That is a Known Lie: Detecting Previously Fact-Checked Claims

Shaden Shaar[1], **Nikolay Babulkov**[2], **Giovanni Da San Martino**[1], **Preslav Nakov**[1]
[1]Qatar Computing Research Institute, HBKU, Doha, Qatar
[2]Sofia University, Sofia, Bulgaria
{sshar, gmartino, pnakov}@hbku.edu.qa
nbabulkov@gmail.com

Abstract

The recent proliferation of "fake news" has triggered a number of responses, most notably the emergence of several manual fact-checking initiatives. As a result and over time, a large number of fact-checked claims have been accumulated, which increases the likelihood that a new claim in social media or a new statement by a politician might have already been fact-checked by some trusted fact-checking organization, as viral claims often come back after a while in social media, and politicians like to repeat their favorite statements, true or false, over and over again. As manual fact-checking is very time-consuming (and fully automatic fact-checking has credibility issues), it is important to try to save this effort and to avoid wasting time on claims that have already been fact-checked. Interestingly, despite the importance of the task, it has been largely ignored by the research community so far. Here, we aim to bridge this gap. In particular, we formulate the task and we discuss how it relates to, but also differs from, previous work. We further create a specialized dataset, which we release to the research community. Finally, we present learning-to-rank experiments that demonstrate sizable improvements over state-of-the-art retrieval and textual similarity approaches.

1 Introduction

The year 2016 was marked by massive disinformation campaigns related to Brexit and the US Presidential Elections. While false statements are not a new phenomenon, e.g., yellow press and tabloids have been around for decades, this time things were notably different in terms of scale and effectiveness thanks to social media platforms, which provided both a medium to reach millions of users and an easy way to micro-target specific narrow groups of voters based on precise geographical, demographic, psychological, and/or political profiling.

Governments, international organizations, tech companies, media, journalists, and regular users launched a number of initiatives to limit the impact of the newly emerging large-scale weaponization of *disinformation*[1] online. Notably, this included manual fact-checking initiatives, which aimed at debunking various false claims, with the hope to limit its impact, but also to educate the public that not all claims online are true.

Over time, the number of such initiatives grew substantially, e.g., at the time of writing, the Duke Reporters' Lab lists 237 active fact-checking organizations plus another 92 inactive.[2] While some organizations debunked just a couple of hundred claims, others such as Politifact,[3] FactCheck.org,[4] Snopes,[5] and Full Fact[6] have fact-checked thousands or even tens of thousands of claims.

The value of these collections of resources has been recognized in the research community, and they have been used to train systems to perform automatic fact-checking (Popat et al., 2017; Wang, 2017; Zlatkova et al., 2019) or to detect check-worthy claims in political debates (Hassan et al., 2015; Gencheva et al., 2017; Patwari et al., 2017; Vasileva et al., 2019). There have also been datasets that combine claims from multiple fact-checking organizations (Augenstein et al., 2019), again with the aim of performing automatic fact-checking.

[1] In the public discourse, the problem is generally known as "*fake news*", a term that was declared Word of the Year 2017 by Collins dictionary. Despite its popularity, it remains a confusing term, with no generally agreed upon definition. It is also misleading as it puts emphasis on (a) the claim being false, while generally ignoring (b) its intention to do harm. In contrast, the term *disinformation* covers both aspects (a) and (b), and it is generally preferred at the EU level.

[2] http://reporterslab.org/fact-checking/

[3] http://www.politifact.com/

[4] http://www.factcheck.org/

[5] http://www.snopes.com/

[6] http://fullfact.org/

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 3607–3618
July 5 - 10, 2020. ©2020 Association for Computational Linguistics

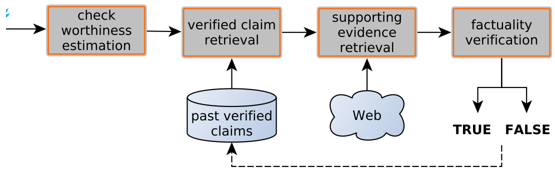

Figure 1: A general information verification pipeline.

It has been argued that checking against a database of previously fact-checked claims should be an integral step of an end-to-end automated fact-checking pipeline (Hassan et al., 2017). This is illustrated in Figure 1, which shows the general steps of such a pipeline (Elsayed et al., 2019): (*i*) assess the check-worthiness of the claim (which could come from social media, from a political debate, etc.), (*ii*) check whether a similar claim has been previously fact-checked (the task we focus on here), (*iii*) retrieve evidence (from the Web, from social media, from Wikipedia, from a knowledge base, etc.), and (*iv*) assess the factuality of the claim.

From a fact-checkers' point of view, the abundance of previously fact-checked claims increases the likelihood that the next claim that needs to be checked would have been fact-checked already by some trusted organization. Indeed, viral claims often come back after a while in social media, and politicians are known to repeat the same claims over and over again.[7] Thus, before spending hours fact-checking a claim manually, it is worth first making sure that nobody has done it already.

On another point, manual fact-checking often comes too late. A study has shown that "fake news" spreads six times faster than real news (Vosoughi et al., 2018). Another study has indicated that over 50% of the spread of some viral claims happens within the first ten minutes of their posting on social media (Zaman et al., 2014). At the same time, detecting that a new viral claim has already been fact-checked can be done automatically and very quickly, thus allowing for a timely action that can limit the spread and the potential malicious impact.

From a journalistic perspective, the ability to check quickly whether a claim has been previously fact-checked could be revolutionizing as it would allow putting politicians on the spot in real time, e.g., during a live interview. In such a scenario, automatic fact-checking would be of limited utility as, given the current state of technology, it does not offer enough credibility in the eyes of a journalist.

Interestingly, despite the importance of the task of detecting whether a claim has been fact-checked in the past, it has been largely ignored by the research community. Here, we aim to bridge this gap. Our contributions can be summarized as follows:

- We formulate the task and we discuss how it relates to, but differs from, previous work.
- We create a specialized dataset, which we release to the research community.[8] Unlike previous work in fact-checking, which used normalized claims from fact-checking datasets, we work with naturally occurring claims, e.g., in debates or in social media.
- We propose a learning-to-rank model that achieves sizable improvements over state-of-the-art retrieval and textual similarity models.

The remainder of this paper is organized as follows: Section 2 discusses related work, Section 3 introduces the task, Section 4 presents the dataset, Section 5 discusses the evaluation measures, Section 6 presents the models we experiment with, Section 7 described our experiments, and Section 8 concludes and discusses future work.

2 Related Work

To the best of our knowledge, the task of detecting whether a claim has been previously fact-checked was not addressed before. Hassan et al. (2017) mentioned it as an integral step of their end-to-end automated fact-checking pipeline, but there was very little detail provided about this component and it was not evaluated.

In an industrial setting, Google has developed *Fact Check Explorer*,[9] which is an exploration tool that allows users to search a number of fact-checking websites (those that use ClaimReview from `schema.org`[10]) for the mentions of a topic, a person, etc. However, the tool cannot handle a complex claim, as it runs Google search, which is not optimized for semantic matching of long claims. While this might change in the future, as there have been reports that Google has started using BERT in its search, at the time of writing, the tool could not handle a long claim as an input.

[7] President Trump has repeated one claim over 80 times: http://tinyurl.com/yblcb5q5.

[8] Data and code are available at the following URL: https://github.com/sshaar/That-is-a-Known-Lie

[9] http://toolbox.google.com/factcheck/explorer

[10] http://schema.org/ClaimReview

A very similar work is the *ClaimsKG* dataset and system (Tchechmedjiev et al., 2019), which includes 28K claims from multiple sources, organized into a knowledge graph (KG). The system can perform data exploration, e.g., it can find all claims that contain a certain named entity or keyphrase. In contrast, we are interested in detecting whether a claim was previously fact-checked.

Other work has focused on creating datasets of textual fact-checked claims, without building KGs. Some of the larger ones include the *Liar, Liar* dataset of 12.8K claims from PolitiFact (Wang, 2017), and the *MultiFC* dataset of 38K claims from 26 fact-checking organizations (Augenstein et al., 2019), the 10K claims *Truth of Various Shades* (Rashkin et al., 2017) dataset, among several other datasets, which were used for automatic fact-checking of individual claims, not for checking whether an input claim was fact-checked previously. Note that while the above work used manually normalized claims as input, we work with naturally occurring claims as they were made in political debates and speeches or in social media.

There has also been a lot of research on automatic fact-checking of claims and rumors, going in several different directions. One research direction focuses on the social aspects of the claim and how users in social media react to it (Canini et al., 2011; Castillo et al., 2011; Ma et al., 2016; Gorrell et al., 2019; Ma et al., 2019). Another direction mines the Web for information that proves or disproves the claim (Mukherjee and Weikum, 2015; Karadzhov et al., 2017; Popat et al., 2017; Baly et al., 2018b; Mihaylova et al., 2018; Nadeem et al., 2019). In either case, it is important to model the reliability of the source as well as the stance of the claim with respect to other claims; in fact, it has been proposed that a claim can be fact-checked based on its source alone (Baly et al., 2018a) or based on its stance alone (Dungs et al., 2018). A third direction performs fact-checking against Wikipedia (Thorne et al., 2018; Nie et al., 2019), or against a general collection of documents (Miranda et al., 2019). A fourth direction uses a knowledge base or a knowledge graph (Ciampaglia et al., 2015; Shiadralkar et al., 2017; Gad-Elrab et al., 2019a,b; Huynh and Papotti, 2019). Yet another direction performs fact-checking based on tables (Chen et al., 2019). There is also recent work on using language models as knowledge bases (Petroni et al., 2019). Ours is yet another research direction.

While our main contribution here is the new task and the new dataset, we should also mentioned some work on retrieving documents. In our experiments, we perform retrieval using BM25 (Robertson and Zaragoza, 2009) and re-ranking using BERT-based similarity, which is a common strategy in recent state-of-the-art retrieval models (Akkalyoncu Yilmaz et al., 2019a; Nogueira and Cho, 2019; Akkalyoncu Yilmaz et al., 2019b).

Our approach is most similar to that of (Akkalyoncu Yilmaz et al., 2019a), but we differ, as we perform matching, both with BM25 and with BERT, against the normalized claim, against the title, and against the full text of the articles in the fact-checking dataset; we also use both scores and reciprocal ranks when combining different scores and rankings. Moreover, we use sentence-BERT instead of BERT. Previous work has argued that BERT by itself does not yield good sentence representation. Thus, approaches such as sentence-BERT (Reimers and Gurevych, 2019) have been proposed, which are specifically trained to produce good sentence-level representations. This is achieved using Siamese BERT networks that are fine-tuned on NLI and STS-B data. Indeed, in our experiments, we found sentence-BERT to perform much better than BERT. The Universal Sentence Encoder (Cer et al., 2018) is another alternative, but sentence-BERT worked better in our experiments.

Finally, our task is related to semantic relatedness tasks, e.g., from the GLUE benchmark (Wang et al., 2018), such as natural language inference, or NLI task (Williams et al., 2018), recognizing textual entailment, or RTE (Bentivogli et al., 2009), paraphrase detection (Dolan and Brockett, 2005), and semantic textual similarity, or STS-B (Cer et al., 2017). However, it also differs from them, as we will see in the following section.

3 Task Definition

We define the task as follows: *Given a check-worthy input claim and a set of verified claims, rank those verified claims, so that the claims that can help verify the input claim, or a sub-claim in it, are ranked above any claim that is not helpful to verify the input claim.*

Table 1 shows some examples of *input–verified* claim pairs, where the input claims are sentences from the 2016 US Presidential debates, and the verified claims are the corresponding fact-checked counter-parts in PolitiFact.

No.	Input claim	Manually annotated claim in PolitiFact
1	*Richard Nixon released tax returns when he was under audit.*	Richard Nixon released tax returns when he was under audit.
2	*Hillary wants to give amnesty.*	Says Hillary Clinton "wants to have open borders."
3	*People with tremendous medical difficulty and medical problems are pouring in, and in many, in many cases, it's contagious.*	Says "2,267 caravan invaders have tuberculosis, HIV, chickenpox and other health issues"
4	*He actually advocated for the actions we took in Libya and urged that Gadhafi be taken out, after actually doing some business with him one time.*	Says Donald Trump is "on record extensively supporting the intervention in Libya."
5	*He actually advocated for the actions we took in Libya and urged that Gadhafi be taken out, after actually doing some business with him one time.*	When Moammar Gadhafi was set to visit the United Nations, and no one would let him stay in New York, Trump allowed Gadhafi to set up an elaborate tent at his Westchester County (New York) estate.

Table 1: **PolitiFact:** *Input–verified* claim pairs. The input claims are sentences from the 2016 US Presidential debates, and the verified claims are their corresponding fact-checked counter-parts in PolitiFact.

We can see on line 1 of Table 1 a trivial case, where the verified claim is identical to the input claim; however, such cases are not very frequent, as the experiments with the BM25 baseline in Section 7 below will show.

Lines 2 and 3 show harder cases, where the input claim and its manually annotated counter-part are quite different in their lexical choice, and yet the latter can serve to verify the former.

Lines 4 and 5, show a complex input claim, which contains two sub-claims, each of which is verified by two corresponding claims in PolitiFact.

From the above examples, it is clear that ours is not a paraphrasing task, as illustrated by examples 2–5. It is also not a natural language inference (NLI) or a recognizing textual entailment (RTE) task, as a claim can have sub-claims, which complicates entailment reasoning (as illustrated by examples 4–5). Finally, the task goes beyond simple textual similarity, and thus it is not just an instance of semantic textual similarity (STS-B).

Note that we do not try to define formally what makes a verified claim a good match for an input claim. Instead, we trust the manual annotations for this by fact-checking experts, which they perform when they comment on the claims made in political debates and speeches. In many cases, the fact-checkers have explicitly indicated which previously fact-checked claim corresponds to a given original claim in a debate/speech. A similar approach was adopted for a related task, e.g., it was used to obtain annotated training and testing data for the Check-Worthiness task of the CLEF CheckThat! Lab (Atanasova et al., 2018, 2019; Barrón-Cedeño et al., 2020).

4 Datasets

We created two datasets by collecting, for each of them, a set of *verified claims* and matching *input–verified* claims pairs (below, we will also refer to these pairs as *Input-VerClaim* pairs): the first dataset, PolitiFact, is about political debates and speeches and it is described in Section 4.1; the second dataset, Snopes, includes tweets, and it is described in Section 4.2.

4.1 PolitiFact Dataset

PolitiFact is a fact-checking website that focuses on claims made by politicians, elected officials, and influential people in general. PolitiFact fact-checks claims by assigning a truth value to them and publishing an article that gives background information and explains the assigned label. This is similar to how other fact-checking websites operate.

We retrieved a total of 16,636 verified claims from PolitiFact, populating for each of them the following fields:

- *VerClaim*: the text of the claim, which is a normalized version of the original claim, as the human fact-checkers typically reformulate it, e.g., to make it clearer, context-independent, and self-contained;
- *TruthValue*: the label assigned to the claim;[11]
- *Title*: the title of the article on PolitiFact that discusses the claim;
- *Body*: the body of the article.

[11] We do not use the claim veracity labels in our experiments, but we collect them for possible future use.

No.	Tweet	Manually annotated claim in Snopes
1	*Welp... its official... Kim Kardashian finally decided to divorce Kanye West https://t.co/C2p25mxWJO — Ashlee Marie Preston (@AshleeMPreston) October 12, 2018*	Kanye West and Kim Kardashian announced that they were divorcing in October 2018.
2	*Kim Kardashian and Kanye West are splitting up https://t.co/epwKG7aSBg pic.twitter.com/u7qqojWVlR — ELLE Magazine (US) (@ELLEmagazine) October 18, 2018*	Kanye West and Kim Kardashian announced that they were divorcing in October 2018.
3	*Everyone should be able to access high-quality, affordable, gender-affirming health care. But the Trump administration is trying to roll back important protections for trans Americans. Help fight back by leaving a comment for HHS in protest: https://t.co/pKDcOqbsc7 — Elizabeth Warren (@ewarren) August 13, 2019*	U.S. Sen. Elizabeth Warren said or argued to the effect that taxpayers must fund sex reassignment surgery.

Table 2: **Snopes:** *Input–VerClaim* claim pairs. The input claims are tweets and the verified claims are their corresponding fact-checked counter-parts in Snopes.

Often, after a major political event, such as a political speech or a debate, PolitiFact publishes reports[12] that discuss the factuality of some of the claims made during that event. Importantly for us, in these reports, some of the claims are linked to previously verified claims in PolitiFact. Such pairs of an original claim and a previously verified claim form our *Claim–VerClaim* pairs.

We collected such overview reports for 78 public events in the period 2012–2019, from which we collected a total of 768 *Input–VerClaim* pairs. Given an *Input* claim, we refer to the corresponding *verified* claim in the pair as its *matching VerClaim claim*. In general, there is a 1:1 correspondence, but in some cases an *Input* claim is mapped to multiple *VerClaim* claims in the database, and in other cases, multiple *Input* claims are matched to the same *VerClaim* claim.

Thus, the task in Section 3 reads as follows when instantiated to the PolitiFact dataset: given an *Input* claim, rank all 16,636 *VerClaim* claims, so that its *matching VerClaim* claims are ranked at the top.

4.2 Snopes Dataset

Snopes is a website specialized in fact-checking myths, rumors, and urban legends. We used information from it to create a second dataset, this time focusing on tweets. We started with a typical article about a claim, and we looked inside the article for links to tweets that are possibly making that claim. Note that some tweets mentioned in the article are not making the corresponding verified claim, and some are not making any claims; we manually checked and filtered out such tweets.

We collected 1,000 suitable tweets as *Input* claims, and we paired them with the corresponding claim that the page is about as the *VerClaim* claim. We further extracted from the article its *Title*, and the *TruthValue* of the Input claim (a rating of the claims assigned from Snopes[13]).

Examples of *input–VerClaim* pairs are shown in Table 2. Comparing them to the ones from Table 1, we can observe that the Snopes tweets are generally more self-contained and context-independent.

Finally, we created a set of *VerClaim* claims to match against using the Snopes claims in the *ClaimsKG* dataset (Tchechmedjiev et al., 2019). Ultimately, our Snopes dataset consists of 1,000 *input–VerClaim* pairs and 10,396 verified claims.

Statistics about the datasets are shown in Table 3; the datasets are available online.[8]

4.3 Analysis

In section 3, we discussed that matching some of the input claims with the corresponding verified claims can be a non-trivial task, and we gave examples of easy and hard cases. To capture this distinction, we classify Input–VerClaim pairs into two types. *Type-1* pairs are such for which the Input claim can be matched to the VerClaim using simple approximate string matching techniques., e.g., as in line 1 of Table 1 and lines 1-2 of Table 2. Conversely, *Type-2* pairs are such for which the Input claim cannot be easily mapped to the VerClaim, e.g., as in lines 2-5 of Table 1 and line 3 of Table 2. We manually annotated a sample of 100 pairs from PolitiFact input–VerClaimpairs and we found 48% of them to be of *Type-2*.

[12] Note that these reports discuss multiple claims, unlike the typical PolitiFact article about a specific claim.

[13] `http://www.snopes.com/fact-check-ratings/`

	PolitiFact	**Snopes**
Input–VerClaim pairs	768	1,000
– training	614	800
– testing	154	200
Total # of verified claims	16,636	10,396

Table 3: **Statistics about the datasets:** shown are the number of *Input–VerClaim* pairs and the total number of *VerClaim* claims to match an *Input* claim against. Note that each *VerClaim* comes with an associated fact-checking analysis document in PolitiFact/Snopes.

	PolitiFact		**Snopes**	
Threshold	**#**	**%**	**#**	**%**
0.75	55	8%	11	1%
0.50	128	17%	75	8%
0.25	201	27%	504	50%
0.00	768	100%	1,000	100%

Table 4: **Analysis of the task complexity:** number of *Input–VerClaim* pairs in PolitiFact and Snopes with TF.IDF-weighted cosine similarity above a threshold.

We further analyzed the complexity of matching an *Input* claim to the *VerClaim* from the same *Input–VerClaim* pair using word-level TF.IDF-weighted cosine similarity. Table 4 shows the number of pairs for which this similarity is above a threshold. We can see that, for PolitiFact, only 27% of the pairs have a similarity score that is above 0.25, while for Snopes, this percentage is at 50%, which suggests Snopes should be easier than PolitiFact.

5 Evaluation Measures

We treat the task as a ranking problem. Thus, we use ranking evaluation measures, namely mean reciprocal rank (MRR), Mean Average Precision (MAP), and MAP truncated to rank k (MAP@k). We also report HasPositive@k, i.e., whether there is a true positive among the top-k results.

Measures such as MAP@k and HasPositive@k for $k \in \{1, 3, 5\}$ would be relevant in a scenario, where a journalist needs to verify claims in real time, in which case the system would return a short list of 3-5 claims that the journalist can quickly skim and make sure they are indeed a true match.

We further report MAP@k and HasPositive@k for $k \in \{10, 20\}$ as well as MAP (untruncated), which would be more suitable in a non-real-time scenario, where recall would be more important.

6 Models

Here, we describe the models we experiment with.

6.1 BM25

A simple baseline is to use BM25 (Robertson and Zaragoza, 2009), which is classical approach in information retrieval. BM25 assigns a score to each query-document pair based on exact matching between the words in the query and the words in a target document, and it uses this score for ranking. We experiment with BM25 using the input claim as a query against different representations of the verified claims:

- **IR (Title):** the article titles;
- **IR (VerClaim):** the verified claims;
- **IR (Body):** the article bodies;
- Combinations of the above.

6.2 BERT-based Models

The BM25 algorithm focuses on exact matches, but as lines 2–5 in Table 1 and line 3 in Table 2 show, the input claim can use quite different words. Thus, we further try semantic matching using BERT.

Initially, we tried to fine-tune BERT (Devlin et al., 2019), but this did not work well, probably because we did not have enough data to perform the fine-tuning. Thus, eventually we opted to use BERT (and variations thereof) as a sentence encoder, and to perform max-pooling on the penultimate layer to obtain a representation for an input piece of text. Then, we calculate the cosine similarity between the representation of the input claim and of the verified claims in the dataset, and we use this similarity for ranking.

- **BERT:base,uncased**: the base, uncased model of BERT;
- **RoBERTa:base**: the base, cased model of RoBERTa (Liu et al., 2019);
- **sentence-BERT:base**: BERT, specifically trained to produce good sentence representations (Reimers and Gurevych, 2019); this is unlike BERT and RoBERTa, for which we found the cosine similarity between totally unrelated claims often to be quite high;
- **sentence-BERT:large**: the large version of sentence-BERT.

Experiment	**MRR**	**MAP@k**						**HasPositives@k**					
		1	3	5	10	20	all	1	3	5	10	20	50
IR (BM25; full database)													
IR:Title	.288	.216	.259	.261	.268	.272	.276	.220	.330	.346	.401	.464	.535
IR:VerClaim	.435	.366	.404	.413	.415	.419	.422	.378	.472	.511	.527	.574	.629
IR:Body	**.565**	**.484**	**.538**	**.546**	**.552**	**.556**	**.560**	**.488**	**.614**	**.653**	**.700**	**.740**	**.811**
IR:Title+VerClaim+Body	.526	.425	.504	.507	.513	.516	.519	.433	**.614**	.630	.661	.717	.772
Semantic Matching - BERT & co. (matching against VerClaim only; full db)													
BERT:base,uncased	.268	.204	.242	.251	.259	.260	.264	.204	.299	.338	.393	.409	.496
RoBERTa:base	.209	.173	.194	.198	.203	.205	.207	.173	.220	.236	.283	.315	.346
sentence-BERT:base	.377	.311	.352	.354	.361	.366	.370	.315	**.417**	**.425**	**.480**	**.551**	**.614**
sentence-BERT:large	**.395**	**.354**	**.367**	**.372**	**.381**	**.382**	**.386**	**.362**	.393	.417	**.480**	.496	.582
BERT on Full Articles (sent.BERT:large matching against VerClaim + Title + top-n article body sent.; full db)													
sentence-BERT: n = 3	.515	**.441**	.478	.493	.498	.501	.505	**.457**	.528	**.598**	**.638**	**.693**	**.756**
sentence-BERT: n = 4	**.517**	**.441**	**.487**	**.497**	**.500**	**.505**	**.508**	**.457**	**.551**	**.598**	.622	.685	**.756**
sentence-BERT: n = 5	.515	.433	.484	.491	.498	.502	.505	.449	**.551**	.583	**.638**	.685	.748
sentence-BERT: n = 6	.509	.429	.480	.485	.491	.497	.500	.441	.543	.567	.614	**.693**	.740
Reranking (IR & sent.BERT:large matching against VerClaim + Title + top-4 article body sent.; top-N from IR)													
Rerank-IR-top-10	.586	.528	.572	.578	.583	—	—	.512	.638	.693	.701	—	—
Rerank-IR-top-20	.586	.521	.572	.577	.580	.583	—	.512	.646	.685	.709	.740	—
Rerank-IR-top-50	.600	.519	.568	.584	.590	.590	.594	.520	.638	**.717**	**.772**	.780	.811
Rerank-IR-top-100	**.608**	**.531**	**.580**	**.588**	**.597**	**.599**	**.602**	**.535**	**.654**	.685	.740	**.787**	**.819**
Rerank-IR-top-200	.605	.529	.575	.585	.594	.598	.599	**.535**	.646	.685	.756	.803	.811

Table 5: **PolitiFact:** evaluation results on the test set.

- **BERT on full articles:** We further extend the above models to match against the body of the document, borrowing and further developing an idea from (Yang et al., 2019). We use sentence-BERT to encode each sentence in the *Body*, and then we compute the cosine similarity between the input claim and each of those sentences. Next, we collect scores for each claim-document pair, as opposed to having only a single score representing the similarity between the input and a verified claim. These scores include the cosine similarity for (*i*) claim vs. *VerClaim*, (*ii*) claim vs. *Title*, and (*iii*) top-n scores of the claim vs. *Body* sentences. Finally, we train a binary classifier that takes all these scores and predicts whether the claim-document pair is a good match.

6.3 Reranking

Since BM25 and BERT capture different types of information, they can be combined to create a set of features based on the rankings returned by BM25 and the similarity scores computed on the embedding of the claim pairs. Following (Nogueira et al., 2019), we use a reranking algorithm, namely rankSVM with an RBF kernel, which learns to rank using a pairwise loss.

7 Experiments

Below we describe our experiments on the PolitiFact and the Snopes datasets. We start with IR-based models, followed by BERT-based semantic similarity on claims and articles, and finally we experiment with pairwise learning-to-rank models.

7.1 Politifact Experiments

For the PolitFact dataset, we perform experiments with all models from Section 6, and we report the results in Table 5.

7.1.1 Experiment 1: BM25-based Baselines

We ran experiments matching the *Input* against *Title*, *VerClaim*, *Body* and *Title+VerClaim+Body*. We can see in Table 5 that using the *Title* yields the lowest results by a large margin. This is because the *Title* is only a summary, while *VerClaim* and *Body* contain more details and context. We can further see that the best representation, on all measures, is to use the *Body*, which performs better than using *VerClaim* by 0.12-0.14 in terms of MAP@k and MAP, and by 0.09 on MRR. This is probably because the article body is longer, which increases the probability of having more words matching the input claim. Finally, matching against all three targets is slightly worse than using *Body* only.

Experiment	**MRR**	**MAP@k**						**HasPositives@k**					
		1	3	5	10	20	all	1	3	5	10	20	50
IR (BM25; full database)													
IR:Title	.619	.538	.573	.583	.587	.590	.592	.538	.673	.724	.759	.804	.844
IR:VerClaim	.655	**.555**	.589	.598	.600	.602	.605	**.558**	.729	.769	.784	.809	.864
IR:VerClaim+Title	**.664**	**.555**	**.592**	**.600**	**.605**	**.608**	**.609**	**.558**	**.739**	**.774**	**.814**	**.849**	**.879**
Semantic Matching - BERT & co. (matching against VerClaim only; full database)													
sent.BERT-base:Title	.474	.397	.417	.425	.430	.434	.437	.397	.528	.563	.598	.658	.729
sent.BERT-base:VerClaim	.515	.402	.489	.504	.510	.512	.515	.402	.593	**.653**	**.698**	**.739**	.784
sent.BERT-large:VerClaim	**.543**	**.457**	**.518**	**.527**	**.533**	**.535**	**.538**	**.457**	**.603**	.648	.693	.724	**.794**
Reranking (IR & sentence-BERT:large matching against VerClaim + Title; top-N from IR)													
Rerank-IR-top-10	.764	.687	.762	.764	.764	—	—	.673	.859	.869	.869	—	—
Rerank-IR-top-20	.781	.686	.773	.780	.781	.781	—	.678	.869	**.905**	**.920**	.920	—
Rerank-IR-top-50	**.788**	**.691**	**.780**	**.782**	**.784**	**.784**	**.787**	**.693**	**.874**	.894	.915	.925	.930
Rerank-IR-top-100	.775	.669	.758	.760	.760	.760	.774	.673	.859	.889	.910	.925	.930
Rerank-IR-top-200	.778	.672	.762	.764	.764	.764	.777	.678	.864	.884	.910	**.930**	**.950**

Table 6: **Snopes:** evaluation results on the test set.

7.1.2 Experiment 2: Semantic Matching

Next, we experimented with cosine similarity between the *Input* claim and *VerClaim*, as the BM25 experiments above have shown that using *VerClaim* is better than using *Title*.

We can see in Table 5 that BERT:uncased is better than RoBERTa (which is case sensitive) on all measures, which suggests that casing might not matter. We further see that the best semantic model is sentence-BERT: both the base and the large variants of sentence-BERT beat BERT and RoBERTa by at least 13% absolute across all measures (and in some cases, by a much larger margin).

7.1.3 Experiment 3: BERT on Full Articles

Next, we performed full article experiments, where we used the large model of sentence-BERT, as it outperformed the rest of the BERT models shown in Table 5. We extracted similarity scores for each claim-document pair using sentence-BERT:large. We then trained a simple neural network (20-relu-10-relu) for classification. We trained the model for 15 epochs with a batch size of 2,048 using the Adam optimizer with a learning rate of 1e-3. We further used class weighting because the data was severely imbalanced: there were 614 positive exampled out of 10M claim-document pairs, as we paired each of the 614 input claims with each of the 16,636 verified claims in the database.

We ran the experiment for various numbers of top-n cosine scores obtained from the *Body*, as we wanted to investigate the relationship between the model performance and the information it uses.

In the *BERT on Full Articles* section in Table 5, we can see that using the scores for the top-4 best-matching sentences from the article body, together with scores for *VerClaim* and for the article title, yielded the best performance. Moreover, the results got closer to those for BM25, even though overall they still lag a bit behind.

7.1.4 Experiment 4: Reranking

Finally, we trained a pairwise RankSVM model to re-rank the top-N results retrieved using *IR:Body*. For each claim-document pair in the top-N list, we collected the scores for *IR:Title*, *IR:VerClaim*, *IR:Body*, as well as from sentence-BERT:large for $n = 4$ with their corresponding reciprocal ranks for the rankings they induce. As described in Section 6.3, using both methods yields better predictions as this combines exact matching and semantic similarities.

We can see in Table 5 that the re-ranker yielded consistent and sizable improvement over the models from the previous experiments, by 0.04-0.05 points absolute across the different measures, which is remarkable as it is well-known from the literature that BM25 is a very strong baseline for IR tasks. This is because our reranker is able to use both exact and semantic matching to target the different kinds of pairs that are found in the dataset. We also notice that the performance of the re-ranker improves as we increase the length of the list that is being re-ranked until a length of 100, and it starts degrading after that.

7.2 Experiments on Snopes

On the Snopes dataset, we performed experiments analogous to those for the PolitiFact dataset, but with some differences, the most important being that this time we did not perform matching against the article body as the tweets that serve as input claims in our Snopes dataset were extracted from the article body. Note that this was not an issue for the PolitiFact dataset, as the input claim in a debate/speech required a lot of normalization and could not be found in the article body verbatim. Table 6 reports the evaluation results.

7.2.1 Experiment 1: BM25-based Baselines

We ran three experiments using BM25 to match the *Input* against *Title*, *VerClaim*, and *Title+VerClaim*. We can see in Table 6 that, just like for PolitiFact, using *VerClaim* performed better than using the article title, which is true for all evaluation measures; however, this time the margin was much smaller than it was for PolitiFact. We further noticed a small improvement for all MAP@k measures when matching against both the article *Title* and the *VerClaim*. Overall, BM25 is a very strong baseline for Snopes due to the high word overlap between the input claims and the verified claims (also, compared to PolitiFact, as we have seen in Table 4 above).

7.2.2 Experiment 2: Semantic Matching

Based on the lessons learned from PolitiFact, for semantic matching, we only experimented with sentence-BERT. We can see in Table 6 that this yielded results that were lower than for BM25 by a margin of at least 0.10 absolute for almost every reported measure; yet, this margin is smaller than for PolitiFact. For these experiments, once again matching against the verified claim outperformed matching against the article title by a sizable margin.

7.2.3 Experiment 3: BERT on Full Articles

As mentioned above, we did not perform matching of the input tweet against the article body, as this would easily give away the answer: the tweet can be found verbatim inside the target article.

For the purpose of comparison, we tried to filter out the text of the input tweet from the text of the article body before attempting the matching, but we still got unrealistically high results. Thus, ultimately we decided to abandon these experiments.

7.2.4 Experiment 4: Reranking

Finally, we trained a pairwise RankSVM model to re-rank the top-N results from *IR:VerClaim+Title*. For each claim-document pair in the top-N list, we extracted the scores from *IR:Title*, *IR:VerClaim*, *IR:VerClaim+Title*, *sentence-BERT:large:Title*, and *sentence-BERT:large:VerClaim*, as well as the corresponding reciprocal ranks for all target documents according to each of these scores. This is the same as for PolitiFact, except that now we do not use scores for matching the input to a document body. We can see in Table 6 that the best re-ranking model yielded sizable improvements over the best individual model by 0.09-0.18 points absolute on all evaluation measures.

Comparing the best re-ranking models for Snopes and PolitiFact, we can see that Snopes performed best when using a top-50 list, compared to top-100 for PolitiFact. We believe that this is due to the difference in performance of the retrieval models used to extract the top-N pairs: for Snopes, *IR:VerClaim+Title* has an MMR score of 0.664, while the best PolitiFact model, *IR:Body*, has an MRR score of 0.565. Thus, for Snopes we rerank an N-best list extracted by a stronger IR model, and thus there is no need to go that deep in the list.

8 Conclusions and Future Work

We have argued for the need to address detecting previously fact-checked claims as a task of its own right, which could be an integral part of automatic fact-checking, or a tool to help human fact-checkers or journalists. We have created specialized datasets, which we have released, together with our code, to the research community in order to enable further research. Finally, we have presented learning-to-rank experiments, demonstrating sizable improvements over state-of-the-art retrieval and textual similarity approaches.

In future work, we plan to extend this work to more datasets and to more languages. We further want to go beyond textual claims, and to take claim-image and claim-video pairs as an input.

Acknowledgments

This research is part of the Tanbih project,[14] which aims to limit the effect of ‘fake news,” disinformation, propaganda, and media bias by making users aware of what they are reading.

[14] http://tanbih.qcri.org/

References

Zeynep Akkalyoncu Yilmaz, Shengjin Wang, Wei Yang, Haotian Zhang, and Jimmy Lin. 2019a. Applying BERT to document retrieval with Birch. 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 '19, pages 19–24, Hong Kong, China.

Zeynep Akkalyoncu Yilmaz, Wei Yang, Haotian Zhang, and Jimmy Lin. 2019b. Cross-domain modeling of sentence-level evidence for document retrieval. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, EMNLP-IJCNLP '19, pages 3488–3494, Hong Kong, China.

Pepa Atanasova, Lluís Màrquez, Alberto Barrón-Cedeño, Tamer Elsayed, Reem Suwaileh, Wajdi Zaghouani, Spas Kyuchukov, Giovanni Da San Martino, and Preslav Nakov. 2018. Overview of the CLEF-2018 CheckThat! Lab on automatic identification and verification of political claims, Task 1: Check-worthiness. In *Working Notes of the Conference and Labs of the Evaluation Forum*, CLEF '18, Avignon, France.

Pepa Atanasova, Preslav Nakov, Georgi Karadzhov, Mitra Mohtarami, and Giovanni Da San Martino. 2019. Overview of the CLEF-2019 CheckThat! Lab on Automatic Identification and Verification of Claims. Task 1: Check-Worthiness. In *Working Notes of the Conference and Labs of the Evaluation Forum*, CLEF '19, Lugano, Switzerland.

Isabelle Augenstein, Christina Lioma, Dongsheng Wang, Lucas Chaves Lima, Casper Hansen, Christian Hansen, and Jakob Grue Simonsen. 2019. MultiFC: A real-world multi-domain dataset for evidence-based fact checking of claims. 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 '19, pages 4684–4696, Hong Kong, China.

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 Conference on Empirical Methods in Natural Language Processing*, EMNLP '18, pages 3528–3539, Brussels, Belgium.

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 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT '18, pages 21–27, New Orleans, LA, USA.

Alberto Barrón-Cedeño, Tamer Elsayed, Giovanni Da San Martino, Maram Hasanain, Reem Suwaileh, Fatima Haouari, and Preslav Nakov. 2020. CheckThat! at CLEF 2020: Enabling the automatic identification and verification of claims on social media. In *Proceedings of the European Conference on Information Retrieval*, ECIR '20, pages 499–507, Lisbon, Portugal.

Luisa Bentivogli, Ido Dagan, Hoa Trang Dang, Danilo Giampiccolo, and Bernardo Magnini. 2009. The fifth PASCAL recognizing textual entailment challenge. In *Proceedings of the Text Analysis Conference*, TAC '09, Gaithersburg, MD, USA.

Kevin R. Canini, Bongwon Suh, and Peter L. Pirolli. 2011. Finding credible information sources in social networks based on content and social structure. In *Proceedings of the IEEE International Conference on Privacy, Security, Risk, and Trust, and the IEEE International Conference on Social Computing*, SocialCom/PASSAT '11, pages 1–8, Boston, MA, USA.

Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. 2011. Information credibility on Twitter. In *Proceedings of the 20th International Conference on World Wide Web*, WWW '11, pages 675–684, Hyderabad, India.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation*, SemEval '17, pages 1–14, Vancouver, Canada.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, EMNLP '18, pages 169–174, Brussels, Belgium.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2019. TabFact: A large-scale dataset for table-based fact verification.

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):1–13.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT '19, pages 4171–4186, Minneapolis, MN, USA.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing*, IWP '05, Jeju Island, Korea.

Sebastian Dungs, Ahmet Aker, Norbert Fuhr, and Kalina Bontcheva. 2018. Can rumour stance alone predict veracity? In *Proceedings of the 27th International Conference on Computational Linguistics*, COLING '18, pages 3360–3370, Santa Fe, NM, USA.

Tamer Elsayed, Preslav Nakov, Alberto Barrón-Cedeño, Maram Hasanain, Reem Suwaileh, Pepa Atanasova, and Giovanni Da San Martino. 2019. CheckThat! at CLEF 2019: Automatic identification and verification of claims. In *Proceedings of the 41st European Conference on Information Retrieval*, ECIR '19, pages 309–315, Cologne, Germany.

Mohamed H. Gad-Elrab, Daria Stepanova, Jacopo Urbani, and Gerhard Weikum. 2019a. ExFaKT: A framework for explaining facts over knowledge graphs and text. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, WSDM '19, pages 87–95, Melbourne, Australia.

Mohamed H. Gad-Elrab, Daria Stepanova, Jacopo Urbani, and Gerhard Weikum. 2019b. Tracy: Tracing facts over knowledge graphs and text. In *The World Wide Web Conference*, WWW '19, pages 3516–3520, San Francisco, CA, USA.

Pepa Gencheva, Preslav Nakov, Lluís Màrquez, Alberto Barrón-Cedeño, and Ivan Koychev. 2017. A context-aware approach for detecting worth-checking claims in political debates. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*, RANLP '17, pages 267–276, Varna, Bulgaria.

Genevieve Gorrell, Elena Kochkina, Maria Liakata, Ahmet Aker, Arkaitz Zubiaga, Kalina Bontcheva, and Leon Derczynski. 2019. SemEval-2019 task 7: RumourEval, determining rumour veracity and support for rumours. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, SemEval '19, pages 845–854, Minneapolis, MN, USA.

Naeemul Hassan, Chengkai Li, and Mark Tremayne. 2015. Detecting check-worthy factual claims in presidential debates. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management*, CIKM '15, pages 1835–1838, Melbourne, Australia.

Naeemul Hassan, Gensheng Zhang, Fatma Arslan, Josue Caraballo, Damian Jimenez, Siddhant Gawsane, Shohedul Hasan, Minumol Joseph, Aaditya Kulkarni, Anil Kumar Nayak, Vikas Sable, Chengkai Li, and Mark Tremayne. 2017. ClaimBuster: The first-ever end-to-end fact-checking system. *Proc. VLDB Endow.*, 10(12):1945–1948.

Viet-Phi Huynh and Paolo Papotti. 2019. A benchmark for fact checking algorithms built on knowledge bases. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, pages 689–698, Beijing, China.

Georgi Karadzhov, Preslav Nakov, Lluís Màrquez, Alberto Barrón-Cedeño, and Ivan Koychev. 2017. Fully automated fact checking using external sources. In *Proceedings of the Conference on Recent Advances in Natural Language Processing*, RANLP '17, pages 344–353, Varna, Bulgaria.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv:1907.11692*.

Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J. Jansen, Kam-Fai Wong, and Meeyoung Cha. 2016. Detecting rumors from microblogs with recurrent neural networks. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, IJCAI '16, pages 3818–3824, New York, NY, USA.

Jing Ma, Wei Gao, and Kam-Fai Wong. 2019. Detect rumors on Twitter by promoting information campaigns with generative adversarial learning. In *Proceedings of the World Wide Web Conference*, WWW '19, pages 3049–3055, San Francisco, CA, USA.

Tsvetomila Mihaylova, Preslav Nakov, Lluís Màrquez, Alberto Barrón-Cedeño, Mitra Mohtarami, Georgi Karadjov, and James Glass. 2018. Fact checking in community forums. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, AAAI '18, pages 879–886, New Orleans, LA, USA.

Sebastião Miranda, David Nogueira, Afonso Mendes, Andreas Vlachos, Andrew Secker, Rebecca Garrett, Jeff Mitchel, and Zita Marinho. 2019. Automated fact checking in the news room. In *Proceedings of the World Wide Web Conference*, WWW '19, pages 3579–3583, San Francisco, CA, USA.

Subhabrata Mukherjee and Gerhard Weikum. 2015. Leveraging joint interactions for credibility analysis in news communities. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, pages 353–362, Melbourne, Australia.

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 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL-HLT '19, pages 78–83, Minneapolis, MN, USA.

Yixin Nie, Haonan Chen, and Mohit Bansal. 2019. Combining fact extraction and verification with neural semantic matching networks. In *Proceedings of* *the 33rd AAAI Conference on Artificial Intelligence*, AAAI '19, pages 6859–6866, Honolulu, HI, USA.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with BERT. *arXiv:1901.04085*.

Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-stage document ranking with BERT. *arXiv:1910.14424*.

Ayush Patwari, Dan Goldwasser, and Saurabh Bagchi. 2017. TATHYA: A multi-classifier system for detecting check-worthy statements in political debates. In *Proceedings of the 2017 ACM Conference on Information and Knowledge Management*, CIKM '17, pages 2259–2262, Singapore.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? 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 '19, pages 2463–2473, Hong Kong, China.

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, Perth, Australia.

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*, EMNLP '17, pages 2931–2937, Copenhagen, Denmark.

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 '19, pages 3980–3990, Hong Kong, China.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Prashant Shiadralkar, Alessandro Flammini, Filippo Menczer, and Giovanni Luca Ciampaglia. 2017. Finding streams in knowledge graphs to support fact checking. In *Proceedings of the 2017 IEEE International Conference on Data Mining*, ICDM '17, pages 859–864, New Orleans, LA, USA.

Andon Tchechmedjiev, Pavlos Fafalios, Katarina Boland, Malo Gasquet, Matthäus Zloch, Benjamin Zapilko, Stefan Dietze, and Konstantin Todorov. 2019. ClaimsKG: A knowledge graph of fact-checked claims. In *Proceedings of the 18th International Semantic Web Conference*, ISWC '19, pages 309–324, Auckland, New Zealand.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT '18, pages 809–819, New Orleans, LA, USA.

Slavena Vasileva, Pepa Atanasova, Lluís Màrquez, Alberto Barrón-Cedeño, and Preslav Nakov. 2019. It takes nine to smell a rat: Neural multi-task learning for check-worthiness prediction. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*, RANLP '19, pages 1229–1239, Varna, Bulgaria.

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

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, Brussels, Belgium.

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*, ACL '17, pages 422–426, Vancouver, Canada.

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*, NAACL-HLT '18, pages 1112–1122, New Orleans, LA, USA.

Wei Yang, Haotian Zhang, and Jimmy Lin. 2019. Simple applications of BERT for ad hoc document retrieval. *arXiv:1903.10972*.

Tauhid Zaman, Emily B. Fox, and Eric T. Bradlow. 2014. A Bayesian approach for predicting the popularity of tweets. *Ann. Appl. Stat.*, 8(3):1583–1611.

Dimitrina Zlatkova, Preslav Nakov, and Ivan Koychev. 2019. Fact-checking meets fauxtography: Verifying claims about images. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, EMNLP '19, pages 2099–2108, Hong Kong, China.