, our concatenation methods consistently outperform corresponding aggregation methods: this suggests that T5 is able to capture inter-sentence semantics and use information from multiple, possibly diverse, pieces of evidence to come to veracity conclusions. Specifically, the T5 variant on line (2e) achieves 78.02%[4] (174/223) label accuracy on claims in the development set that require retrieving at least two pieces of evidence in conjunction to verify, which is close to our overall label accuracy of 81.26%. This finding suggests that T5 is capable of incorporating and corroborating the information contained in multiple pieces of evidence, which is one of the most common needed areas of improvement described in previous papers.

Table 2 compares the LisT5 sentence selection results of the monoT5 variations described in Section 3.2. We include some results from baselines, using recall at five as the primary sentence selection metric, which by definition is an upper-bound for the downstream FEVER score. We format the results for LisT5 as an ablation analysis focused on sentence selection. Line (2a) shows the results of the full monoT5 model with NER and fine-tuned on the FEVER dataset; monoT5 without NER features but fine-tuned on the FEVER dataset is shown on line (2b). Finally, we have zero-shot monoT5 on line (2c) to show the results of monoT5 without fine-tuning on the FEVER dataset, i.e., directly from the model checkpoints of Nogueira

[4]These only include claims where an entire gold evidence set is contained in the sentence selection output $\hat{S}(q)$.

Method		P@5 (%)	R@5 (%)	F1@5 (%)	MAP@5 (%)
(1a)	UNC (Nie et al., 2019a)	36.49	86.79	51.38	–
(1b)	Soleimani et al. (2019)	25.13	88.29	39.13	–
(1c)	HESM (Subramanian and Lee, 2020)	–	90.50	–	–
(1d)	GEAR (Zhou et al., 2019)	**40.60**	86.36	**55.23**	–
(1e)	DREAM (Zhong et al., 2020)	26.67	87.64	40.90	–
(2a)	monoT5 w/ NER (full model)	25.66	**90.54**	37.17	**85.62**
(2b)	monoT5 w/o NER (fine-tuned)	25.50	90.08	36.94	84.87
(2c)	monoT5 w/o NER (zero-shot)	22.70	85.39	33.86	76.87

Table 2: Comparison of sentence selection methods on the FEVER development set.

et al. (2020). We explain this in more detail in Appendix A.3. From these results, it is clear that monoT5 supplemented with named entities – on line (2a) – performs the best, achieving the highest recall and mean average precision, better than the other monoT5 variations or any of the baselines. It is worth noting that the full monoT5 model on line (2a) achieves 90.53 recall on the blind test set, consistent with the development set results.

While document retrieval is not our focus, our pipeline performs competitively compared to prior work and is further discussed in Appendix A.4.

5 Error Analysis

We randomly select 200 incorrectly predicted claims by LisT5 and summarize the most common issues, hoping to identify areas of improvement for future fact verification systems.

One common issue is failing to distinguish between similar but semantically different words or phrases. An example of this is the claim "Shane McMahon officially retired on the first day of 2010" to which our document retrieval and sentence selection stages retrieve the sentence "In 2009, McMahon announced his resignation from WWE which went into effect January 1, 2010". Here, retirement and resignation are semantically similar words that both describe individuals leaving their positions. These similarities may have been learned by the pretrained transformer, but it is not always the case that the words imply one another, leading to an incorrect prediction for this claim.

Another frequent issue is incorrectly labelled claims in the FEVER dataset, often due to missing evidence in $S(q)$. An example of this is the claim "Mickey Rourke appeared in a sequel" to which our document retrieval and sentence selection stages retrieve the sentence "Since then, Rourke has appeared in several commercially successful films including the 2010 films Iron Man 2 and The Expendables and the 2011 film Immortals". However, the claim was labelled NOINFO in the dataset, which is incorrect due to Iron Man 2 indeed being a sequel. In short, we are bumping into data quality issues in the annotations themselves.

6 Conclusion

In this paper, we present the LisT5 framework for automated fact verification. LisT5 consists of a three-stage pipeline – document retrieval, sentence selection, and label prediction. For document retrieval, we combine two strong document retrieval baselines. For sentence selection, we fine-tune a T5 model as a reranker with named entities provided as additional features. For label prediction, we present evidence in a listwise manner to a T5 model, trained on augmented data. Our experimental results indicate that LisT5 achieves the state of the art on the FEVER task, which we attribute to the framework's ability to reason jointly over multiple pieces of evidence.

Acknowledgments

This research was supported in part by the Canada First Research Excellence Fund, the Natural Sciences and Engineering Research Council (NSERC) of Canada, and the Waterloo–Huawei Joint Innovation Laboratory. Additionally, we would like to thank Google's TensorFlow Research Cloud (TFRC) for access to Cloud TPUs.

References

Zeynep Akkalyoncu Yilmaz, Wei Yang, Haotian Zhang, and Jimmy Lin. 2019. Cross-domain modeling of sentence-level evidence for document retrieval. In *Proceedings of the 2019 Conference on*

Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 3490–3496, Hong Kong, China.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2018. MS MARCO: A human generated MAchine Reading COmprehension dataset. *arXiv:1611.09268v3*.

Tuhin Chakrabarty, Tariq Alhindi, and Smaranda Muresan. 2018. Robust document retrieval and individual evidence modeling for fact extraction and verification. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 127–131, Brussels, Belgium.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017a. 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, Vancouver, Canada.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017b. Enhanced LSTM for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1657–1668, Vancouver, Canada.

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 *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 103–108, Brussels, Belgium.

Hang Li. 2011. *Learning to Rank for Information Retrieval and Natural Language Processing*. Morgan & Claypool Publishers.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: A Python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021)*.

Zhenghao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. 2020. Fine-grained fact verification with kernel graph attention network. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7342–7351, Online.

Yixin Nie, Haonan Chen, and Mohit Bansal. 2019a. Combining fact extraction and verification with neural semantic matching networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6859–6866.

Yixin Nie, Songhe Wang, and Mohit Bansal. 2019b. Revealing the importance of semantic retrieval for machine reading at scale. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2553–2566, Hong Kong, China.

Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. 2020. Document ranking with a pre-trained sequence-to-sequence model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 708–718.

Ronak Pradeep, Xueguang Ma, Rodrigo Nogueira, and Jimmy Lin. 2021a. Scientific claim verification with VerT5erini. In *Proceedings of the 12th International Workshop on Health Text Mining and Information Analysis*, pages 94–103, online.

Ronak Pradeep, Xueguang Ma, Rodrigo Nogueira, and Jimmy Lin. 2021b. Vera: Prediction techniques for reducing harmful misinformation in consumer health search. In *Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021)*.

Ronak Pradeep, Xueguang Ma, Xinyu Zhang, Hang Cui, Ruizhou Xu, Rodrigo Nogueira, and Jimmy Lin. 2020. H_2oloo at TREC 2020: When all you got is a hammer... Deep Learning, Health Misinformation, and Precision Medicine. In *Proceedings of the Twenty-Ninth Text REtrieval Conference (TREC 2020)*.

Ronak Pradeep, Rodrigo Nogueira, and Jimmy Lin. 2021c. The expando-mono-duo design pattern for text ranking with pretrained sequence-to-sequence models. *arXiv:2101.05667*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389.

Mike Schuster and Kuldip Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Amir Soleimani, Christof Monz, and Marcel Worring. 2019. BERT for evidence retrieval and claim verification. *arXiv:1910.02655*.

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.

Shyam Subramanian and Kyumin Lee. 2020. Hierarchical Evidence Set Modeling for automated fact extraction and verification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7798–7809, Online.

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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Shuohang Wang, Mo Yu, Jing Jiang, Wei Zhang, Xiaoxiao Guo, Shiyu Chang, Zhiguo Wang, Tim Klinger, Gerald Tesauro, and Murray Campbell. 2018. Evidence aggregation for answer re-ranking in open-domain question answering. In *Proceedings of the 6th International Conference on Learning Representations (ICLR 2018)*, Vancouver, Canada.

Peilin Yang, Hui Fang, and Jimmy Lin. 2017. Anserini: enabling the use of Lucene for information retrieval research. In *Proceedings of the 40th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2017)*, pages 1253–1256, Tokyo, Japan.

Deming Ye, Yankai Lin, Jiaju Du, Zhenghao Liu, Peng Li, Maosong Sun, and Zhiyuan Liu. 2020. Coreferential Reasoning Learning for Language Representation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7170–7186, Online.

Takuma Yoneda, Jeff Mitchell, Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. UCL machine reading group: Four factor framework for fact finding (HexaF). In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 97–102, Brussels, Belgium.

Edwin Zhang, Nikhil Gupta, Raphael Tang, Xiao Han, Ronak Pradeep, Kuang Lu, Yue Zhang, Rodrigo Nogueira, Kyunghyun Cho, Hui Fang, and Jimmy Lin. 2020. Covidex: Neural ranking models and keyword search infrastructure for the COVID-19 Open Research Dataset. In *Proceedings of the First Workshop on Scholarly Document Processing*, pages 31–41, Online.

Wanjun Zhong, Jingjing Xu, Duyu Tang, Zenan Xu, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. 2020. Reasoning over semantic-level graph for fact checking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6170–6180, Online.

Jie Zhou, Xu Han, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2019. GEAR: Graph-based evidence aggregating and reasoning for fact verification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 892–901, Florence, Italy.

A Appendix

A.1 FEVER Dataset

The dataset used for training and evaluating our fact verification system is FEVER (Thorne et al., 2018), a large-scale dataset consisting of 185K claims with evidence taken from Wikipedia. We include the label distribution of the dataset across its training, development, and blind test set in Table 3.

Split	SUPPORTS	REFUTES	NOINFO
Train	80,035	29,775	35,639
Dev	6,666	6,666	6,666
Test	6,666	6,666	6,666

Table 3: Label distribution of the FEVER dataset.

A.2 Baseline Details

As discussed in Section 3, most fact verification systems, especially for the FEVER task, consist of a three-stage pipeline similar to the one used in LisT5. The stages are as follows:

Document Retrieval Many systems use the document retrieval component of DrQA (Chen et al., 2017a), which performs retrieval with TF-IDF feature vectors along with bigram features. Some other systems leverage external search APIs, such as Hanselowski et al. (2018), who use the MediaWiki API, Wikipedia's own search engine, and Chakrabarty et al. (2018), who use the Google Search API.

Sentence Selection When the FEVER task was introduced in 2018, many of the initial top-scoring systems (Nie et al., 2019a; Hanselowski et al., 2018) employed variations of the Enhanced Sequential Inference Model (ESIM) (Chen et al., 2017b), which consists of Bidirectional Long Short-Term Memory networks (BiLSTMs) (Schuster and Paliwal, 1997) as its primary building block. However, with the advent of the transformer architecture (Vaswani et al., 2017), most systems today (Soleimani et al., 2019; Subramanian and Lee, 2020) use transformers to perform semantic matching between the claim and each candidate sentence.

Label Prediction Framing the problem as that of natural language inference (NLI), Nie et al. (2019a), Yoneda et al. (2018), and Hanselowski et al. (2018) again use variations of ESIM for label prediction. Similar to the sentence selection stage, many recent systems use transformers for this stage as well. However, there has also been active research into graph-based models for knowledge aggregation by modelling evidence sentences as nodes in a graph (Zhou et al., 2019; Liu et al., 2020; Zhong et al., 2020).

A.3 Implementation and Training Details

Document Retrieval We retrieve with BM25 using the parameters $k_1 = 0.6$ and $b = 0.5$. These parameters are tuned by running a grid search over parameter values in 0.1 increments over a subset of the training set.

Sentence Selection Whenever we fine-tune monoT5, we use the T5-3B variant, which as its name suggests, contains three billion parameters. We fine-tune the model with batch size 128 over one epoch, using the configurations prescribed by Raffel et al. (2020), except that we use learning rate 0.0001 instead of 0.001. While training, we save checkpoints at evenly spaced iteration intervals, usually around 1000 iterations per checkpoint depending on the size of the training data. Thus, whenever we report the results of a model, we use the results of the best performing checkpoint on the FEVER development set. We fine-tune on TPU v3-8 nodes on the Google Cloud Platform, which takes around 24 hours.

Note that we first fine-tune a pretrained T5 model on the MS MARCO passage dataset (Bajaj et al., 2018) for 10000 iterations, following best practices reported in previous work (Akkalyoncu Yilmaz et al., 2019; Nogueira et al., 2020; Zhang et al., 2020; Pradeep et al., 2020, 2021c,b), which has shown that this leads to improved effectiveness. This procedure also gives us a zero-shot setting for fact verification, which we experiment with before fine-tuning on the FEVER dataset directly.

In our experiments, we note that negative sampling sentences from highly-ranked documents in $\hat{D}(q)$ leads to poorly performing models. This may be due to false negatives in the data, where some claims are labelled as NOINFO but are actually verifiable, with relevant evidence retrieved by our document retrieval stage. To avoid negative sampling such false negative evidence, we negative sample sentences ranked between 50 and 200.

Label Prediction Again, we use the T5-3B variant as the model for label prediction. We use similar settings for fine-tuning T5 as before for monoT5, except that we use the default learning rate 0.001.

Method	R@1000 (%)
MediaWiki API	89.56
Anserini	94.76
Anserini + MediaWiki API	**96.87**

Table 4: Comparison of document retrieval methods on the FEVER development set. The code for retrieval using the MediaWiki API is courtesy of Hanselowski et al. (2018).

We also fine-tune on TPU v3-8 nodes on the Google Cloud Platform, which takes around 8 hours.

To avoid similar negative sampling issues encountered in fine-tuning models for sentence selection, we sample from sentences ranked between 10 and 25 here.

A.4 Document Retrieval Results

We report the importance of combining the two document retrieval methods described in Section 3.1 by comparing their recall at rank 1000 in Table 4. These figures show that combining the two techniques results in being only a few points away from perfectly retrieving all relevant documents.