Automatic Fact-Checking with Document-level Annotations using BERT and Multiple Instance Learning

Aalok Sathe[*]
Massachusetts Institute of Technology
asathe@mit.edu

Joonsuk Park
University of Richmond
NAVER AI Lab
park@joonsuk.org

Abstract

Automatic fact-checking is crucial for recognizing misinformation spreading on the internet. Most existing fact-checkers break down the process into several subtasks, one of which determines candidate evidence sentences that can potentially support or refute the claim to be verified; typically, evidence sentences with gold-standard labels are needed for this. In a more realistic setting, however, such sentence-level annotations are not available. In this paper, we tackle the natural language inference (NLI) subtask—given a document and a (sentence) claim, determine whether the document supports or refutes the claim—only using document-level annotations. Using fine-tuned BERT and multiple instance learning, we achieve 81.9% accuracy, significantly outperforming the existing results on the WikiFactCheck-English dataset.

1 Introduction

As the volume of information present on the internet steeply increases, automatic fact-checking has become a promising approach to identify and stop the spread of misinformation. Active research in this area has been supported in part by a handful of carefully curated datasets (Thorne et al., 2018a; Hanselowski et al., 2019; Wadden et al., 2020).

While these datasets have been playing a crucial role in the development of the latest fact-checkers, they do not faithfully represent the reality in certain aspects. First, these datasets come with short evidence snippets for the claims; most existing fact-checkers rely on such sentence-level evidence annotations to build the natural language inference (NLI) component of their systems—e.g. given an evidence sentence and a claim sentence, determine if the claim is supported (Thorne et al., 2018b). Second, most of the datasets consist of synthetic claims written by annotators based on snippets of evidence. Both aspects render it difficult to readily apply the findings in real applications.

Field	Content
id	115724
claim	The hindwings are uniform grey with a narrow marginal line.
context	Eupoca sanctalis is a moth in the Crambidae family. [...]
refuted	The hindwings are uniform blue with a broad marginal line.
url	http://digitalcommons.unl.edu/[...]

Table 1: Example entry from WikiFactCheck-English. The URL is for the evidence document cited in support of the claim in the Wikipedia article.

WikiFactCheck-English (Sathe et al., 2020) was constructed to address the aforementioned concerns. The dataset consists of 124k entries each consisting of a claim[1], context, and evidence document extracted from English Wikipedia articles. (See Table 1 for an example.) We believe that the real claims and lengthy evidence documents without sentence-level annotations will lead to fact-checkers that can better handle claims in the wild.

In this paper, we tackle the NLI subtask—given a document and a (sentence) claim, determine whether the document supports or refutes the claim—only using document-level annotations. We improve on existing systems trained and tested on the WikiFactCheck-English dataset during both steps of the 2-step pipeline: evidence retrieval and support verification. We find that fine-tuned BERT with multiple instance learning (MIL)—to use multiple candidate evidence sentences—results in about 13% increase in accuracy over the baseline. However, incorporating Sentence-BERT (SBERT) (Reimers and Gurevych, 2019) to identify candidate evidence sentences during evidence

[*] Work done while at the University of Richmond

[1] 34k of these come with manually written 'refuted' claims.

retrieval does not lead to a noticeable improvement.

2 Related Works

Many recent works on automatic fact-checking have attempted to employ Transformers, BERT in particular. This was motivated by BERT having showed promising results for many NLP tasks (Devlin et al., 2019). To mention a few relevant examples, Soleimani et al. (2020) tackle the shared task from the third FEVER workshop (Christodoulopoulos et al., 2020) by constructing a 2-step pipeline. They use BERT in the evidence retrieval step to rank evidence by relevance. Then, BERT is employed once more in the support verification step to make the final prediction using the retrieved evidence. We also have a 2-step pipeline, but methods used in each step is distinct from their work.

Lee et al. (2020) take a new approach using BERT to the otherwise traditional pipeline of fact-checking in FEVER-like tasks. The authors treat BERT as a knowledge base and use its masked language modeling predictions to decide the factual correctness of the claim. While it is a novel approach, it does not provide a promising performance gain in practice. Also, we are working with a dataset where the claims are extracted from Wikipedia, but BERT uses Wikipedia as part of its training. For these reasons, we decided that this approach would not be suitable in our work.

Zhong et al. (2020) highlight the complexity of the fact-checking task, where more than one sentence collectively support or refute a claim. Thus, the authors create semantic graphs and reason over these structures using graph convolutional network and graph attention netowrk. There report improvement in performance in the FEVER task (Thorne et al., 2018a). In our work, we assume a simple semantic structure of evidence sentences. This is based on an observation that when people cite a document in support of their factual claim in Wikipedia, the claim tends to be one of the main points of the document. And points are typically stated in a single sentence or span over consecutive sentences.

The work we build off of in this paper is from Sathe et al. (2020). The authors released a new fact-checking dataset with only document-level annotations that we use in this work. Along with it, they presented baseline systems consisting of a two-step pipeline: evidence retrieval and support verification. In our work, we adopt the same pipeline but make improvements to each step using BERT-based approaches. Also, unlike their approach, we retrieve multiple sentences during the evidence retrieval step and compute the final aggregated prediction using multiple instance learning.

3 Method

We tackle the NLI subtask of automatic fact checking—given a relevant document (a list of sentences E_c) and a claim c, determine whether the E_c supports or refutes c. This is done in two-steps: (1) evidence retrieval and (2) support verification, as shown in Figure 1. The same pipeline was used in Sathe et al. (2020); we improve both steps using BERT-based approaches and multiple instance learning to make use of multiple candidate evidence sentences.

3.1 Evidence Retrieval

The evidence retrieval step involves retrieving candidate evidence sentence(s) from the evidence document E_c that either supports or refutes the claim c. We retrieve the top k sentences $(e_1, e_2, ..., e_k) \in E_c$ that are most similar to c. These in turn are likely to be relevant for verifying c.

We improve on the Levenshtein distance (LD) used in the baseline (Sathe et al., 2020), by incorporating SBERT (Reimers and Gurevych, 2019). SBERT uses siamese and triplet networks to derive sentence embeddings capturing semantic similarities. Because the claims and evidence documents in the WikiFactCheck-English dataset are drawn from distinct sources, they are often dissimilar in their content and style. Thus, we expect SBERT capturing semantic similarity to be more effective than LD capturing surface-level textual similarity. We use a pretrained SBERT architecture fine-tuned to STS-B (semantic textual similarity benchmark), part of the popular NLU benchmark set 'GLUE' (generalized language understanding and evaluation) (Wang et al., 2018).

Given the large number of sentences in E_c, we first filter out sentences least-likely to be the evidence sentence: Of the sentence in E_c that have at least 1 word overlap with the claim, we use SBERT with a classification layer outputting a scalar similarity score to determine k most similar sentences for $k = 1, 3, 5$.

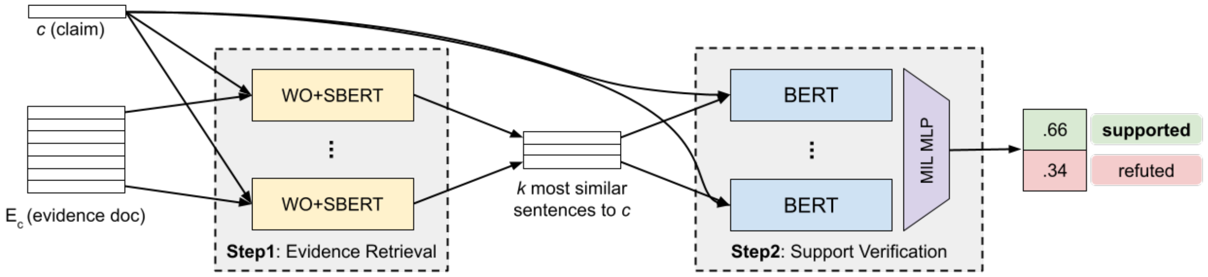

Figure 1: The 2-step pipeline to determine if evidence document E_c supports or refutes claim c. The final output is a distribution over probabilities of E_c supporting and refuting c. The components depicted are of our best performing system: the evidence retrieval step consists of word overlap (WO) and Sentence-BERT (SBERT); the support verification step consists of BERT and multiple instance learning (MIL).

3.2 Support Verification

The support verification step takes a claim c and a list of candidate evidence sentence(s) $e_1, e_2, ..., e_k \in E_c$ from the previous step and outputs a distribution over two labels denoting the relative likelihood of whether the claim is 'supported' or 'refuted' by E_c. Our approach to this step involves two subparts: natural language inference (NLI) and attention-based aggregation. Again, E_c may contain one or more sentences that support or refute c, but there is no sentence-level gold-standard label for NLI for each sentence in E_c.

Thus, we use multiple instance learning (MIL) (Angelidis and Lapata, 2018; Ilse et al., 2018) to learn sentence-level labels in a semi-supervised manner by aggregating over multiple labels and computing loss at the document level.

In the following, $\mathbf{BERT}\langle\texttt{[CLS]}...\rangle$ denotes the contextualized representation of the $\texttt{[CLS]}$ token obtained after passing in $\texttt{[CLS]}, \texttt{E}_1, \texttt{E}_2, \ldots \texttt{[SEP]}, \texttt{C}_1, \texttt{C}_2, \ldots$ as the input to BERT. Here, the `[CLS]` token is a special token whose representation is meant to capture relevant features of the input for classification (Devlin et al., 2019).

We first initialize a binary ("supported" and "refuted") NLI classifier using $\mathbf{BERT}\langle\texttt{[CLS]}...\rangle$:

$$\mathbf{y}' = \mathit{ReLU}\left(W \cdot \mathbf{BERT}\langle\texttt{[CLS]}...\rangle + b_W\right) \quad (1)$$

where $\mathbf{y}'$ contains the predictions for the claim and evidence sentence pairs.

Notice that for a claim c and top-k candidate evidence sentences $e_1, e_2, \ldots, e_k$, we have k input pairs $\langle e_i, c\rangle$. For each pair, we have an NLI prediction $\mathbf{y}'_i$. However, we only have the ground truth label corresponding to $\langle E_c, c\rangle$. Therefore, we will use another MLP to transform the same representation used for NLI to compute softmaxed attention weights $\mathbf{a}$ over the sentences.

$$\mathbf{a}' = \mathit{Sigmoid}\left(V \cdot \mathbf{BERT}\langle\texttt{[CLS]}...\rangle + b_V\right) \quad (2)$$

$$\mathbf{a} = \mathit{Softmax}(\mathbf{a}') \quad (3)$$

Here, each element $\mathbf{a}_i$ of $\mathbf{a}$ is the attention weight for aggregating the predictions for $\langle c, e_i\rangle$. Then our aggregated document-level predicted label is:

$$\mathbf{y} = \mathbf{a}^T \mathbf{y}' \quad (4)$$

where $\mathbf{y}$ is a vector giving the probability distribution over the two labels, "supported" and "refuted". Loss is computed using Cross Entropy loss with the appropriate ground truth label. In the case when $k = 1$, the attention weight for the single prediction is by default 1, and this automatically reduces to not using MIL.

4 Experiments

4.1 Setup

The experiments were conducted on the WikiFactCheck-English dataset. We trained the models using a randomly sampled set of 10k supported claims, each with a corresponding refuted claim, for a total of 20k examples selected from the training set. Then, the performances were measured on the heldout test set of about 10k supported claims, each with a corresponding refuted claim, for a total of about 20k examples to match the setup from Sathe et al. (2020).

We performed a hyperparameter search for learning rates between $1e{-}5$ to $5e{-}4$ in increments of powers of 10; for gradient accumulation steps between 4 and 16 in increments of 4; pretrained BERT models among `bert-base-uncased` (the 'vanilla' BERT) and `bert-base-mnli` (vanilla BERT finetuned on the MNLI dataset from Williams et al. (2018)).

System	System Components			Performance			
	Evidence Retrieval	k	Support Verification	Prec.	Rec.	F1	Acc.
Sathe et al. (2020)	(WO +) LD	1	Logistic Regression	.664	.729	.695	.680
LD + BERT	(WO +) LD	1	BERT	.856	.748	.798	.811
	(WO +) LD	3	BERT (with MIL)	.845	**.767**	.804	.813
	(WO +) LD	5	BERT (with MIL)	.847	**.767**	.805	.815
SBERT + BERT	(WO +) SBERT	1	BERT	.853	.746	.796	.809
	(WO +) SBERT	3	BERT (with MIL)	.855	.762	.806	.816
	(WO +) SBERT	5	BERT (with MIL)	**.857**	**.767**	**.810**	**.819**

Table 2: We take the best performing system (with respect to accuracy) by Sathe et al. (2020) as the baseline. In that system, Levenshtein distance (LD) is used to find k=1 most similar sentence to the claim in the evidence document, among the sentences that have at least one word overlap (WO) with the claim. Then, logistic regression is used to make a prediction. LD + BERT improves on the baseline in the support verification step using BERT and larger k's—# of candidate evidence sentences—with multiple instance learning (MIL). SBERT+BERT improves on LD+BERT by using SBERT to find k most similar sentences to the claim among those with one or more WO.

4.2 Results and Analysis

As summarized in Table 2, the best performing system, SBERT+BERT with $k = 5$, achieves 81.9% accuracy and 81.0% F1. This is a significant improvement over the best performing system from Sathe et al. (2020). Most of the improvement is attributable to the use of fine-tuned BERT during support verification; simply replacing logistic regression with BERT in the inference stage leads to 13% increase in accuracy. This is expected given the strong performance of BERT on various NLP tasks.

Small additional gains seem to result from using multiple candidate evidence sentences. For $k = 1, 3, 5$, there is a mostly consistent upward trend in performance across the board, though the magnitude is small. The increase in performance is greater between $k = 1$ and 3, compared that between to 3 and 5, meaning the positive impact may diminish as k increases. In theory, using multiple candidate evidence sentences can be helpful in at least two ways. First, it can reduce the error propagating from evidence retrieval to support verification. That is, it is less likely that the true evidence sentence is not included in the top-5 most similar sentences than in the top-1. Second, multiple sentences may collectively serve as evidence for a given claim. In this case, each evidence sentence would partially verify the claim, thus having a single evidence sentence would not be enough.

Incorporating SBERT in evidence retrieval improves the performance only when multiple evidence sentences are used. And when it does help, the difference is marginal. This suggests that identifying evidence sentences based on LD, which quantifies the textual similarity, is comparable to relying on SBERT, which is intended to measure semantic similarity. We suspect that this could be due to the characteristics of this particular dataset; the evidence documents are previously published documents cited in support of factual claims made on Wikipedia articles. Often, these claims are very similar to a sentence in the document, as there is little incentive to rephrase the claim. This is different from other datasets in which annotators are instructed to write novel claims based on evidence sentences or snippets.

5 Conclusion and Future Work

We presented a 2-step pipeline to determine if a document supports or refutes a claim only using document-level annotations. Fine-tuned BERT with MIL to use multiple candidate evidence sentences resulted in about 13% increase in accuracy over the baseline. However, incorporating SBERT in evidence retrieval did not lead to additional gains that are noticeable.

In the future, we want to leverage the context, as sentences in this dataset, and in the wild, are often not self-contained. We expect the context to be useful for disambiguating keywords in both evidence retrieval and support verification. Note that both the claim and evidence sentences have contexts, and the context of an evidence sentence is a potential evidence sentence itself. Both opens up many possibilities to be explored.

Acknowledgments

We would like to thank the University of Richmond and the Thomas F. and Kate Miller Jeffress Memorial Trust, Bank of America, Trustee for their generous support for this project. We also thank the anonymous reviewers for their comments.

References

Stefanos Angelidis and Mirella Lapata. 2018. Multiple instance learning networks for fine-grained sentiment analysis. *Transactions of the Association for Computational Linguistics*, 6:17–31.

Pepa Atanasova, Preslav Nakov, Lluís Màrquez, Alberto Barrón-Cedeño, Georgi Karadzhov, Tsvetomila Mihaylova, Mitra Mohtarami, and James Glass. 2019. Automatic fact-checking using context and discourse information. *Journal of Data and Information Quality (JDIQ)*, 11(3):1–27.

Lukas Biewald. 2020. Experiment tracking with weights and biases. Software available from wandb.com.

Yonatan Bilu, Daniel Hershcovich, and Noam Slonim. 2015. Automatic claim negation: why, how and when. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 84–93.

Peter Bloem. 2019. Transformers from scratch.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

John Burger and Lisa Ferro. 2005. Generating an entailment corpus from news headlines. pages 49–54. Association for Computational Linguistics.

Christos Christodoulopoulos, James Thorne, Andreas Vlachos, Oana Cocarascu, and Arpit Mittal, editors. 2020. *Proceedings of the Third Workshop on Fact Extraction and VERification (FEVER)*. Association for Computational Linguistics, Online.

Giovanni Luca Ciampaglia, Prashant Shiralkar, Luis M Rocha, Johan Bollen, Filippo Menczer, and Alessandro Flammini. 2015. Computational fact checking from knowledge networks. *PloS one*, 10(6):e0128193.

Ido Dagan, Bill Dolan, Bernardo Magnini, and Dan Roth. 2010. Recognizing textual entailment: Rational, evaluation and approaches–erratum. *Natural Language Engineering*, 16(1):105–105.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*.

Andrea Esuli and Fabrizio Sebastiani. 2006. Sentiwordnet: A publicly available lexical resource for opinion mining. In *LREC*, volume 6, pages 417–422. Citeseer.

William Ferreira and Andreas Vlachos. 2016. Emergent: a novel data-set for stance classification. pages 1163–1168.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. Allennlp: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6.

Swapnil Ghuge and Arindam Bhattacharya. 2014. Survey in textual entailment.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112.

Andreas Hanselowski, Christian Stab, Claudia Schulz, Zile Li, and Iryna Gurevych. 2019. A richly annotated corpus for different tasks in automated fact-checking. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 493–503, Hong Kong, China. Association for Computational Linguistics.

Andrew Hickl, John Williams, Jeremy Bensley, Kirk Roberts, Bryan Rink, and Ying Shi. 2006. Recognizing textual entailment with lcc's groundhog system. volume 18.

Christopher Hidey, Tuhin Chakrabarty, Tariq Alhindi, Siddharth Varia, Kriste Krstovski, Mona Diab, and Smaranda Muresan. 2020. Deseption: Dual sequence prediction and adversarial examples for improved fact-checking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8593–8606.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

M Ilse, JM Tomczak, and M Welling. 2018. Attention-based deep multiple instance learning. *Proceedings of Machine Learning Research*, 80:2127–2136.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. Scitail: A textual entailment dataset from science question answering. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Nayeon Lee, Belinda Z Li, Sinong Wang, Wen-tau Yih, Hao Ma, and Madian Khabsa. 2020. Language models as fact checkers? In *Proceedings of the*

Third Workshop on Fact Extraction and VERification (FEVER), pages 36–41.

Edward Loper and Steven Bird. 2002. NLTK: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics*, pages 63–70.

Daniel Marcu and Abdessamad Echihabi. 2002. An unsupervised approach to recognizing discourse relations. In *ACL*, pages 368–375.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255.

Joonsuk Park and Claire Cardie. 2012. Improving implicit discourse relation recognition through feature set optimization. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, SIGDIAL '12, pages 108–112, Stroudsburg, PA, USA. Association for Computational Linguistics.

Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. 2018. Automatic detection of fake news. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3391–3401.

Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. 2015. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pages 2641–2649.

Yada Pruksachatkun, Jason Phang, Haokun Liu, Phu Mon Htut, Xiaoyi Zhang, Richard Yuanzhe Pang, Clara Vania, Katharina Kann, and Samuel Bowman. 2020. Intermediate-task transfer learning with pretrained language models: When and why does it work? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5231–5247.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. 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 3982–3992.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2021. A Primer in BERTology: What We Know About How BERT Works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Aalok Sathe, Salar Ather, Tuan Manh Le, Nathan Perry, and Joonsuk Park. 2020. Automated fact-checking of claims from wikipedia. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 6874–6882.

Karishma Sharma, Feng Qian, He Jiang, Natali Ruchansky, Ming Zhang, and Yan Liu. 2019. Combating fake news: A survey on identification and mitigation techniques. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(3):21.

Amir Soleimani, Christof Monz, and Marcel Worring. 2020. Bert for evidence retrieval and claim verification. In *European Conference on Information Retrieval*, pages 359–366. Springer.

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.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018b. The fact extraction and verification (fever) shared task. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 1–9.

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.

Andreas Vlachos and Sebastian Riedel. 2014. Fact checking: Task definition and dataset construction. pages 18–22.

Soroush Vosoughi, Deb Roy, and Sinan Aral. 2018. The spread of true and false news online. *Science*, 359(6380):1146–1151.

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.

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.

William Yang Wang. 2017. "liar, liar pants on fire": A new benchmark dataset for fake news detection. In *Proceedings of the 55th Annual Meeting of the*

Association for Computational Linguistics (Volume 2: Short Papers), pages 422–426.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. 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 1112–1122.

Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, Morgan Funtowicz, Joe Davison, Sam Shleifer, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

Wenpeng Yin and Dan Roth. 2018. Twowingos: A two-wing optimization strategy for evidential claim verification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 105–114.

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.