FAKTA: An Automatic End-to-End Fact Checking System

Moin Nadeem, Wei Fang, Brian Xu, Mitra Mohtarami, James Glass
MIT Computer Science and Artificial Intelligence Laboratory
Cambridge, MA, USA
{mnadeem, weifang, bwxu, mitram, glass}@mit.edu

Abstract

We present FAKTA which is a unified framework that integrates various components of a fact checking process: document retrieval from media sources with various types of reliability, stance detection of documents with respect to given claims, evidence extraction, and linguistic analysis. FAKTA predicts the factuality of given claims and provides evidence at the document and sentence level to explain its predictions.

1 Introduction

With the rapid increase of fake news in social media and its negative influence on people and public opinion (Mihaylov et al., 2015; Mihaylov and Nakov, 2016; Vosoughi et al., 2018), various organizations are now performing *manual* fact checking on suspicious claims. However, manual fact-checking is a time consuming and challenging process. As an alternative, researchers are investigating *automatic* fact checking which is a multi-step process and involves: (*i*) retrieving potentially relevant documents for a given claim (Mihaylova et al., 2018; Karadzhov et al., 2017), (*ii*) checking the reliability of the media sources from which documents are retrieved, (*iii*) predicting the stance of each document with respect to the claim (Mohtarami et al., 2018; Xu et al., 2018), and finally (*iv*) predicting factuality of given claims (Mihaylova et al., 2018). While previous works separately investigated individual components of the fact checking process, in this work, we present a unified framework titled FAKTA that integrates these components to not only predict the factuality of given claims, but also provide evidence at the document and sentence level to explain its predictions. To the best of our knowledge, FAKTA is the only system that offers such a capability.

2 FAKTA

Figure 1 illustrates the general architecture of FAKTA. The system is accessible via a Web browser and has two sides: client and server. When a user at the client side submits a textual claim for fact checking, the server handles the request by first passing it into the document retrieval component to retrieve a list of top-K relevant documents (see Section 2.1) from four types of sources: Wikipedia, highly-reliable, mixed reliability and low reliability mainstream media (see Section 2.2). The retrieved documents are passed to the re-ranking model to refine the retrieval result (see Section 2.1). Then, the stance detection component detects the stance/perspective of each relevant document with respect to the claim, typically modeled using labels such as *agree*, *disagree* and *discuss*. This component further provides rationales at the sentence level for explaining model predictions (see Section 2.3). Each document is also passed to the linguistic analysis component to analyze the language of the document using different linguistic lexicons (see Section 2.4). Finally, the aggregation component combines the predictions of stance detection for all the relevant documents and makes a final decision about the factuality of the claim (see Section 2.5). We describe the components below.

2.1 Document Retrieval & Re-ranking Model

We first convert an input claim to a query by only considering its verbs, nouns and adjectives (Potthast et al., 2013). Furthermore, claims often contain named entities (e.g., names of persons and organizations). We use the NLTK package to identify named entities in claims, and augment the initial query with all named entities from the claim's text. Ultimately, we generate queries of 5–10 tokens, which we execute against a search engine. If the search engine does not retrieve any results for

Proceedings of NAACL-HLT 2019: Demonstrations, pages 78–83
Minneapolis, Minnesota, June 2 - June 7, 2019.

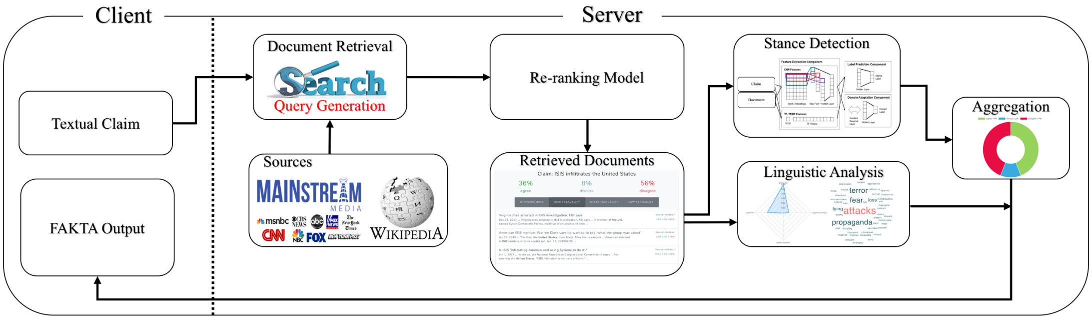

Figure 1: The architecture of our FAKTA system.

the query, we iteratively relax the query by dropping the final tokens one at a time. We also use Apache Lucene[1] to index and retrieve relevant documents from the 2017 Wikipedia dump (see our experiments in Section 3). Furthermore, we use the Google API[2] to search across three pre-defined lists of media sources based on their factuality and reliability as explained in Section 2.2. Finally, the re-ranking model of Lee et al. (2018) is applied to select the top-K relevant documents. This model uses all the POS tags in a claim that carry high discriminating power (NN, NNS, NNP, NNPS, JJ, CD) as keywords. The re-ranking model is defined as follows:

$$f_{rank} = \frac{|match|}{|claim|} \times \frac{|match|}{|title|} \times score_{init}, \quad (1)$$

where $|claim|$, $|title|$, and $|match|$ are the counts of such POS tags in the claim, title of a document, both claim and title respectively, and $score_{init}$ is the initial ranking score computed by Lucene or ranking from Google API.

2.2 Sources

While current search engines (e.g., Google, Bing, Yahoo) retrieve relevant documents for a given query from any media source, we retrieve relevant documents from four types of sources: Wikipedia, and high, mixed and low factual media. Journalists often spend considerable time verifying the reliability of their information sources (Popat et al., 2017; Nguyen et al., 2018), and some fact-checking organizations have been producing lists of unreliable online news sources specified by their journalists. FAKTA utilizes information about news media listed on the Media Bias/Fact Check (MBFC) website[3], which contains manual annotations and analysis of the factuality of $2,500$ news websites. Our list from MBFC includes $1,300$ websites annotated by journalists as *high* or *very high*, 700 websites annotated as *low* and *low-questionable*, and 500 websites annotated as *mixed* (i.e., containing both factually true and false information). Our document retrieval component retrieves documents from these three types of media sources (i.e., *high*, *mixed* and *low*) along with Wikipedia that mostly contains factually-true information.

2.3 Stance Detection & Evidence Extraction

In this work, we use our best model presented in (Xu et al., 2018) for stance detection. To the best of our knowledge, this model is the current state-of-the-art system on the Fake News Challenge (FNC) dataset.[4] Our model combines Bag of Words (BOW) and Convolutional Neural Networks (CNNs) in a two-level *hierarchy* scheme, where the first level predicts whether the label is *related* or *unrelated* (see Figure 2, the top-left pie chart in FAKTA), and then related documents are passed to the second level to determine their stances, *agree*, *disagree*, and *discuss* labels (see Figure 2, the bottom-left pie chart in FAKTA). Our model is further supplemented with an adversarial domain adaptation technique which helps it overcome the limited size of labeled data when training through different domains.

To provide rationales for model prediction, FAKTA further processes each sentence in the document with respect to the claim and computes a stance score for each sentence. The relevant sentences in the document are then highlighted and color coded with respect to stance labels (see Figure 2). FAKTA provides the option for re-ordering these rationales according to a specific stance label.

[1] https://lucene.apache.org
[2] https://developers.google.com/custom-search
[3] https://mediabiasfactcheck.com
[4] http://www.fakenewschallenge.org

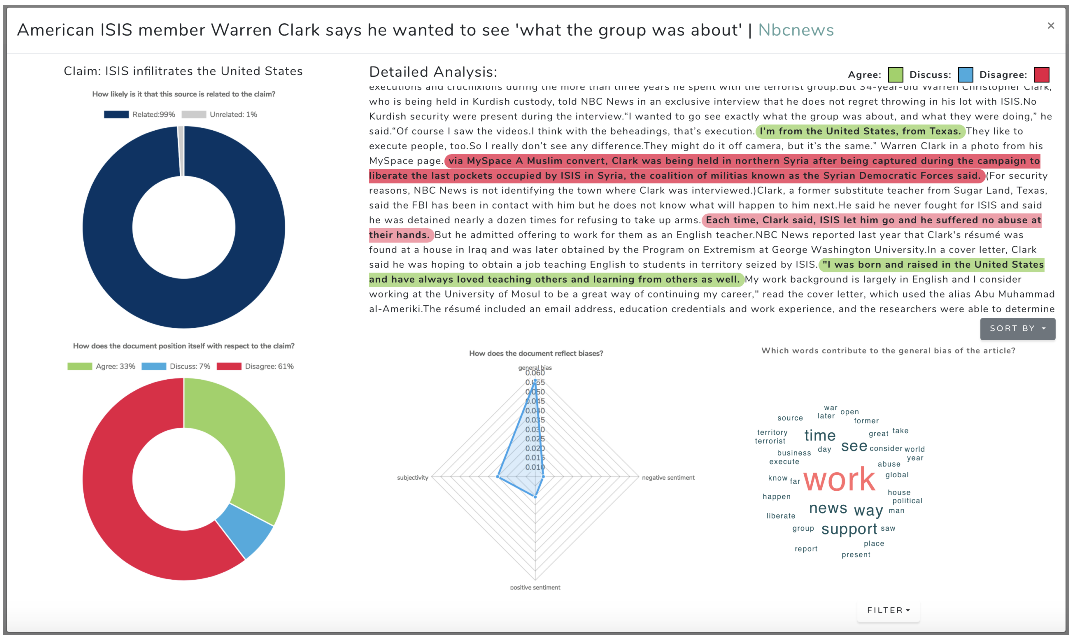

Figure 2: Screenshot of FAKTA for a document retrieved for the claim "ISIS infiltrates the United States."

2.4 Linguistic Analysis

We analyze the language used in documents using the following linguistic markers:
—*Subjectivity lexicon* (Riloff and Wiebe, 2003): which contains weak and strong subjective terms (we only consider the strong subjectivity cues),
—*Sentiment cues* (Liu et al., 2005): which contains *positive* and *negative* sentiment cues, and
—*Wiki-bias lexicon* (Recasens et al., 2013): which involves bias cues and controversial words (e.g., *abortion* and *execute*) extracted from the Neutral Point of View Wikipedia corpus (Recasens et al., 2013).

Finally, we compute a score for the document using these cues according to Equation (2), where for each lexicon type L_i and document D_j, the frequency of the cues for L_i in D_j is normalized by the total number of words in D_j:

$$L_i(D_j) = \frac{\sum\limits_{cue \in L_i} count(cue, D_j)}{\sum\limits_{w_k \in D_j} count(w_k, D_j)} \quad (2)$$

These scores are shown in a radar chart in Figure 2. Furthermore, FAKTA provides the option to see a lexicon-specific word cloud of frequent words in each documents (see Figure 2, the right side of the radar chart which shows the word cloud of Sentiment cues in the document).

2.5 Aggregation

Stance Detection and Linguistic Analysis components are executed in parallel against all documents retrieved by our document retrieval component from each type of sources. All the stance scores are averaged across these documents, and the aggregated scores are shown for each *agree*, *disagree* and *discuss* categories at the top of the ranked list of retrieved documents. Higher agree score indicates the claim is factually true, and higher disagree score indicates the claim is factually false.

3 Evaluation and Results

We use the Fact Extraction and VERification (FEVER) dataset (Thorne et al., 2018) to evaluate our system. In FEVER, each claim is assigned to its relevant Wikipedia documents with agree/disagree stances to the claim, and claims are labeled as *supported* (SUP, i.e. factually true), *refuted* (REF, i.e. factually false), and *not enough information* (NEI, i.e., there is not any relevant document for the claim in Wikipedia). The data includes a total of 145K claims, with around 80K, 30K and 35K SUP, REF and NEI labels respectively.

Document Retrieval: Table 1 shows results for document retrieval. We use various search and ranking algorithms that measure the similarity between each input claim as query and Web documents. Lines 1–11 in the table show the results when we use Lucene to index and search the data corpus with the following retrieval models: BM25 (Robertson et al., 1994) (Line 1), Classic based on the TF.IDF model (Line 2), and Divergence from Independence (DFI) (Kocabaş et al.,

Model		R@1	R@5	R@10	R@20
1.	BM25	28.84	38.66	62.34	70.10
2.	Classic	9.14	23.10	31.65	40.70
3.	DFI	40.93	66.98	74.84	81.22
4.	DFR_{H3}	43.67	71.18	78.32	83.16
5.	DFR_{Z}	43.14	71.17	78.60	83.88
6.	IB_{LL}	41.86	68.02	75.46	81.13
7.	IB_{SPL}	42.27	69.55	77.03	81.99
8.	LMDirichlet	39.00	68.86	77.39	83.04
9.	$\text{LMJelinek}_{0.05}$	37.39	59.75	67.58	74.15
10.	$\text{LMJelinek}_{0.10}$	37.30	59.85	67.58	74.44
11.	$\text{LMJelinek}_{0.20}$	37.01	59.60	67.60	74.62
using Query Generation					
12.	Lucene_{DFR_Z}	40.70	68.48	76.21	81.93
13.	Google API	56.62	71.92	73.86	74.89
using Re-ranking Model					
14.	Lucene_{DFR_Z}	**62.37**	**78.12**	**80.84**	**82.11**
15.	Google API	57.80	72.10	74.15	74.89

Table 1: Results of document retrieval on FEVER.

2014) (Line 3). We also use Divergence from Independence Randomness (DFR) (Amati and Van Rijsbergen, 2002) with different term frequency normalization, such as the normalization provided by Dirichlet prior (DFR_{H_3}) (Line 4) or a Zipfian relation prior (DFR_z) (Line 5). We also consider Information Based (IB) models (Clinchant and Gaussier, 2010) with Log-logistic (IB_{LL}) (Line 6) or Smoothed power-law (IB_{SPL}) (Line 7) distributions. Finally, we consider LMDirichlet (Zhai and Lafferty, 2001) (Line 8), and LMJelinek (Zhai and Lafferty, 2001) with different settings for its hyperparameter (Lines 9–11). According to the resulting performance at different ranks $\{1\text{–}20\}$, we select the ranking algorithm DFR_z (Lucene_{DFR_Z}) as our retrieval model.

In addition, Lines 12–13 show the results when claims are converted to queries as explained in Section 2.1. The results (Lines 5 and 12) show that Lucene performance decreases with query generation. This might be because the resulting queries become more abstract than their corresponding claims which may introduce some noise to the intended meaning of claims. However, Lines 14–15 show that our re-ranking model, explained in Section 2.1, can improve both Lucene and Google results.

FAKTA Full Pipeline: The complete pipeline consists of document retrieval and re-ranking model (Section 2.1), stance detection and rationale extraction[5] (Section 2.3) and aggregation model (Section 2.5). Table 2 shows the results for the full pipeline. Lines 1–3 show the results for all three SUP, REF, and NEI labels (3lbl) and Ran-

Model		Settings	$\text{F}_{1(SUP/REF/NEI)}$	$\text{F}_{1(Macro)}$	Acc.
1.	MLP	3lbl/RS	-	-	40.63
2.	FAKTA	L/3lbl/RS	41.33/23.55/44.79	36.56	38.76
3.	FAKTA	G/3lbl/RS	47.49/43.01/28.17	39.65	41.21
4.	FAKTA	L/2lbl	58.33/57.71/-	58.02	58.03
5.	FAKTA	G/2lbl	58.96/59.74/-	59.35	59.35

Table 2: FAKTA full pipeline Results on FEVER.

domly Sampled (RS) documents from Wikipedia for the NEI label. We label claims as NEI if the most relevant document retrieved has a retrieval score less than a threshold, which was determined by tuning on development data. Line 1 is the multilayer perceptron (MLP) model presented in (Riedel et al., 2017). Lines 2–3 are the results for our system when using Lucene (L) and Google API (G) for document retrieval. The results show that our system achieves the highest performance on both $\text{F}_{1(Macro)}$ and accuracy (Acc) using Google as retrieval engine. We repeat our experiments when considering only SUP and REF labels (2lbl) and the results are significantly higher than the results with 3lbl (Lines 4–5).

4 The System in Action

The current version of FAKTA[6] and its short introduction video[7] and source code[8] are available online. FAKTA consists of three views:

—*The text entry view*: to enter a claim to be checked for factuality.

—*Overall result view*: includes four lists of retrieved documents from four factuality types of sources: Wikipedia, and high-, mixed-, and low-factuality media (Section 2.2). For each list, the final factuality score for the input claim is shown at the top of the page (Section 2.5), and the stance detection score for each document appears beside it.

—*Document result view*: when selecting a retrieved document, FAKTA shows the text of the document and highlights its important sentences according to their stance scores with respect to the claim. The stance detection results for the document are further shown as pie chart at the left side of the view (Section 2.3), and the linguistic analysis is shown at the bottom of the view (Section 2.4).

5 Related Work

Automatic fact checking (Xu et al., 2018) centers on evidence extraction for given claims, re-

[5]We used Intel AI's Distiller (Zmora et al., 2018) to compress the model.

[6]http://fakta.mit.edu

[7]http://fakta.mit.edu/video

[8]https://github.com/moinnadeem/fakta

liability evaluation of media sources (Baly et al., 2018a), stance detection of documents with respect to claims (Mohtarami et al., 2018; Xu et al., 2018; Baly et al., 2018b), and fact checking of claims (Mihaylova et al., 2018). These steps correspond to different Natural Language Processing (NLP) and Information Retrieval (IR) tasks including information extraction and question answering (Shiralkar et al., 2017). Veracity inference has been mostly approached as text classification problem and mainly tackled by developing linguistic, stylistic, and semantic features (Rashkin et al., 2017; Mihaylova et al., 2018; Nakov et al., 2017), as well as using information from *external* sources (Mihaylova et al., 2018; Karadzhov et al., 2017).

These steps are typically handled in isolation. For example, previous works (Wang, 2017; OBrien et al., 2018) proposed algorithms to predict factuality of claims by mainly focusing on only input claims (i.e., step (*iv*) and their metadata information (e.g., the speaker of the claim). In addition, recent works on the Fact Extraction and VERification (FEVER) (Thorne et al., 2018) has focused on a specific domain (e.g., Wikipedia).

To the best of our knowledge, there is currently no end-to-end systems for fact checking which can search through Wikipedia and mainstream media sources across the Web to fact check given claims. To address these gaps, our FAKTA system covers all fact-checking steps and can search across different sources, predict the factuality of claims, and present a set of evidence to explain its prediction.

6 Conclusion

We have presented FAKTA–an online system for automatic end-to-end fact checking of claims. FAKTA can assist individuals and professional fact-checkers to check the factuality of claims by presenting relevant documents and rationales as evidence for its predictions. In future work, we plan to improve FAKTA's underlying components (e.g., stance detection), extend FAKTA to cross-lingual settings, and incorporate temporal information for fact checking.

Acknowledgments

We thank anonymous reviewers for their insightful comments, suggestions, and feedback. This research was supported in part by HBKU Qatar Computing Research Institute (QCRI), DSTA of Singapore, and Intel AI.

References

Gianni Amati and Cornelis Joost Van Rijsbergen. 2002. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Transactions on Information Systems (TOIS)*, 20(4):357–389.

Ramy Baly, Georgi Karadzhov, Dimitar Alexandrov, James Glass, and Preslav Nakov. 2018a. Predicting factuality of reporting and bias of news media sources. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3528–3539. Association for Computational Linguistics.

Ramy Baly, Mitra Mohtarami, James Glass, Lluís Màrquez, Alessandro Moschitti, and Preslav Nakov. 2018b. Integrating stance detection and fact checking in a unified corpus. In *Proceedings of the 16th Annualw Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL-HLT '18, New Orleans, LA, USA.

Stéphane Clinchant and Eric Gaussier. 2010. Information-based models for ad hoc ir. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'10, pages 234–241, New York, NY, USA. ACM.

Georgi Karadzhov, Preslav Nakov, Lluís Màrquez, Alberto Barrón-Cedeño, and Ivan Koychev. 2017. Fully automated fact checking using external sources. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 344–353. INCOMA Ltd.

İlker Kocabaş, Bekir Taner Dinçer, and Bahar Karaoğlan. 2014. A nonparametric term weighting method for information retrieval based on measuring the divergence from independence. *Inf. Retr.*, 17(2):153–176.

Nayeon Lee, Chien-Sheng Wu, and Pascale Fung. 2018. Improving large-scale fact-checking using decomposable attention models and lexical tagging. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1133–1138. Association for Computational Linguistics.

Bing Liu, Minqing Hu, and Junsheng Cheng. 2005. Opinion observer: Analyzing and comparing opinions on the web. In *Proceedings of the 14th International Conference on World Wide Web*, pages 342–351, Chiba, Japan.

Todor Mihaylov, Georgi Georgiev, and Preslav Nakov. 2015. Finding opinion manipulation trolls in news community forums. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 310–314. Association for Computational Linguistics.

Todor Mihaylov and Preslav Nakov. 2016. Hunting for troll comments in news community forums. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 399–405, Berlin, Germany.

Tsvetomila Mihaylova, Preslav Nakov, Lluis Marquez, Alberto Barron-Cedeno, Mitra Mohtarami, Georgi Karadzhov, and James Glass. 2018. Fact checking in community forums. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 5309–5316, New Orleans, LA, USA.

Mitra Mohtarami, Ramy Baly, James Glass, Preslav Nakov, Lluís Màrquez, and Alessandro Moschitti. 2018. Automatic stance detection using end-to-end memory networks. In *Proceedings of the 16th Annualw Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL-HLT '18, New Orleans, LA, USA.

Preslav Nakov, Tsvetomila Mihaylova, Lluıs Marquez, Yashkumar Shiroya, and Ivan Koychev. 2017. Do not trust the trolls: Predicting credibility in community question answering forums. In *Proceedings of the International Conference Recent Advances in Natural Language Processing (RANLP)*, pages 551–560.

An Nguyen, Aditya Kharosekar, Matthew Lease, and Byron Wallace. 2018. An interpretable joint graphical model for fact-checking from crowds. In *AAAI Conference on Artificial Intelligence*.

Nicole OBrien, Sophia Latessa, Georgios Evangelopoulos, and Xavier Boix. 2018. The language of fake news: Opening the black-box of deep learning based detectors. In *Proceedings of the Thirty-second Annual Conference on Neural Information Processing Systems (NeurIPS)–AI for Social Good*.

Kashyap Popat, Subhabrata Mukherjee, Jannik Strötgen, and Gerhard Weikum. 2017. Where the truth lies: Explaining the credibility of emerging claims on the web and social media. In *Proceedings of the 26th International Conference on World Wide Web Companion*, WWW '17 Companion, pages 1003–1012, Republic and Canton of Geneva, Switzerland.

Martin Potthast, Matthias Hagen, Tim Gollub, Martin Tippmann, Johannes Kiesel, Paolo Rosso, Efstathios Stamatatos, and Benno Stein. 2013. Overview of the 5th international competition on plagiarism detection. In *Working Notes for CLEF 2013 Conference , Valencia, Spain, September 23-26, 2013*.

Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. 2017. Truth of varying shades: Analyzing language in fake news and political fact-checking. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2921–2927.

Marta Recasens, Cristian Danescu-Niculescu-Mizil, and Dan Jurafsky. 2013. Linguistic models for analyzing and detecting biased language. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 1650–1659, Sofia, Bulgaria.

Benjamin Riedel, Isabelle Augenstein, Georgios P. Spithourakis, and Sebastian Riedel. 2017. A simple but tough-to-beat baseline for the fake news challenge stance detection task. *CoRR*, abs/1707.03264.

Ellen Riloff and Janyce Wiebe. 2003. Learning extraction patterns for subjective expressions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 105–112, Sapporo, Japan.

Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1994. Okapi at trec-3. In *TREC*, pages 109–126. National Institute of Standards and Technology (NIST).

Prashant Shiralkar, Alessandro Flammini, Filippo Menczer, and Giovanni Luca Ciampaglia. 2017. Finding streams in knowledge graphs to support fact checking. *arXiv preprint arXiv:1708.07239*.

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 (HLT-NAACL)*, pages 809–819.

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

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. Association for Computational Linguistics.

Brian Xu, Mitra Mohtarami, and James Glass. 2018. Adversarial doman adaptation for stance detection. In *Proceedings of the Thirty-second Annual Conference on Neural Information Processing Systems (NeurIPS)–Continual Learning*.

Chengxiang Zhai and John Lafferty. 2001. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'01, pages 334–342, New York, NY, USA. ACM.

Neta Zmora, Guy Jacob, and Gal Novik. 2018. Neural network distiller. Available at https://doi.org/10.5281/zenodo.1297430.