WikiFactFind: Semi-automated fact-checking based on Wikipedia

Mykola Trokhymovych
Ukrainian Catholic University
Ukraine
trokhymovych@ucu.edu.ua

Diego Saez-Trumper
Wikimedia Foundation
Spain
diego@wikimedia.org

ABSTRACT

Fact verification has become an essential task, being used in various areas from checking auto-generated content to fighting disinformation in hybrid wars. However, even though there has been relevant advances in creating automatic fact-checking systems, nowadays manual work is crucial to deliver good quality results. The manual fact verification usually consists of information retrieval, and logical reasoning for making the final verdict. In this work we concentrate on the process of searching for the fact proofs, reveal possible problems while searching, and propose a tool that helps increase the speed of fact checking without sacrificing its quality.

CCS CONCEPTS

• **Information systems → Expert search**; **Learning to rank**.

KEYWORDS

Wikipedia, search, fact-checking, NLP, applied research

ACM Reference Format:
Mykola Trokhymovych and Diego Saez-Trumper. 2022. WikiFactFind: Semi-automated fact-checking based on Wikipedia. In *WikiWorkshop2022.* ACM, New York, NY, USA, 5 pages.

1 INTRODUCTION

The rapid growth of social networks and various media also increases fake news and disinformation. It leads to demand in services and communities for labeling and filtering misleading facts like Facebook's Third-Party Fact-Checking Program[1] or Birdwatch[2] by Twitter. Community effort aims to disclose misinformation and reduce its harmful impact on society. Although Artificial Intelligence (AI) community is trying to fight against false facts by creating Automated Fact-Checking Systems (AFCS) [6, 15], fact verification is usually conducted manually nowadays.

Depending on the complexity of a given claim, its verification can take from several minutes to hours. Automation can help to reduce dramatically the time to "stick" in people's minds [1]. However, even state-of-the-art fully automated solutions show the accuracy of about 75%, which is far from desired human-level performance [9]. One possible solution is to involve the human in the process, providing extended assistance and hints.

Manual fact-checking (*a.k.a manual fact verification*) always involves information retrieval through open, reliable knowledge sources. Search is often performed through search engines like Google or inside specific knowledge sources like Wikipedia. The search engine result page (SERP) is further analyzed to find evidence for the correctness of the initial claim. Automatic and precise retrieval of relevant data may help fact-checkers save much time. At the same time, the quality of pages provided in SERP is also essential for manual fact-checker to make the final verdict and should be considered.

In this work, we analyze possible problems that occur while manual fact-checking. Mainly, we concentrate our efforts on the stage of Information retrieval. We analyze different manual search strategies (MSS) for finding desired evidence pages. We define MSS as a method that can be applied to find desired information through search engines. They include but are not limited to query processing, construction, and augmentation.

Initially, we state the following research questions that will be tackled:

- **RQ1**: How do manual search strategies impact the fact-checking process?
- **RQ2**: Does the claim label influence the results of evidence search?
- **RQ3**: What is the relation of Wikipedia article quality and SERP results?

As a result, we contribute to the community of fact-checkers with answering research question stated before. We propose an optimal strategy for "query building" for manual fact checking. Finally, we build initial semi-automated procedure that reduces the time of manual work while searching for fact checking and increases its accuracy using machine learning algorithm for reranking.

2 RELATED WORK

Misinformation on the social media and internet led to much research fighting false facts. One line of research is trying to create fully automated solution that aim to replace manual work [2, 7, 15]. However, these approach have crucial problems like dependence on specific knowledge base or unacceptable accuracy. At the same time, even if full automation remains unreachable, tools that support fact-checkers in their manual work are to be welcomed [10].

Automated solutions usually match each of the manual stages with the technical tasks that can be automated and performed by machines (Table 1). That was well presented during the FEVER shared task competition, where teams compete in developing an end-to-end fact-checking system using the FEVER dataset based on Wikipedia knowledge base [13].

In our work, we concentrate on the initial stage of fact checking - evidence retrieval. Several relevant works were done as a solutions

[1] Facebook's Third-Party Fact-Checking Program https://www.facebook.com/journalismproject/programs/third-party-fact-checking.
[2] Birdwatch https://twitter.github.io/birdwatch/about/overview/.

WikiWorkshop '22, June 03–05, 2018, Lyon, France

Table 1: Mapping between manual and automated fact checking steps and their description.

Manual fact-checking stage	Technical tasks	Description
Construct appropriate questions	Information retrieval	Given the claim, process it and search for potential candidates to be evidence.
Obtain the answers from relevant sources	Natural language inference	Evaluate the relationship between the claim and evidence candidates.
Reach a verdict using these answers	Aggregation. Ranking. Classification	Aggregate results and provide final verdict and its interpretation.

for the FEVER Shared task [14]. Along with a baseline system presented by [12], most of the solutions are multistage models that perform document retrieval, sentence selection, and sentence classification. Baseline exploits the basic TF-IDF-based retrieval to find the relevant evidence. The UNC-NLP solution achieves 0.64 FEVER score comparing to 0.28 of baseline [11, 12]. It uses neural models to perform deep semantic matching for both document and sentence retrieval stages. UCL MRG team proposes to use logistic regression for the document and sentences retrieval [18]. Athene team use entity linking and WikiMedia search API for article search [4]. Such models were build to use specifically on FEVER dataset and use different variant of text matching.

At the same time, there are works showing that search for fact checking differs from regular web search. Claim-document relation is usually not enough, and a set of other factors influence the success of evidence retrieval. Wang *et al.* build multistage facts search system, that uses also different features including text similarity and publication timestamp in order to classify whether a related document is relevant to the initial claim [16]. Hasanain *et al.* also research on retrieving pages useful for fact checking, called evidential pages. Their paper shows that retrieving evidential pages is weakly correlated with regular retrieval optimized by search engines [5]. It also shows that there are linguistic cues that can help predict page usefulness like length of sentence, presence of named entities or quotes, etc. In our work we will observe how documents features like quality score influence retrieving evidential pages.

3 DATASET PREPARATION

We use the FEVER dataset as the main benchmark for the task of fact-checking. It includes the claims that should be verified along with lists of evidence in links to sentences in articles from the Wikipedia dump dated June 2017. Initially, the dataset consists of 185,445 claims labeled with SUPPORTS (S), REFUTES (R), or NOT ENOUGH INFO (NEI) classes. As we are concentrating on search, we consider filtering out NEI class samples, as they do not include any links to articles, so they cannot be used to validate the search. Finally, we got a dataset with 123142 R, or S labeled samples. The distribution of classes within the train and test parts are presented in Figure 1. We used a predefined FEVER split, and the distributions differ between train and test so that the testset is balanced.

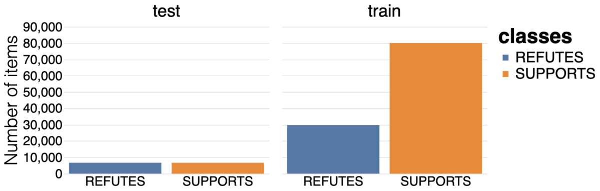

Figure 1: Distribution of labels within train and test

One more issue we faced was changing the names of articles in time. In further experiments, we plan to use search services that work on an up-to-date version of Wikipedia. It may result in the situation when we find the correct article, but with a changed name. It means that the name found does not match the corresponding one in the dataset, which leads to misinterpretation of results. So, we created a mapping from old to new names. The filtered dataset included 14533 unique articles presented in the evidence. Our investigation showed that 1082 (7.4%) of them have changed the name in the period after FEVER creation before now.

4 MANUAL EVIDENCE SEARCH

This section observes different strategies the manual fact-checker can use to find the evidence for the given claim. Also, we analyze how the quality of pages and labels influence search results. We use prepared dataset (presented in Sec. 3) for validation.

We observed three main characteristics: *(i)* Rate of found items (RFI), which is equal to Recall@10; *(ii)* Rate of correctly placed items on first position (taking into account only those searches, where correct evidence article was found); *(iii)* distribution of desired evidence on top-10 positions of SERP. RFI shows the ability to find the correct evidence page using search results. The second and third characteristics help understand if correct items appear earlier than not-useful ones. We consider only top-10 results from a search, as the further results have a low probability of being observed while manual search [8]. It is crucial to show correct items earlier as it increases their chance of being observed by a manual fact-checker.

4.1 Using raw Wikipedia search

The experiment represents the basic logic when the whole claim is passed to Wikimedia API without any changes. It copies the manual search through Wikipedia using a built-in search engine. Such an approach is easy to perform as it does not require additional logical reasoning. We are applying such a strategy to the prepared FEVER dataset. We finalized the Rate of found items of 0.539 for the test and 0.681 for the train part. RCPI is 0.773 for the test and 0.705 for the train part. The distribution of evidence position in SERP is shown in Figure 2. One more important observation is that metrics significantly differ for test and train parts so that RFI for train part is larger than for test.

4.2 Using Wikipedia search for named entities

Wikipedia is an encyclopedia representing articles about specific entities. The possible way to find appropriate information is by

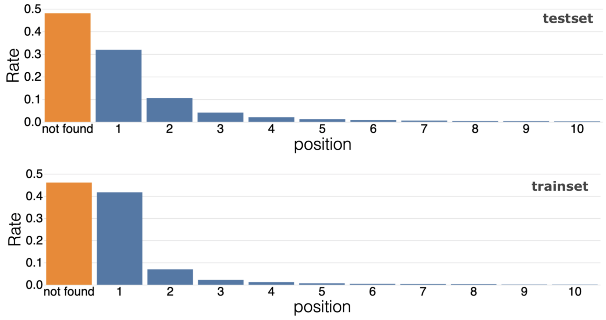

Figure 2: Search without query modification. Position of true items in SERP.

searching for those entities. We experimented with query modification. As for that experiment, we parsed named entities from the initial claim and passed them independently to the search. We used the best performing strategy presented during previous research, which is based on Flair[3] ner-fast model for named entities extraction getting top three search results for each [15]. It is essential to mention that the proposed approach may use more than one query in the case of multiple named entities presented in the initial claim. The final results list consists of mixed results sorted by rank from each query.

Such a strategy shows significant improvement compared to the previous one (presented in 4.1). It resulted in an RFI of 0.827 for the test and 0.885 for test parts. RCPI for such an experiment is 0.847 for test and 0.87 for train parts. The distribution of evidence position in SERP is shown in Figure 3. Metrics for test and train parts differ less. According to RCPI, the evidence appears after the first position in about 15% of cases, so possible re-ranking is needed.

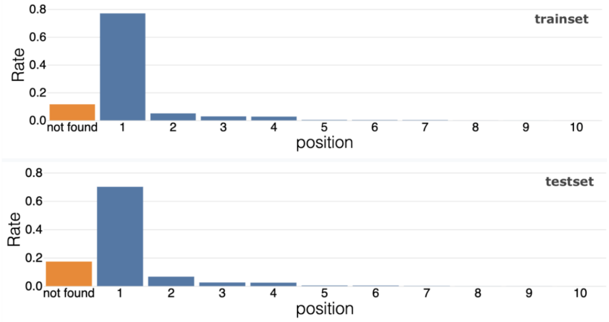

Figure 3: Search with query modification. Position of true items in SERP.

4.3 Using Google search engine

Evidence search can also be performed through search engines like Google, so we experimented with it. As we use the validation dataset that includes only evidence from Wikipedia, we applied the filter to search only through the English version of Wikipedia for experiment fairness. We used random search agents and open proxy servers to avoid banning from the Google side. As for search query, we used the entire claim to experiment with how regular search engine deals with it. As a result, we were able to find 84.3% of true evidence pages. We got 0.749 RCPI and the distribution of found items presented in Figure 4.

That is important to mention that we used a random 10% sample of the initial dataset, as automated search through Google is costly and time-consuming. As a result, we got comparable RFI to the experiment presented in 4.2. However, the RCPI appears to be lower, which means that relevant items appear on the first, the most observable position, rarer. At the same time, we should add that the whole raw claim was used for search, which means that no additional effort is needed to perform a search.

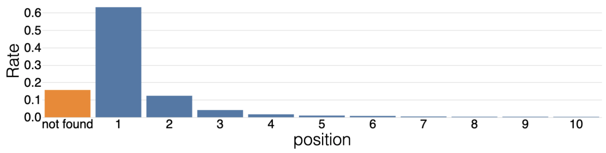

Figure 4: Google search without query modification. Position of true items in SERP.

4.4 Comparing performance for different labels

In order to answer **RQ2**, we analyzed the results of a search for different labels for each of the proposed strategies using the FEVER dataset. We assume that there is a difference in checking for correct and incorrect statements, especially on the search stage. The results of our investigation are presented in Figure 5.

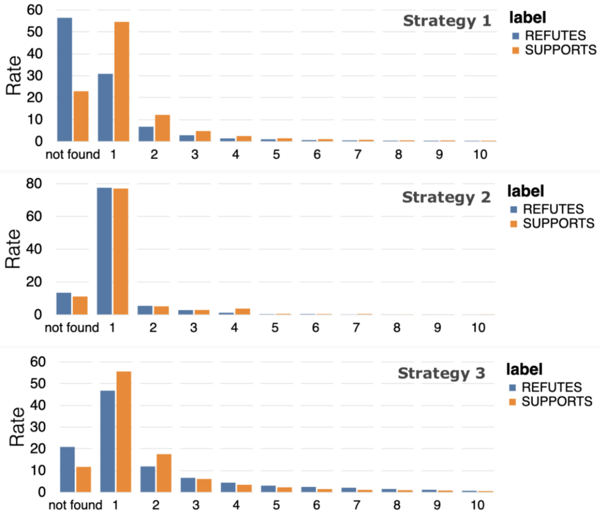

Figure 5: Position of true items in SERP for trainset: *(i)* Wikipedia search without query modification; *(ii)* Wikipedia search with query modification; *(iii)* Google search without query modification.

[3] https://github.com/flairNLP/flair

We concluded that for strategies without query modification, the results for **R** class are much worse than for **S** class. For example, as for Wikipedia search without query modification, we got RFI of only 0.437 for **R** compared to 0.772 for **S** class. It concludes that searching for evidence to disprove facts might be more difficult. On the other hand, a search strategy that uses query modification is free from such bias. The difference in RFI for **S** and **R** classes for such strategy is only 0.023 comparing to 0.335 for *Strategy 1*.

4.5 Influence of evidence quality

The quality of Wikipedia articles is one of the core concepts for the encyclopedia. At the same time, the evaluation process requires much manual work. Also, most articles are constantly updated, so it is impossible to measure the quality manually. However, the Objective Revision Evaluation Service (ORES)[4] was developed by the Wikimedia Machine Learning team. It is a machine learning tool that can automatically evaluate the quality of the page and edits.

In this subsection, we aim to answer **RQ3**. As for the experiment, we use ORES API scores to evaluate the quality of each possible evidence page that appears in SERP. We calculate the scores for specific page revision that was up-to-date for the time of Wikipedia dump used in FEVER.

As for our research we are using an article quality model named **WP10** [17]. **WP10** is a multi-label classification model aims to allocate article to one of the quality classes of *FA*, *GA*, *B*, *C*, *Start*, *Stub*, where *FA* stands for Featured articles (the best articles Wikipedia has to offer)[5].

We analyzed the WP10 label across SERP position distribution. The results of our investigation are presented in Figure 6. As a result, we found out that the most frequent class presented on the first three positions is *C*, but is replaced by *B* on the next positions. The general observation is that an increase in position also increases the chance to observe higher quality articles. At the same time, it should be mentioned that such an experiment is highly dependent on the FEVER dataset and probably should be observed in more detail in further research.

Also, one more limitation that should be mentioned is that strategy we use for such a search experiment gets the first three search results for each entity from the claim. When only one entity is found in the claim, we have only three results. So, we have fewer items for the fourth and fifth positions in general distribution. That can be a possible reason for the difference between the first three vs. fourth and fifth search results quality distributions. Further research should test this result with more data and other search strategies.

5 TOOL FOR EFFICIENT FACTS SEARCH

In the previous sections, we observed different search strategies and page quality features relations with search results. We concluded that the best search strategy was presented in Section 4.2. We will use it for the experiment presented in this section. Also, we observed a relation between articles' quality features and position in SERP. As a result we aimed to create a tool that implements the best strategy search and adapt it to fact checking domain by training re-ranking model.

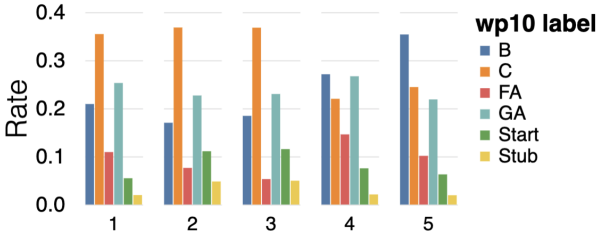

Figure 6: WP10 label across SERP position distribution

The approach we propose is presented in Figure 7. The basic idea is to use the actual search results, enhance the data with ORES features, and train Learning-to-rank (LTR) model. As for the LTR model, we use Catboost with YetiRankPairwise loss for training model [3]. We fit the model using a predefined FEVER trainset and evaluate the test. As the primary metric for evaluation, we use RCPI (Recall@1). We are training a model with default parameters with 100 iterations.

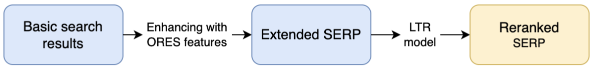

Figure 7: Re-ranking approach schema

As a result, we increased Recall@1 from 0.847 to 0.875 with the basic model. It shows that such an approach can improve search results and fact-checking domain adaptation.

6 CONCLUSIONS

During this research, we were working on fact-checking domain-specific searches. We processed the FEVER dataset to be used for search evaluation. Using this data, we tested three strategies of performing the evidence search that can be used in a manual process. While answering *RQ1*, we found out that strategy selection has a significant impact on recall of search. Also, we discovered that even for the best performing strategy, about 15% of correct results appear in the non-first position. It reduces the chance of being observed.

Also, we observed how the page quality differs across positions in search, answering *RQ3*. We found out that there is a difference in distributions of pages' quality across positions in search results. We assume that such relations can be used to train models to re-rank search results.

Finally, we build basic learning to rank model that shows that using page quality features can increase the recall for first positions. Consequently, it may increase the chance of correct evidence being observed. At the same time, we should add that the presented model is an initial one, and more research is needed to make it more precise.

7 DISCUSSION AND LIMITATIONS

The main limitation of this work is that we use Wikipedia as the only source for evidence. However, the only ground truth usually

[4] https://www.mediawiki.org/wiki/ORES
[5] https://en.wikipedia.org/wiki/Wikipedia:Featured_articles

does not exist, so there is a need to work with heterogeneous data. One more limitation is that we tested only several search strategies, and this list should be extended.

One more critical observation found during answering *RQ2* that searching for sources to refute incorrect claims can be more complicated than looking for correct statement evidence. On the other hand, the strategy with query processing may reduce that effect by searching for mentioned named entities instead of using the whole claim as a query.

REFERENCES

[1] Sylvie Cazalens, Philippe Lamarre, Julien Leblay, Ioana Manolescu, and Xavier Tannier. 2018. A Content Management Perspective on Fact-Checking. In *Companion Proceedings of the The Web Conference 2018* (Lyon, France) *(WWW '18)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 565–574. https://doi.org/10.1145/3184558.3188727

[2] Anton Chernyavskiy, Dmitry Ilvovsky, and Preslav Nakov. 2021. WhatTheWikiFact: Fact-Checking Claims Against Wikipedia. *CoRR* abs/2105.00826 (2021). arXiv:2105.00826 https://arxiv.org/abs/2105.00826

[3] Andrey Gulin, Igor Kuralenok, and Dimitry Pavlov. 2011. Winning The Transfer Learning Track of Yahoo!'s Learning To Rank Challenge with YetiRank. In *Proceedings of the Learning to Rank Challenge (Proceedings of Machine Learning Research, Vol. 14)*, Olivier Chapelle, Yi Chang, and Tie-Yan Liu (Eds.). PMLR, Haifa, Israel, 63–76. https://proceedings.mlr.press/v14/gulin11a.html

[4] 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)*. Association for Computational Linguistics, Brussels, Belgium, 103–108. https://doi.org/10.18653/v1/W18-5516

[5] Maram Hasanain and Tamer Elsayed. [n.d.]. Studying effectiveness of Web search for fact checking. *Journal of the Association for Information Science and Technology* n/a, n/a ([n. d.]). https://doi.org/10.1002/asi.24577 arXiv:https://asistdl.onlinelibrary.wiley.com/doi/pdf/10.1002/asi.24577

[6] Naeemul Hassan, Anil Nayak, Vikas Sable, Chengkai Li, Mark Tremayne, Gensheng Zhang, Fatma Arslan, Josue Caraballo, Damian Jimenez, Siddhant Gawsane, Shohedul Hasan, Minumol Joseph, and Aaditya Kulkarni. 2017. ClaimBuster: the first-ever end-to-end fact-checking system. *Proceedings of the VLDB Endowment* 10 (08 2017), 1945–1948. https://doi.org/10.14778/3137765.3137815

[7] Naeemul Hassan, Anil Nayak, Vikas Sable, Chengkai Li, Mark Tremayne, Gensheng Zhang, Fatma Arslan, Josue Caraballo, Damian Jimenez, Siddhant Gawsane, Shohedul Hasan, Minumol Joseph, and Aaditya Kulkarni. 2017. ClaimBuster: the first-ever end-to-end fact-checking system. *Proceedings of the VLDB Endowment* 10 (08 2017), 1945–1948. https://doi.org/10.14778/3137765.3137815

[8] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. 2017. Accurately Interpreting Clickthrough Data as Implicit Feedback. *SIGIR Forum* 51, 1 (aug 2017), 4–11. https://doi.org/10.1145/3130332.3130334

[9] Zhenghao Liu, Chenyan Xiong, and Maosong Sun. 2019. Kernel Graph Attention Network for Fact Verification. *CoRR* abs/1910.09796 (2019). arXiv:1910.09796 http://arxiv.org/abs/1910.09796

[10] Preslav Nakov, David Corney, Maram Hasanain, Firoj Alam, Tamer Elsayed, Alberto Barrón-Cedeño, Paolo Papotti, Shaden Shaar, and Giovanni Martino. 2021. Automated Fact-Checking for Assisting Human Fact-Checkers.

[11] Yixin Nie, Haonan Chen, and Mohit Bansal. 2018. Combining Fact Extraction and Verification with Neural Semantic Matching Networks. arXiv:1811.07039 [cs.CL]

[12] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for Fact Extraction and VERification. arXiv:1803.05355 [cs.CL]

[13] James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018. The Fact Extraction and VERification (FEVER) Shared Task. *CoRR* abs/1811.10971 (2018). arXiv:1811.10971 http://arxiv.org/abs/1811.10971

[14] James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018. The Fact Extraction and VERification (FEVER) Shared Task. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*. Association for Computational Linguistics, Brussels, Belgium, 1–9. https://doi.org/10.18653/v1/W18-5501

[15] Mykola Trokhymovych and Diego Sáez-Trumper. 2021. WikiCheck: An end-to-end open source Automatic Fact-Checking API based on Wikipedia. *CoRR* abs/2109.00835 (2021). arXiv:2109.00835 https://arxiv.org/abs/2109.00835

[16] Xuezhi Wang, Cong Yu, Simon Baumgartner, and Flip Korn. 2018. Relevant Document Discovery for Fact-Checking Articles. In *Companion Proceedings of the The Web Conference 2018* (Lyon, France) *(WWW '18)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 525–533. https://doi.org/10.1145/3184558.3188723

[17] Morten Warncke-Wang, Vladislav R. Ayukaev, Brent Hecht, and Loren G. Terveen. 2015. The Success and Failure of Quality Improvement Projects in Peer Production Communities. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work amp; Social Computing* (Vancouver, BC, Canada) *(CSCW '15)*. Association for Computing Machinery, New York, NY, USA, 743–756. https://doi.org/10.1145/2675133.2675241

[18] 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)*. Association for Computational Linguistics, Brussels, Belgium, 97–102. https://doi.org/10.18653/v1/W18-5515