**WhatTheWikiFact**: Fact-Checking Claims Against Wikipedia

Anton Chernyavskiy  
HSE University  
Moscow, Russia  
aschernyavskiy_1@edu.hse.ru

Dmitry Ilvovsky  
HSE University  
Moscow, Russia  
dilovsky@hse.ru

Preslav Nakov  
Qatar Computing Research Institute, HBKU  
Doha, Qatar  
pnakov@hbku.edu.qa

**ABSTRACT**

The rise of Internet has made it a major source of information. Unfortunately, not all information online is true, and thus a number of fact-checking initiatives have been launched, both manual and automatic, to deal with the problem. Here, we present our contribution in this regard: WhatTheWikiFact, a system for automatic claim verification using Wikipedia. The system can predict the veracity of an input claim, and it further shows the evidence it has retrieved as part of the verification process. It shows confidence scores and a list of relevant Wikipedia articles, together with detailed information about each article, including the phrase used to retrieve it, the most relevant sentences extracted from it and their stance with respect to the input claim, as well as the associated probabilities. The system supports several languages: Bulgarian, English, and Russian.

**CCS CONCEPTS**

- Information systems → Information retrieval; Information systems applications; Web searching and information discovery; Computer systems organization → Real-time systems; Applied computing → Document management and text processing; Computing methodologies → Natural language processing; Information extraction.

**KEYWORDS**

Fact-checking, Factuality, Veracity, Fake News, Disinformation, Misinformation, Document Retrieval, Sentence Retrieval, Stance Detection, Natural Language Inference, Wikipedia.

**ACM Reference Format**:

Anton Chernyavskiy, Dmitry Ilvovsky, and Preslav Nakov. 2021. WhatTheWikiFact: Fact-Checking Claims Against Wikipedia. In Proceedings of . ACM, New York, NY, USA, 6 pages.

**1 INTRODUCTION**

Internet is abundant in platforms that allow users to share information online such as social networks, blogs, and forums. Unfortunately, not all information online is true, and there is a need to verify questionable claims that users encounter online.

As manual fact-checking is a complex and time-consuming task, it is important to develop tools that can help automate the process. Various task formulations have been proposed to automate fact-checking, e.g., SemEval RumourEval tasks [13, 20], the SemEval task on Fact Checking in Community Question Answering Forums [34], the CLEF CheckThat! lab [6, 16, 38, 40], the Fake News Challenge [21], and the FEVER task on Fact Extraction and VERification [3, 63, 64]. Here, we focus on the FEVER task, as it offers a large-scale training dataset, and enables explainable systems.

Interestingly, despite the popularity of research using the FEVER task formulation and its dataset, to the best of our knowledge, there are no publicly available running systems based on it. Here, we aim to bridge this gap with our WhatTheWikiFact system, which allows users to check claims against Wikipedia. It supports several languages (Bulgarian, English, and Russian) and uses a local Wikipedia for each language. The system first identifies relevant Wikipedia pages, and then finds relevant sentences within them. Then, it analyzes each claim submitted as an input against all extracted text fragments and makes a final verdict on the veracity of the claim: Truth, Lying, or Not Enough Info. WhatTheWikiFact further displays all the information it used to make its decision, together with intermediate results for each verified claim, which includes (i) a list of the most relevant documents with links to the corresponding Wikipedia pages, and (ii) detailed information about each document as a table of retrieved text fragments (position in the document | text fragment), and a bar chart showing the confidence of the classifier in each label (Supports, Refutes or Not Enough Info) for each text fragment. This allows the user to quickly analyze the result.

The remainder of this paper is organized as follows: Section 2 discusses related work. Section 3 describes the dataset we used for training. Section 4 offers an overview of the system and its components. Section 4 discusses the core implementation details. Section 5 presents some evaluation results. Section 6 describes the system interface and its functionality, with some examples. Finally, Section 7 concludes and discusses future work.

**2 RELATED WORK**

Many task formulations have been proposed to address the spread of misinformation and disinformation online, and for each formulation, a number of approaches have been tried. Some good readings on the topic include surveys such as that by Shu et al. [59], who adopted a data mining perspective on "fake news" and focused on social media. Another survey [70] studied rumor detection in social media. The survey by Thorne and Vlachos [61] took a fact-checking perspective on "fake news" and related problems.

Li et al. [32] covered truth discovery in general. Lazer et al. [29] offered a general overview and discussion on the science of "fake news", while Vosoughi et al. [65] focused on the process of proliferation of true and false news online. Other recent surveys focused on stance detection [28], propaganda [12], social bots [17], false information [68] and bias on the Web [4]. Some very recent surveys focused on stance for misinformation and disinformation.
makes a prediction about a claim’s factuality and gives sentence-level evidence from Wikipedia to explain its decision. Moreover, our system supports several languages besides English.

3 DATA

We train our system on the FEVER dataset [62], which includes a dump of 5.4M Wikipedia pages, and 220K claims labeled with one of the following three classes: Supports, Refutes or Not Enough Info. An example is shown in Figure 1. For the former two labels, there is also evidence provided, i.e., a sentence or a set of sentences that would allow one to make a verdict about the veracity of the input claim, while the Not Enough Info label indicates that there is no enough evidence in Wikipedia to prove or to refute the claim. There can be different sets of evidence for a given claim, but all of them would support the same label.

Figure 1: Manually labeled claim from FEVER.

4 SYSTEM OVERVIEW

Figure 2 shows the general architecture of our system, which is similar to the one described in [11]. First, the Document Retrieval (DR) module finds potentially relevant documents from Wikipedia. Then, the Sentence Retrieval (SR) module extracts the top-20 most relevant sentences from these documents. Afterwards, the Natural Language Inference (NLI) module classifies each claim–sentence pair as support/refute/NEL. Finally, the aggregation module makes a final prediction. We describe these steps below in more detail.

Unlike these systems, we perform textual evidence-based fact-checking using Wikipedia, following the FEVER formulation of the task. The most related demo, FAKTA, is also trained on data from FEVER, but it focuses on the stance of a document, e.g., retrieved from the Web, with respect to the input claim, while our system
constituency parser [26] to extract noun phrases, which we use as separate queries. We further generate a query from the part of the claim up to the head word, which we extract using a dependency parser [15]. For each query, we retain the top-3 returned documents. As there can be many queries for an input claim, we further filter the results to improve the inference speed. Thus, we process the input claim and all retrieved titles shortened to the first bracket symbol using the Porter Stemmer [49], and we select them for the final set only if they are fully contained in the input.

Sentence Retrieval (SR). This component selects the top-20 most relevant sentences from the documents retrieved by DR component. The relevance is estimated as the cosine between the TF.IDF representations of the claim and of the document titles. We apply a variant of TF.IDF that uses binarization of the non-zero term counts, and considers words with a document fraction higher than 0.85 as stopwords. Note that there are no thresholds for the relevance score, and thus all retrieved sentences can potentially have a score of zero, e.g., if all top-20 documents happen to be irrelevant.

Natural Language Inference (NLI). This component takes as input the source claim, a separator, a sentence retrieved by the SR component, another separator, and the title of the document. Here, we use all retrieved sentences independently. For this component, we use BERT [14], which we fine-tune using a balanced training dataset. In particular, we use instances of the classes Supports and Refutes from the training set, and also the top-3 retrieved sentences for each claim, which we label as Not Enough Info.

Aggregation. We use CatBoost gradient boosting [50] to aggregate the sentence-level predictions of the NLI component. We train the model on part of the validation set using the stacked label probabilities predicted by the NLI model. Thus, we use a 60-dimensional (3 scores for 20 sentences) feature vector, potentially padded by zeros in case of lack of sentences in the retrieval phase.

We further report the maximal probability of the Supports (Refutes) class among all sentences scaled by the percentage of such sentences among the possibly relevant ones, i.e., excluding Not Enough Info, as the confidence score for Truth (Lying) predictions and the minimal probability of the Not Enough Info, otherwise.

Overall Architecture. WhatTheWikiFact has a server and a client parts, which are connected via a REST API. The client part is implemented using the Streamlit framework\(^3\)—it is the GUI part, that is, the interface in the browser. The server part uses the Fast API\(^4\). We use Allen NLP library [19] for the constituency and the dependency parsers for texts in English, and Natasha\(^5\) and Stanza\(^6\) libraries for texts in Russian and Bulgarian, respectively. We use the official repository for BERT.\(^7\) We preload these models on the server, and we serve them using POST requests made by the client.

The parsers receive a piece of text as input, which is then tokenized and stemmed using the NLTK library \(^7\) for English and Russian, and using BulStem [36, 37] for Bulgarian. BERT receives a list of sentence pairs, preprocessed using the BERT tokenizer.

---

\(^3\)http://streamlit.io/
\(^4\)http://fastapi.tiangolo.com/
\(^5\)http://github.com/natasha/natasha
\(^6\)http://stanfordnlp.github.io/stanza/
\(^7\)http://github.com/google-research/bert

Our NLI component is English-only, and thus we use the Google Translator API\(^8\) to translate the input into English first.

5 EVALUATION

The machine learning model we use in our WhatTheWikiFact system achieves an accuracy of 73.22 and a FEVER score of 67.44 on the test part of the FEVER dataset. Note that these scores are better than those for the best system at the FEVER shared task [63], which had an accuracy of 68.21, and a FEVER score of 64.21. They are also on par with the best system from the builder phase of the FEVER2.0 shared task [64], which had a FEVER score of 68.46. We should note that there have been some better results reported in the literature since then. In fact, we also had stronger results in our offline experiments. However, we use this particular model, as we need real-time execution, which requires certain compromise in terms of accuracy for the sake of improved speed of execution, which is essential for a real-time system like ours.

Our analysis shows that the biggest fraction of the system classification errors are in distinguishing Not Enough Info sentences from the rest. At the same time, the accuracy of the retrieval component is almost 91%, which means that in 91% of the cases, we do retrieve correct evidence for the final set of potentially relevant sentences. Therefore, as the system classification errors are not very frequent, such cases can be easily analyzed by the user in our system’s output.

6 USER INTERFACE

Figure 3 shows a snapshot of WhatTheWikiFact’s output for an input claim. We can see that the system offers an overview of the verification results, which includes a verdict, the system’s confidence in that verdict, and a list of possibly relevant documents: title and a link to the corresponding Wikipedia page.

Figure 3 shows that the system has retrieved three articles for the input claim “Napoleon Bonaparte declared Joan of Arc a national symbol.” However, this does not mean that the system considers all of these articles as relevant. It just means that they contain some of the top-20 most relevant sentences.

By clicking on Show info, the user can expand the panel with details about each document. As a result, the following information will be shown (illustrated on Figure 4):

- The part of the input claim used to retrieve the document.
- A bar chart of the predicted stance labels for the input claim with respect to each retrieved sentence. The stance is expressed as one of the classes Supports (SUP), Refutes (REF), or Not Relevant Info (NEI). The chart further shows the class probability, which is also represented as the bar height, sentence number, and label, which is also indicated with the corresponding color. Note that there are three bars for each sentence, i.e., one for each label. Moreover, the bars are ordered (grouped) by labels or optionally by sentences (it is specified in the Output options section).
- A table of the most relevant sentences. For each sentence in that table, we show its position in the document as well as the document length, thus reflecting also the relative position and allowing for easy matching with the bar chart.

---

\(^8\)https://github.com/midhahoff/deep-translator
WhatTheWikiFact: Fact-Checking Claims Against Wikipedia

This tool performs fact checking by comparing information with Wikipedia data.

Select the language

Enter claims here (one per row, max 5)

Napoleon Bonaparte declared Joan of Arc a national symbol.

Output options

Verify

Retrieved articles:

Joan of Arc

Napoleon

Verdict

Confidence

Retrieved articles:

Joan of Arc

Napoleon

Figure 3: Screenshot of what our WhatTheWikiFact system outputs when verifying the claim: “Napoleon Bonaparte declared Joan of Arc a national symbol.”

For example, Figure 4 shows that the article “Joan of Arc” is relevant for fact-checking the input claim, as it includes a sentence that supports it. Other retrieved sentences in this document have almost 100% probability of a Not Enough Info label, and are thus irrelevant. Manual analysis by the user for the remaining two documents—“Napoleon”, which was retrieved by the phrase “Napoleon Bonaparte”, and “National symbol”, which was retrieved by the phrase “a national symbol”—could confirm that they are indeed irrelevant. With these retrieval results, the user can manually inspect the boundary classification cases, especially for veracity prediction with low confidence or in case of Not Enough Info.

7 CONCLUSION AND FUTURE WORK

We have presented WhatTheWikiFact, a system for automatic claim verification using Wikipedia. The system reports the veracity for each input claim supplemented with evidence retrieved during the verification process, thus offering explainability. It also shows confidence scores and a set of relevant Wikipedia articles. Moreover, it allows the user to obtain detailed information about each article, including the exact phrase that was used to retrieve it, a list of the most relevant sentences with to the input claim that it contains, and their stance probabilities regarding the input claim. The system supports several languages: Bulgarian, English, and Russian.

In future work, we plan to implement a more accurate model by distilling knowledge from a larger model. Another direction we want to explore is to add additional languages for verification using local Wikipedias, which can be implemented without additional training, e.g., using multilingual BERT or language adapters [46].

ACKNOWLEDGMENTS

This research was done within the framework of the HSE University Basic Research Program.

Figure 4: Screenshot showing details about “Joan of Arc”, a Wikipedia article retrieved when verifying the claim “Napoleon Bonaparte declared Joan of Arc a national symbol.”

It is also part of the Tanbih mega-project (http://tanhib.qcri.org/), which is developed at the Qatar Computing Research Institute, HBKU, and aims to limit the impact of “fake news,” propaganda, and media bias by making users aware of what they are reading.
REFERENCES

[1] Firoj Alam, Stefano Cresci, Tannoy Chakraborty, Fabrizio Silvestri, Dimitar Dimitrov, Giovanni Da San Martino, Shaden Shaar, Hamed Firooz, and Preslav Nakov. 2021. A Survey on Multimodal Disinformation Detection. arXiv:2105.12541 (2021).

[2] Aimée Alonso-Reina, Robert Sepulveda-Torres, Estela Saquete, and Manuel Palomar. 2019. Team GLESI. Approach for automated fact checking. In Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER ’19). Association for Computational Linguistics, Hong Kong, China, 110–114.

[3] Rami Aly, Zhijiang Guo, Michael Sejr Schlüchtermann, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021. FEVEROUS: Fact Extraction and VERification Over Unstructured and Structured information. In Proceedings of the 15th Conference on Neural Information Processing Systems Datasets and Benchmarks Track (NeurIPS ’21).

[4] Ricardo Baere-Yates. 2018. Bias on the Web. Commun. ACM 61, 6 (2018), 54–61.

[5] Alberto Barrón-Cedeño, Tamer Elsayed, Preslav Nakov, Giovanni Da San Martino, Maram Hasanain, Reem Suwaileh, Fatima Haouari, Nikolay Babulkov, Bayan Hamdan, Alex Nikolov, Shaden Shaar, and Zien Sheikh Ali. 2020. Overview of CheckThat! 2020 – Automatic Identification and Verification of Claims in Social Media. In Proceedings of the 11th International Conference of the CLEF Association: Experimental IR Meets Multilinguality, Multimodality, and Interaction (CLEF ’20). Springer, Thessaloniki, Greece, 215–236.

[6] Alberto Barrón-Cedeño, Tamer Elsayed, Preslav Nakov, Giovanni Da San Martino, Maram Hasanain, Reem Suwaileh, and Fatima Haouari. 2020. CheckThat! at CLEF 2020: Enabling the Automatic Identification and Verification of Claims in Social Media. In Proceedings of the 42nd European Conference on Information Retrieval (ECIR ’20). Springer, Lisbon, Portugal, 499–507.

[7] Stefan Bird and Erez Liper. 2004. NLTK: The Natural Language Toolkit. In Proceedings of the ACL Interactive Poster and Demonstration Sessions (ACL ’04). Association for Computational Linguistics, Barcelona, Spain, 214–217.

[8] Carlos Castillo, Marcelo Mendoza, and Barbara Bobele. 2011. Information Credibility on Twitter. In Proceedings of the 28th International Conference on World Wide Web (WWW ’11). Association for Computing Machinery, Hyderabad, India, 675–684.

[9] Tuhin Chakrabortty, Tarig Alhindi, and Sarmadah Muresan. 2018. Robust Document Retrieval and Indicial Evidence Modeling for Fact Extraction and Verification. In Proceedings of the First Workshop on Fact Extraction and VERification (FEVER ’18). Association for Computational Linguistics, Brussels, Belgium, 127–131.

[10] Wenwu Chen, Hongmin Wang, Jianhui Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2020. TabFact: A large-scale Dataset for Table-based Fact Verification. In Proceedings of the 8th International Conference on Learning Representations (ICLR ’20). OpenReview.net, Addis Ababa, Ethiopia.

[11] Anton Chernyavsky and Dmitry Ilovsky. 2019. Extract and Aggregate: A Novel Domain-Independent Approach to Factual Data Verification. In Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER). Association for Computational Linguistics, Hong Kong, China, 69–74.

[12] Giovanni Da San Martino, Stefano Cresci, Alberto Barrón-Cedeño, Seunghak Zubiaga, Roberto Di Pietro, and Preslav Nakov. 2020. A Survey on Computational Propagation Detection. In Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI-2018) ’20 International Joint Conferences on Artificial Intelligence (IJAIS). International Joint Conferences on Artificial Intelligence (IJAIS), Sapporo (Japan) (online), 4826–4832.

[13] Leon Derczynski, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021. CheckThat! at CLEF 2020: Conference and Labs of the Evaluation Forum (CLEF 2020). CEUR-WS.org, Thessaloniki, Greece.

[14] Christopher Fixley and Monia Diab. 2018. Team SWEEPER: Joint Sentence Extraction and Fact Checking with Pointer Networks. In Proceedings of the First Workshop on Fact Extraction and VERification (FEVER ’18). Association for Computational Linguistics, Brussels, Belgium, 150–155.

[15] Vidur Joshi, Matthew Peters, and Mark Hopkins. 2018. Extending a Farser to Distinct Domains Using a Few Dozen Partially Annotated Examples. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL ’18). Association for Computational Linguistics, Melbourne, Australia, 1190–1199.

[16] Georgios Karagiannis, Mohammed Saeed, Paolo Papotti, and Immanuel Trummer. 2020. Scrutinizer: A Mixed-Initiative Approach to Large-Scale, Data-Driven Claim Verification. Proc. VLDB Endow. 13, 12 (2020), 2508–2521.

[17] Dilek Kicik and Fazli Can. 2020. Stance Detection: A Survey. ACM Comput. Surv. 53, 1, Article 12 (2020), 52 pages.

[18] David M.J. Lazer, Matthew A. Baum, Yochai Benkler, Adam J. Berinsky, Kelly M. Greenhill, Filippo Menczer, Miriam J. Metzger, Brendan Nyhan, Gordon Pennycook, David Rothschild, Michael Schudson, Steven A. Sloman, Cass R. Sunstein, Emily A. Thornton, Duncan J. Watts, and Jonathan L. Zittrain. 2018. The science of fake news. Science 359, 6380 (2018), 1094–1096.

[19] Nayeon Lee, Jeun Bang, Andrea Madotto, and Pascale Fung. 2021. Towards Few-shot Fact-Checking via Perplexity: In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT ’21). Association for Computational Linguistics, Online, 1971–1981.

[20] Nayeon Lee, Belinda Li, Sinong Wang, Wen-tau Yih, Hao Ma, and Madian Khasha. 2020. Language Models as Fact Checkers? In Proceedings of the Third Workshop on Fact Extraction and VERification (FEVER ’20). Association for Computational Linguistics, Online, 36–41.

[21] Yalang Li, Jing Gao, Chushi Meng, Qi Li, Lu Su, Bo Zhao, Wei Fan, and Jiawei Han. 2016. A Survey on Truth Discovery. SIGKDD Explor. Newsl. 17, 2 (2016), 1–16.

[22] Christopher Malon. 2018. Team Papelo: Transformer Networks at FEVER. In Proceedings of the First Workshop on Fact Extraction and VERification (FEVER ’18). Association for Computational Linguistics, Brussels, Belgium, 109–113.

[23] Tsvetomila Mihaylova, Georgi Karadzhov, Pepa Atanasova, Ramy Baly, Mitra Mohtarami, and James Glass. 2019. FAKTA: An Automatic End-to-End Fact Checking System. In Proceedings of the 3rd International Conference on Computational Tools for Natural Language Processing (CoNLL ’19). Association for Computational Linguistics, Minneapolis, Minnesota, USA, 860–869.

[24] Moin Nadeem, Wei Fang, Brian Xu, Mitra Mohtarami, and James Glass. 2019. FAKTA: An Automatic End-to-End Fact Checking System. In Proceedings of the 35th Conference on Neural Information Processing Systems (NIPS ’19). Association for Computational Linguistics, Minneapolis, Minnesota, USA, 3516–3520.

[25] Preslav Nakov. 2003. BulStem: Design and Evaluation of Inflectional Stemmer. In Proceedings of the 4th International Conference on Computer Systems and Technologies (CST 2003). Sofia, Bulgaria, 419–424.

[26] Preslav Nakov. 2003. BulStem: Design and Evaluation of Inflectional Stemmer. In Proceedings of the Workshop on Balkan Language Resources and Tools. Thessaloniki, Greece.
