Assisting the Human Fact-Checkers: Detecting All Previously Fact-Checked Claims in a Document

Shaden Shaar[1]*, Nikola Georgiev[2], Firoj Alam[3], Giovanni Da San Martino[4], Aisha Mohamed[5], Preslav Nakov[6]
[1]Cornell University, [2]Sofia University, [3]Qatar Computing Research Institute, HBKU,
[4]University of Padova, [5]University of Wisconsin-Madison,
[6]Mohamed bin Zayed University of Artificial Intelligence
sshaar31@gmail.com, nikikg95@gmail.com, falam@hbku.edu.qa
dasan@math.unipd.it, ahmohamed2@wisc.edu, preslav.nakov@mbzuai.ac.ae

Abstract

Given the recent proliferation of false claims online, there has been a lot of manual fact-checking effort. As this is very time-consuming, human fact-checkers can benefit from tools that can support them and make them more efficient. Here, we focus on building a system that could provide such support. Given an input document, it aims to detect all sentences that contain a claim that can be verified by some previously fact-checked claims (from a given database). The output is a re-ranked list of the document sentences, so that those that can be verified are ranked as high as possible, together with corresponding evidence. Unlike previous work, which has looked into claim retrieval, here we take a document-level perspective. We create a new manually annotated dataset for this task, and we propose suitable evaluation measures. We further experiment with a learning-to-rank approach, achieving sizable performance gains over several strong baselines. Our analysis demonstrates the importance of modeling text similarity and stance, while also taking into account the veracity of the retrieved previously fact-checked claims. We believe that this research would be of interest to fact-checkers, journalists, media, and regulatory authorities.

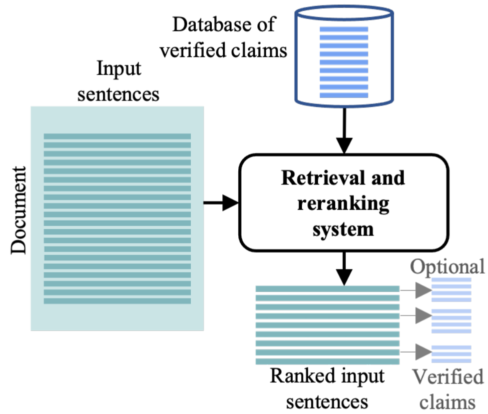

Figure 1: The architecture of our system. Given an input document, it aims to detect all sentences that contain a claim that can be verified by some previously fact-checked claims (from a given database). The output is a re-ranked list of the document sentences, so that those that can be verified are ranked as high as possible, together with corresponding evidence.

1 Introduction

Recent years have brought us a proliferation of false claims, which spread fast online, especially in social media; in fact, much faster than the truth (Vosoughi et al., 2018). To deal with the problem, a number of fact-checking initiatives have been launched, such as FactCheck, FullFact, PolitiFact, and Snopes, where professional fact-checkers verify claims. Yet, manual fact-checking is very time-consuming and tedious.

Thus, automatic fact-checking has been proposed as an alternative (Li et al., 2016; Shu et al., 2017; Rashkin et al., 2017; Hassan et al., 2017; Vo and Lee, 2018; Lee et al., 2018; Li et al., 2018; Thorne and Vlachos, 2018; Lazer et al., 2018; Vosoughi et al., 2018; Zhang et al., 2020b; Alam et al., 2022; Nguyen et al., 2022). While it scales better and works faster, it lags behind in quality, credibility, transparency, and explainability.

Manual and automatic fact-checking can benefit from each other as automatic methods are trained on data that human fact-checkers produce, while human fact-checkers can be assisted by automatic tools. A middle ground between manual and automatic fact-checking is to verify an input claim by finding a previously fact-checked claim that allows us to make a true/false judgment on the veracity of the input claim. This is the problem we will explore below.

*Work done while author was at QCRI, HBKU.

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 2069–2080

Previous work has approached the problem at the sentence level: given an input *sentence/tweet*, produce a ranked list of relevant previously fact-checked claims that can verify it (Shaar et al., 2020a). However, this formulation does not factor in whether the factuality of the input *sentence/tweet* can be determined using the database of previously fact-checked claims, as it is formulated as a ranking task. For example, in a US presidential debate that has 1,300 sentences on average, only a small fraction would be verifiable using previously fact-checked claims from PolitiFact. Therefore, we target a more challenging reformulation at the *document* level, where the system needs to prioritize which sentences are most likely to be verifiable using the database of previously fact-checked claims. This is still a ranking formulation, but here we rank the sentences in the input document (by verifiability using the database of claims), as opposed to ranking database claims for one input sentence (by similarity with respect to that sentence).

In our problem formulation, given an input *document*, the system needs to detect all sentences that contain a claim that can be verified by a previously fact-checked claim (from a given database of such claims). The output is a re-ranked list of the document sentences, so that those that can be verified are ranked as high as possible, as illustrated in Figure 1. The system could optionally further provide a corresponding fact-checked claim (or a list of such claims) from the database as evidence. Note that we are interested in returning claims that would not just be relevant when fact-checking the claims in the input sentence, but also would be enough to decide on a verdict for its factuality.

This novel formulation of the problem would be of interest to fact-checkers not only when they are facing a new document to analyze, but also when they want to check whether politicians keep repeating claims that have been previously debunked, so that they can be approached for comments. It would also be of interest to journalists, as it could bring them a tool that can allow them to put politicians and public officials on the spot, e.g., during a political debate, a press conference, or an interview, by showing the journalist in real time which claims have been previously fact-checked and found false. Finally, media outlets would benefit from such tools for self monitoring and quality assurance, and so would regulatory authorities such as Ofcom.

Our contributions can be summarized as follows:

- We introduce a new real-world task formulation to assist fact-checkers, journalists, media, and regulators in finding which claims in a document have been previously fact-checked.
- We develop a new dataset for this task formulation, which consists of seven debates, 5,054 sentences, 16,636 target verified claims to match against, and 75,810 manually annotated sentence–verified claim pairs.
- We define new evaluation measures (variants of MAP), which are specifically tailored for our task.
- We address the problem using a learning-to-rank approach, and we demonstrate sizable performance gains over strong baselines.
- We offer analysis and discussion, which can facilitate future research.
- We release our data and code.[1]

2 Related Work

Disinformation, misinformation, and "fake news" thrive in social media. See (Lazer et al., 2018) and (Vosoughi et al., 2018) for a general discussion on the science of "fake news" and the process of proliferation of true and false news online. There have also been several interesting surveys, e.g., Shu et al. (2017) studied how information is disseminated and consumed in social media. Another survey by Thorne and Vlachos (2018) took a fact-checking perspective on "fake news" and related problems.

More relevant to the present work, Nakov et al. (2021a) studied how AI technology can assist professional fact-checkers, and pointed to the following research problems: (*i*) identifying claims worth fact-checking, (*ii*) detecting relevant previously fact-checked claims, (*iii*) retrieving relevant evidence to fact-check a claim, and (*iv*) actually verifying the claim.

The vast majority of previous work has focused on the latter problem, while the other three problems remain understudied, even though there is an awareness that they are integral steps of an end-to-end automated fact-checking pipeline (Vlachos and Riedel, 2014; Hassan et al., 2017).

[1] `https://github.com/firojalam/assisting-fact-checking`

This situation is gradually changing, and the research community has recently started paying more attention to all four problems, in part thanks to the emergence of evaluation campaigns that feature all steps such as the CLEF CheckThat! lab.

Shaar et al. (2020a) proposed a *claim-focused* task formulation, and released two datasets: one based on PolitiFact, and another one based on Snopes. They had a ranking formulation: given a claim, they asked to retrieve a ranked list of previously fact-checked claims from a given database of such claims; the database included the verified claims together with corresponding articles. One can argue that this formulation falls somewhere between (*ii*) detecting relevant previously fact-checked claims and (*iii*) retrieving relevant evidence to fact-check a claim. The same formulation was adopted at the CLEF CheckThat! lab in 2020, where the focus was on tweets, and in 2021-2022, which featured both tweets and political debates (Barrón-Cedeño et al., 2020; Shaar et al., 2020b; Barrón-Cedeño et al., 2020; Nakov et al., 2021b; Shaar et al., 2021; Nakov et al., 2022a,b).

The best systems at the CLEF CheckThat! 2021 lab used BM25 retrieval, semantic similarity using embeddings, and reranking (Chernyavskiy et al., 2021; Mihaylova et al., 2021; Pritzkau, 2021). A follow-up work used a batch softmax contrastive loss to better fine-tune BERT for the task (Chernyavskiy et al., 2022).

It has been further shown that it is important to match not only against the target claim, but also using the full text of the associated article that fact-checkers wrote to explain their verdict. Thus, in a follow-up work, Shaar et al. (2022) focused on modeling the context when checking an input sentence from a political debate, both on the source side and on the target side, e.g., by looking at neighboring sentences and using co-reference resolution.

Sheng et al. (2021) proposed a re-ranker based on memory-enhanced transformers for matching (MTM) to rank fact-checked articles using key sentences selected using lexical, semantic and pattern-based similarity. Si et al. (2021) modeled claim-matching using topic-aware evidence reasoning and stance-aware aggregation, which model semantic interaction and topical consistency to learn latent evidence representation. Kazemi et al. (2021) developed two datasets (one consisting of *claim-like statements* and the other one using annotation of *claim similarity*) covering four languages.

Jiang et al. (2021) used sequence-to-sequence transformer models for sentence selection and label prediction. Wan et al. (2021) proposed a deep Q-learning network, i.e., a reinforcement learning approach, which computes candidate pairs of precise evidence and their labels, and then uses post-processing to refine the candidate pairs.

Vo and Lee (2020) looked into multimodality. They focused on tweets that discuss images and tried to detect the corresponding verified claim by matching both the text and the image against the images in the verified claim's article. They mined their dataset from pairs of tweets and corresponding fact-checking articles proposed by Twitter users as a response. Hardalov et al. (2022) used a similar crowd-checking idea, and further proposed how to learn from potentially noisy data.

Finally, the task was also addressed in a reverse formulation, i.e., given a database of fact-checked claims (e.g., a short list of common misconceptions about COVID-19), find social media posts that make similar claims (Hossain et al., 2020).

Unlike the above work, our input is a *document*, and the goal is to detect all sentences that contain a claim that can be verified by some previously fact-checked claim (from a given database).

3 Task Definition

We define the task as follows (see also Figure 1):

Given an input <u>document</u> and a database of previously fact-checked claims, produce a ranked list of its sentences, so that those that contain claims that can be verified by a claim from the database are ranked as high as possible. We further want the system to be able to point to the database claims that <u>verify</u> a claim in an input sentence.

Note that we want the *Input* sentence to be verified as true/false, and thus we want to skip matches against *Verified* claims with labels of unsure veracity such as *half-true*. Note also that solving this problem requires going beyond stance, i.e., whether a previously fact-checked claim *agrees*/*disagrees* with the input sentence (Miranda et al., 2019). In certain cases, other factors might also be important, such as, (*i*) whether the two claims express the same degree of specificity, (*ii*) whether they are made by the same person and during the same time period, (*iii*) whether the verified claim is true/false or is of mixed factuality, etc. Table 1 shows some examples.

No.	*Input* Sentence	*Verified* Claim	Label & Date	Stance	Verdict
1	*But the Democrats, by the way, are very weak on immigration.*	**Donald Trump:** The weak illegal immigration policies of the Obama Admin. allowed bad MS 13 gangs to form in cities across U.S. We are removing them fast!	*False*, stated on April 18, 2017	*agree*	Unknown
2	*ICE we're getting MS13 out by the thousands.*	**Donald Trump:** Says of MS13 gang members, "We are getting them out of our country by the thousands."	*Mostly-False*, stated on May 15, 2018	*agree*	False
3	*ICE we're getting MS13 out by the thousands.*	**Donald Trump:** I have watched ICE liberate towns from the grasp of MS13.	*False*, stated on June 30, 2018	*agree*	Unknown
4	*We have one of the highest business tax rates anywhere in the world, pushing jobs and wealth out of our country.*	**Barack Obama:** "There are so many loopholes ... our businesses pay effectively one of the lowest tax rates in the world."	*Half-True*, stated on September 26, 2008	*disagree*	Unknown

Table 1: Example sentences from Donald Trump's interview with *Fox and Friends* on June 6, 2018.

4 Dataset

4.1 Background

We construct a dataset using fact-checked claims from PolitiFact,[2] which focuses on claims by politicians. For each fact-checked claim, there is a factuality label and an article explaining the reason for assigning that label. PolitiFact further publishes commentaries that highlight some of the claims made in a debate or speech, with links to fact-checking articles about these claims from their website. These commentaries were used in previous work as a way to obtain a mapping from *Input* sentences in a debate/speech to *Verified* claims. For example, Shaar et al. (2020a) collected 16,636 *Verified* claims and 768 *Input–Verified* claim pairs from 70 debates and speeches, together with the transcript of the target event. For each *Verified* claim, they released *VerifiedStatement*, *TruthValue* {*Pants-on-Fire!*, *False*, *Mostly-False*, *Half-True*, *Mostly-True*, *True*}, *Title* and *Body*.

The above dataset has high precision, and it is suitable for their formulation of the task: given a sentence (one of the 768 ones), identify the correct claim that verifies it (from the set of 16,636 *Verified* claims). However, it turned out not to be suitable for our purposes due to recall issues: missing links between *Input* sentences in the debate/speech and the set of *Verified* claims. This is because PolitiFact journalists were not interested in making an exhaustive list of all possible correct mappings between *Input* sentences and *Verified* claims in their database; instead, they only pointed to some such links, which they wanted to emphasize.

Moreover, if the debate made some claim multiple times, they would include a link for only one of these instances (or they would skip the claim altogether). Moreover, if the claims made in a sentence are verified by multiple claims in the database, they might only include a link to one of these claims (or to none).

However, we have a document-level task, where identifying sentences that can be verified using a database of fact-checked claims is our primary objective (while returning the matching claims is secondary), we need not only high precision, but also high recall for the *Input–Verified* claim pairs.

4.2 Our Dataset

We manually checked and *re-annotated* seven debates from the dataset of Shaar et al. (2020a) by linking *Verified* claims from PolitiFact to the *Input* sentences in the transcript. This includes 5,054 sentences, and ideally, we would have wanted to compare each of them against each of the 16,636 *Verified* claims, which would have resulted in a huge and very imbalanced set of *Input–Verified* pairs: $5,054 \times 16,636 = 84,078,344$. Thus, we decided to pre-filter the *Input* sentences and the *Input–Verified* claim pairs. The process is sketched in Figure 2 and described in more detail below.

4.3 Phase 1: *Input* Sentence Filtering

Not all sentences in a speech/debate contain a verifiable factual claim, especially when uttered in a live setting. In speeches, politicians would make a claim and then would proceed to provide the numbers and the anecdotes to emphasize and to create an emotional connection with the audience.

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

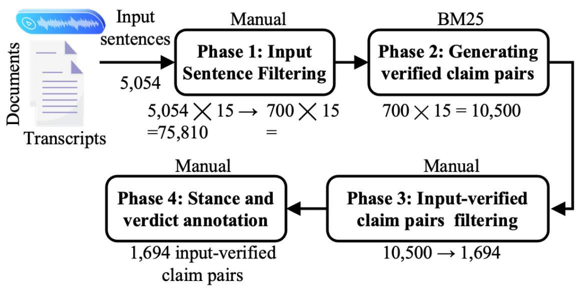

Figure 2: Data preparation pipeline.

In our case, we only need to focus on claims. We also know that not all claims are important enough to be fact-checked. Thus, we follow (Konstantinovskiy et al., 2021) as guidance to define which *Input* sentences are worth fact-checking. Based on this definition, positive examples include, but are not limited to (*a*) stating a definition, (*b*) mentioning a quantity in the present or in the past, (*c*) making a verifiable prediction about the future, (*d*) referencing laws, procedures, and rules of operation, or (*e*) implying correlation or causation (such correlation/causation needs to be explicit). Negative examples include personal opinions and preferences, among others. In this step, three annotators independently made judgments about the *Input* sentences for check-worthiness (i.e., check-worthy *vs.* not check-worthy), and we only rejected a sentence if all three annotators judged it to be not check-worthy. As a result, we reduced the number of *input sentences* that need further manual checking from 5,054 to 700.

4.4 Phase 2: Generating *Input–Verified* Pairs

Next, we indexed the *Verified* claims and we queried with the *Input* sentence using BM25 to retrieve 15 *Verified* claims per *Input* sentence. As a result, we managed to reduce the number of *pairs* to check from $700 \times 16,636 = 11,645,200$ to just $700 \times 15 = 10,500$.

4.5 Phase 3: *Input–Verified* Pairs Filtering

Then, we manually went through the 10,500 *Input–Verified* pairs, and we filtered out the ones that were incorrectly retrieved by the BM25 algorithm. Again, we were aiming for high recall, and thus we only rejected a pair if all three out of the three annotators independently proposed to reject it. As a result, the final number of *pairs* to check git reduced to just 1,694.

4.6 Phase 4: Stance and Verdict Annotation

As in the previous phase, three annotators manually annotated the 1,694 *Input–Verified* pairs with stance and verdict labels using the following label inventory:

- **stance**: *agree*, *disagree*, *unrelated*, *not–claim*;
- **verdict**: *true*, *false*, *unknown*, *not–claim*.

The label for **stance** is *agree* if the *Verified* claim agrees with the *Input* claim, *disagree* if it opposes it, and *unrelated* if there is no *agree/disagree* relation (this includes truly unrelated claims or related but without agreement/disagreement, e.g., discussing the same topic).

The **verdict** is *true/false* if the *Input* sentence makes a claim whose veracity can be determined to be *true/false* based on the paired *Verified* claim and its veracity label; it is *unknown* otherwise. The veracity can be unknown for various reasons, e.g., (*i*) the *Verified* claim states something (a bit) different, (*ii*) the two claims are about different events, (*iii*) the veracity label of the *Verified* claim is ambiguous. We only need the verdict annotation to determine whether the *Input* sentence is verifiable; yet, we use the stance to construct suitable *Input–Verified* claim pairs.

4.7 Final Dataset

Our final dataset consists of 5,054 *Input* sentences, and 75,810 *Input–Verified* claim pairs. This includes 125 *Input* sentences that can be verified using a database of 16,663 fact-checked claims, and 198 *Input–Verified* claim pairs where the *Verified* claim can verify the *Input* sentence (as some *Input* sentences can be verified by more than one *Verified* claim). Table 2 reports some statistics about each transcript, and it also shows overall statistics (in the last row).

4.8 Annotation and Annotators' Agreement

Note that each *Input–Verified* claim pair was annotated by three annotators: one male and two females, with BSc and PhD degrees. The disagreements were resolved by majority voting, and, if this was not possible, in a discussion with additional consolidators. We measured the inter-annotator agreement on phase 4 (phases 1 and 3 aimed for high recall rather than agreement). We obtained a Fleiss Kappa (κ) of 0.416 for stance and of 0.420 for the verdict, both corresponding to moderate level of agreement.

Date	Event	# Topic	Sent.	Sent.-Var. Pairs	# Stance-Input	# Stance-pairs	# Verdict-Input	# Verdict-pairs
2017-08-03	Rally Speech	3-4	291	4,365	34	62	20	32
2017-08-22	Rally Speech	5+	792	11,880	50	116	23	40
2018-04-26	Interview	5+	597	8,955	28	52	17	32
2018-05-25	Naval Grad. Speech	1-2	279	4,185	14	19	4	5
2018-06-12	North Korea Summit Speech	1-2	1,245	18,675	29	45	15	15
2018-06-15	Interview	3-4	814	12,210	24	36	11	17
2018-06-28	Rally Speech	5+	1,036	15,540	49	82	35	57
Total			5,054	75,810	228	412	125	198

Table 2: **Statistics about our dataset:** number of sentences in each transcript, and distribution of clear stance (*agree* + *disagree*) and clear verdict (true + false) labels. The number of topics was manually decided by looking at the keywords detected in each transcript. *Sent.* is the number of input sentences, and *Sent.-Var. Pairs* is the number of input sentences with top-15 verified claim pairs.

Politifact Truth Value	True/False	Unknown
Pants on Fire!	24	191
FALSE	76	382
Mostly–False	44	312
Half–True	2	260
Mostly–True	42	227
TRUE	11	85

Table 3: **Distribution of the labels:** *Input–Verified* pairs with a true/false verdict vs. the *TruthValue* for *Verified* claim from PolitiFact.

5 Evaluation Measures

Given a document, the goal is to rank its sentences, so that those that can be verified (i.e., with a true/-false verdict; *Verdict-Input* in Table 2) are ranked as high as possible, and also to provide a relevant *Verified* claim (i.e., one that could justify the verdict; *Verdict-pairs* in Table 2). This is a (double) ranking task, and thus we use ranking evaluation measures based on Mean Average Precision (MAP). First, let us recall the standard AP:

$$AP = \frac{\sum_{k=1}^{n} P_1(k) \times rel(k)}{rel.sentences}, \qquad (1)$$

where $P_1(k)$ is the precision at a cut-off k in the list, $rel(k)$ is 1 if the k-th ranked sentence is relevant (i.e., has either a true or a false verdict), and *rel. sentences* is the number of *Input* sentences that can be verified in the transcript.

We define more strict AP measures, AP_H^r, AP_0^r, and $AP_{0.5}^r$, which only give credit for an *Input* sentence with a known verdict, if also a corresponding *Verified* claim is correctly identified:

$$AP_H^r = \frac{\sum_{k=1}^{n} P_1^r(k) \times rel_H^r(k)}{rel.sentences} \qquad (2)$$

where $rel_H^r(k)$ is 1 if the k-th ranked *Input* sentence is relevant and at least one relevant *Verified* claim was retrieved in the top-r *Verified* claim list.

$$AP_0^r = \frac{\sum_{k=1}^{n} P_0^r(k) \times rel(k)}{rel.\ sentences} \qquad (3)$$

$$AP_{0.5}^r = \frac{\sum_{k=1}^{n} P_{0.5}^r(k) \times rel(k)}{rel.\ sentences} \qquad (4)$$

where $P_m^r(k)$, is precision at cut-off k, so that it increments by m, if **none** of the relevant *Verified* claim was retrieved in the top-r *Verified* claim list; otherwise, it increments by 1.

Note that the simple AP can also be represented as AP_1^r, as it increments by 1 regardless of whether a relevant *Verified* claim is in the top-r of the list of *Verified* claims.

We compute MAP, MAP_H^r, MAP_0^r, and $MAP_{0.5}^r$ by averaging AP, AP_H^r, AP_0^r, and $AP_{0.5}^r$, respectively, over the test transcripts.

We also compute MAP_{inner} by averaging the AP_{inner} on the *Verified* claims: we compute AP_{inner} for a given *Input* sentence, by scoring the rankings of the retrieved *Verified* claims as in the task presented in (Shaar et al., 2020a).

6 Model

The task we are trying to solve has two subtasks. The *first* sorts the *Input* sentences in the transcript in a way, so that the *Input* sentences that can be verified using the database are on top. The *second* one consists of retrieving a list of matching *Verified* claims for a given *Input* sentence. While we show experiments for both subtasks, our main focus is on solving the first one.

6.1 *Input–Verified* Pair Representation

In order to rank the *Input* sentences from the transcript, we need to find ways to represent them, so that we would have information about whether the database of *Verified* claims can indeed verify some claim from the *Input* sentence.

To do that, we propose to compute multiple similarity measures between all possible *Input–Verified* pairs, where we can match the *Input* sentence against the *VerifiedStatement*, the *Title*, and the *Body* of the verified claims' fact-checking article in PolitiFact.

- **BM25 (Robertson and Zaragoza, 2009):** These are BM25 scores when matching the *Input* sentence against the *VerifiedStatement*, the *Title*, and the *Body*, respectively (3 features);
- **NLI Score (Nie et al., 2020):** These are posterior probabilities for NLI over the labels {*entailment*, *neutral*, *contradiction*} between the *Input* sentence and the *VerifiedStatement* (3 features);
- **BERTScore (Zhang et al., 2020a):** F1 score from the BERTScore similarity scores between the *Input* sentence and the *VerifiedStatement* (1 feature);
- **Sentence-BERT (SBERT) (Reimers and Gurevych, 2019):** Cosine similarity for sentence-BERT-large embedding of the *Input* sentence compared to the embedding for the *VerifiedStatement*, the *Title*, and the *Body*. Since the *Body* is longer, we obtain the cosine similarity between the *Input* sentence vs. each sentence from the *Body*, and we only keep the four highest scores (6 features);
- **SimCSE (Gao et al., 2021):** Similarly to SBERT, we compute the cosine similarity between the SimCSE embeddings of the *Input* sentence against the *VerifiedStatement*, the *Title*, and the *Body*. Again, we use the top-4 scores when matching against the *Body* sentences (6 features: 1 from the *VerifiedStatement* + 1 from the *Title* + 4 from the *Body*).

6.2 Single-Score Baselines

Each of the above scores, e.g., SBERT, can be calculated for each *Input–Verified* claim pair. For a given *Input* sentence, this makes 16,663 scores (one for each *Verified* from the database), and as a baseline, we assign to the *Input* sentence the maximum over these scores. Then, we sort the sentences of the input document based on these scores, and we evaluate the resulting ranking.

Experiment	**MAP**$_{inner}$
BERTScore (F1) on *VerifiedStatement*	0.638
NLI (Entl) on *VerifiedStatement*	0.574
NLI (Neut) on *VerifiedStatement*	0.112
NLI (Contr) on *VerifiedStatement*	0.025
NLI (Entl+Contr) on *VerifiedStatement*	0.553
SimCSE on *Title*	0.220
SimCSE on *VerifiedStatement*	0.451
SimCSE on *Body*	0.576
SBERT on *Title*	0.165
SBERT on *VerifiedStatement*	0.531
SBERT on *Body*	0.649
BM25 on *VerifiedStatement*	0.316
BM25 on *Body*	**0.892**
BM25 on *Title*	0.145

Table 4: **Preliminary *Verified* Claim retrieval experiments** on the annotations obtained from the PolitiFact dataset and the manually annotated pairs with *agree* or *disagree* stance.

6.3 Re-ranking Models

We performed preliminary experiments looking into how the above measures work for retrieving the correct *Verified* claim for an *Input* sentence for which there is at least one match in the *Verified* claims database. This corresponds to the sentence-level task of (Shaar et al., 2020a), but on our dataset, where we augment the matching *Input–Verified* pairs from their dataset with all the *Input–Verified* pairs with a stance of *agree* or *disagree*. The results are shown in Table 4. We can see that *BM25 on Body* yields the best overall MAP score, which matches the observations in (Shaar et al., 2020a).

RankSVM for *Verified* Claim Retrieval Since now we know that the best *Verified* claim retriever uses *BM25 on Body*, we use it to retrieve the top-N *Verified* claims for the *Input* sentence, and then we calculate the above 19 similarity measures for each candidate in this top-N list. Afterwards, we concatenate the scores for these top-N candidates. Thus, we create a feature vector of size $19 \times N$ for each *Input* sentence. For example, a top-3 experiment uses for each *Input* sentence a feature vector of size $19 \times 3 = 57$, which represents each similarity measure based on the top-3 *Verified* claims retrieved by *BM25 on Body*. Then, we train a RankSVM model using this feature representation.

RankSVM–Max Instead of concatenating the 19-dimensional vectors for the top-N candidates, we take the maximum over these candidates for each feature to obtain a new 19-dimensional vector. Then, we train a RankSVM model like before.

Experiment	MAP	MAP^1_0	MAP^3_0	$\text{MAP}^1_{0.5}$	$\text{MAP}^3_{0.5}$	MAP^1_H	MAP^3_H
Baselines: Single Scores							
BERTScore (F1) on *VerifiedStatement*	0.076	0.046	0.050	0.061	0.063	0.034	0.038
NLI (Entl) on *VerifiedStatement*	0.035	0.025	0.029	0.030	0.032	0.017	0.023
NLI (Neut) on *VerifiedStatement*	0.036	0.001	0.003	0.019	0.020	0.000	0.001
NLI (Contr) on *VerifiedStatement*	0.051	0.001	0.001	0.026	0.026	0.000	0.000
NLI (Entl+Contr) on *VerifiedStatement*	0.041	0.005	0.007	0.023	0.024	0.002	0.003
SimCSE on *VerifiedStatement*	0.287	0.249	0.259	0.268	0.273	**0.208**	0.223
SimCSE on *Title*	0.242	0.144	0.213	0.193	0.227	0.093	0.172
SimCSE on *Body*	0.068	0.041	0.048	0.055	0.058	0.025	0.034
SBERT on *VerifiedStatement*	**0.303**	**0.245**	**0.284**	**0.274**	**0.294**	0.203	**0.251**
SBERT on *Title*	0.117	0.044	0.082	0.080	0.099	0.019	0.060
SBERT on *Body*	0.033	0.016	0.021	0.025	0.027	0.008	0.012
BM25 on *VerifiedStatement*	0.146	0.107	0.122	0.127	0.134	0.086	0.100
BM25 on *Title*	0.084	0.047	0.049	0.066	0.067	0.031	0.034
BM25 on *Body*	0.155	0.130	0.144	0.143	0.150	0.107	0.132
RankSVM for Retrieved *Verified* Claims (using BM25 on *Body*)							
Top-1	0.382	0.357	0.373	0.369	0.378	0.310	0.352
Top-3	0.345	0.318	0.336	0.332	0.341	0.278	0.319
Top-5	0.362	0.335	0.353	0.349	0.357	0.292	0.335
Top-10	**0.404**	**0.364**	**0.391**	**0.384**	**0.398**	**0.313**	**0.368**
Top-20	0.400	0.346	0.377	0.373	0.388	0.291	0.352
Top-30	0.357	0.310	0.339	0.333	0.348	0.260	0.318
RankSVM–Max							
Top-1	0.411	0.299	0.390	0.355	0.401	0.253	0.364
Top-3	0.449	0.328	0.429	0.389	0.439	0.273	0.400
Top-5	0.482	0.349	0.464	0.416	0.473	0.291	0.436
Top-10	**0.491**	**0.394**	**0.473**	**0.443**	**0.482**	**0.320**	**0.445**
Top-20	0.488	0.381	0.470	0.434	0.479	0.310	0.439
Top-30	0.486	0.377	0.468	0.432	0.477	0.304	0.435
RankSVM–Max with Skipping Half-True *Verified* claims							
Top-1	0.467	0.353	0.442	0.410	0.455	0.287	0.417
Top-3	0.507	0.370	0.485	0.438	0.496	0.306	0.454
Top-5	**0.522**	0.379	**0.501**	0.451	**0.512**	0.316	**0.468**
Top-10	0.515	**0.401**	0.494	**0.458**	0.505	**0.323**	0.465
Top-20	0.504	0.350	0.481	0.427	0.493	0.293	0.447
Top-30	0.493	0.376	0.468	0.435	0.481	0.301	0.433

Table 5: **Verdict experiments:** Baseline and re-ranking experiments on the PolitiFact dataset. The results highlighted in **bold** are the best results for the particular sets of experiments. The underlined results are the best overall.

RankSVM–Max with Skipping Table 3 shows that almost all *Input–Verified* pairs for which the *TruthValue* of the *Verified* claim is Half–True eventually result in an *Input* sentence for which we cannot determine an actual verdict. This is to be expected as, if we cannot trust the veracity of the *Verified* claim, then even if the statement matches the *Input* sentence, we cannot determine its veracity. Thus, we further experiment with a variant of the **RankSVM–Max** model that skips any scores that belong to a Half–True *Verified* claim.

7 Experiments and Evaluation

We performed a 7-fold cross-validation, where we used 6 out of the 7 transcripts for training and the remaining one for testing. We first computed 19 similarity measures and then used them to test the baselines and to train pairwise learning-to-rank models. The results are shown in Table 5.

7.1 Baselines

First, we discuss the results for the baseline experiments.

We can see in Table 5 that Sentence-BERT and SimCSE, when computed on the *Verified* claims, perform best. An interesting observation can be made by comparing Table 4 and Table 5. In Table 4, we see that the best *Verified* claim retriever uses BM25 on *Body*; however, Table 5 shows poor results when we try to use BM25 to rerank *Input* sentences.

Moreover, while in Table 5 the best-performing model uses SBERT calculated on *VerifiedStatement*, Table 4 shows that the *Verified* retriever using that model performs quite poorly. Our investigation showed that this is because SBERT tends to yield high scores for *Verified* claims, even when there is no relevant *Verified* claim. Thus, it can be a matter of calibration.

Experiment	MAP	MAP_0^1	MAP_0^3	$\text{MAP}_{0.5}^1$	$\text{MAP}_{0.5}^3$	MAP_H^1	MAP_H^3
RankSVM–Max on Top-5 with Skipping	0.522	0.379	0.501	0.451	0.512	0.316	0.468
w/o BERTScore (F1)	0.499	0.376	0.480	0.437	0.489	0.313	0.450
w/o NLI Score (E, N, C)	0.475	0.330	0.451	0.402	0.463	0.279	0.423
w/o SimCSE	0.511	0.353	0.486	0.432	0.499	0.295	0.454
w/o SBERT	0.498	0.381	0.481	0.440	0.490	0.308	0.452
w/o BM25	0.497	0.343	0.473	0.420	0.485	0.287	0.441
w/o scores on *Title*	0.522	0.369	0.501	0.445	0.511	0.308	0.468
w/o scores on *VerifiedStatement*	0.311	0.242	0.293	0.276	0.302	0.198	0.268
w/o scores on *Body*	0.444	0.295	0.427	0.370	0.435	0.249	0.398

Table 6: **Ablation experiments for the verdict** on the best model from Table 5: RankSVM with Top-5 scores from all measures while skipping *half-true Verified* claims.

7.2 RankSVM for *Verified* Claims Retrieval

We trained a RankSVM on the 19 similarity measures computed for the top-N retrieved *Verified* claims, according to BM25, the best system on *Body*. We can see from Table 5 that using the RankSVM on the 19 measures improves the scores by up to 10 MAP points absolute. Moreover, the best model achieves a MAP score of 0.404.

7.3 RankSVM–Max

Using max-pooling instead of BM25-retrieved *Verified* claims yields huge improvements in MAP: from 0.404 to 0.491 using RankSVM on the top-10 scores from the 19 metrics.

A sizable improvement can be observed when we consider MAP_0^3, $\text{MAP}_{0.5}^3$ and MAP_H^3 from RankSVM for *Verified* claims retrieval. Note that, since there is a max over each measure independently, we no longer have a unified *Verified* suggestion, which is required to compute MAP_0, $\text{MAP}_{0.5}$, and MAP_H. Thus, to compute them, we use the best *Verified* claim retriever from Table 4, i.e., BM25 on *Body*.

7.4 RankSVM–Max with Skipping

The highest MAP score, 0.522, is achieved by the RankSVM that uses the top-5 scores from each measure while skipping the Half–True *Verified* claim scores. We can also conclude by looking at the other variants of the MAP score, e.g., MAP_H, that we can identify the *Input* sentences that need to be fact-checked and detect the correct *Verified* claims in the top-3 ranks.

7.5 Ablation Experiments

We performed an ablation study for the best-performing model in Table 5, by removing one feature at a time. We also excluded all scores based on *Title*, *VerifiedStatement*, and *Body*. The results are shown in Table 6.

We can see that the largest drops, and therefore the most important features, are the *VerifiedStatement* and *Body* scores, whereas without *Title* scores the model performs almost identically to the original. We also notice that although the NLI Score did not perform very well by itself (see the baselines in Table 5), it yields a significant drop, from 0.522 to 0.475 MAP points, when it is removed, which shows that it is indeed quite important.

8 Conclusion and Future Work

We introduced a new challenging real-world task formulation to assist fact-checkers, journalists, media, and regulatory authorities in finding which claims in a long document have been previously fact-checked. Given an input document, we aim to detect all sentences containing a claim that can be verified by some previously fact-checked claims (from a given database of previously fact-checked claims). We developed a new dataset for this task formulation, consisting of seven debates, 5,054 sentences, 16,636 target verified claims to match against, and 75,810 manually annotated sentence–verified claim pairs.

We further defined new evaluation measures (variants of MAP), which are better tailored for our task setup. We addressed the problem using learning-to-rank, and we demonstrated sizable performance gains over strong baselines. We offered analysis and discussion, which can facilitate future research, and we released our data and code.

In future work, we plan to focus more on detecting the matching claims, which was our second objective here. We also plan to explore other transformer architectures and novel ranking approaches such as multi-stage document ranking using monoBERT and duoBERT (Yates et al., 2021).

9 Limitations

We have developed a dataset and proposed and evaluated a model using data from PolitiFact, which consists of political statements. We have not evaluated our approach on other topics, e.g., factual claims appearing on social media, which is out of the scope of the present work.

Ethics and Broader Impact

Biases We note that there might be some biases in the data we use, as well as in some judgments for claim matching. These biases, in turn, will likely be exacerbated by the unsupervised models trained on them. This is beyond our control, as the potential biases in pre-trained large-scale transformers such as BERT and RoBERTa, which we use in our experiments.

Intended Use and Misuse Potential Our models can make it possible to put politicians on the spot in real time, e.g., during an interview or a political debate, by providing journalists with tools to do trustable fact-checking in real time. They can also save a lot of time to fact-checkers for unnecessary double-checking something that was already fact-checked. However, these models could also be misused by malicious actors. We, therefore, ask researchers to exercise caution.

Environmental Impact We would like to warn that the use of large-scale Transformers requires a lot of computations and the use of GPUs/TPUs for training, which contributes to global warming (Strubell et al., 2019). This is a bit less of an issue in our case, as we do not train such models from scratch; rather, we fine-tune them on relatively small datasets. Moreover, running on a CPU for inference, once the model is fine-tuned, is perfectly feasible, contributes much less to global warming.

References

Firoj Alam, Stefano Cresci, Tanmoy Chakraborty, Fabrizio Silvestri, Dimiter Dimitrov, Giovanni Da San Martino, Shaden Shaar, Hamed Firooz, and Preslav Nakov. 2022. A survey on multimodal disinformation detection. In *Proceedings of the 29th International Conference on Computational Linguistics*, COLING '20, pages 6625–6643, Gyeongju, Republic of Korea.

Alberto Barrón-Cedeño, Tamer Elsayed, Preslav Nakov, Giovanni Da San Martino, Maram Hasanain, Reem Suwaileh, Fatima Haouari, Nikolay Babulkov, Bayan Hamdan, Alex Nikolov, Shaden Shaar, and Zien Sheikh Ali. 2020. Overview of CheckThat! 2020: Automatic identification and verification of claims in social media. In *Proceedings of the 11th International Conference of the CLEF Association: Experimental IR Meets Multilinguality, Multimodality, and Interaction*, CLEF '2020, pages 215–236.

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

Anton Chernyavskiy, Dmitry Ilvovsky, Pavel Kalinin, and Preslav Nakov. 2022. Batch-softmax contrastive loss for pairwise sentence scoring tasks. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT '22, pages 116–126, Seattle, WA, USA.

Anton Chernyavskiy, Dmitry Ilvovsky, and Preslav Nakov. 2021. Aschern at CLEF CheckThat! 2021: Lambda-calculus of fact-checked claims. In *Proceedings of the Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum*, volume 2936 of *CEUR Workshop Proceedings*, pages 484–493, Bucharest, Romania.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, EMNLP '21, pages 6894–6910, Punta Cana, Dominican Republic.

Momchil Hardalov, Anton Chernyavskiy, Ivan Koychev, Dmitry Ilvovsky, and Preslav Nakov. 2022. CrowdChecked: Detecting previously fact-checked claims in social media. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing*, AACL-IJCNLP '22.

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.

Tamanna Hossain, Robert L. Logan IV, Arjuna Ugarte, Yoshitomo Matsubara, Sean Young, and Sameer Singh. 2020. COVIDLies: Detecting COVID-19 misinformation on social media. In *Proceedings of the 1st Workshop on NLP for COVID-19 (Part 2) at EMNLP 2020*.

Kelvin Jiang, Ronak Pradeep, and Jimmy Lin. 2021. Exploring listwise evidence reasoning with T5 for fact verification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, ACL-IJCNLP '21, pages 402–410.

Ashkan Kazemi, Kiran Garimella, Devin Gaffney, and Scott Hale. 2021. Claim matching beyond English to scale global fact-checking. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, ACL-IJCNLP '21, pages 4504–4517.

Lev Konstantinovskiy, Oliver Price, Mevan Babakar, and Arkaitz Zubiaga. 2021. Towards automated factchecking: Developing an annotation schema and benchmark for consistent automated claim detection. *Digital Threats: Research and Practice*, 2(2):1–16.

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

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*, EMNLP '18, pages 1133–1138, Brussels, Belgium.

Sizhen Li, Shuai Zhao, Bo Cheng, and Hao Yang. 2018. An end-to-end multi-task learning model for fact checking. In *Proceedings of the First Workshop on Fact Extraction and VERification*, FEVER '18, pages 138–144, Brussels, Belgium.

Yaliang Li, Jing Gao, Chuishi Meng, Qi Li, Lu Su, Bo Zhao, Wei Fan, and Jiawei Han. 2016. A survey on truth discovery. *SIGKDD Explor. Newsl.*, 17(2):1–16.

Simona Mihaylova, Iva Borisova, Dzhovani Chemishanov, Preslav Hadzhitsanev, Momchil Hardalov, and Preslav Nakov. 2021. DIPS at CheckThat! 2021: Verified claim retrieval. In *Proceedings of the Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum*, volume 2936 of *CEUR Workshop Proceedings*, pages 558–571, Bucharest, Romania.

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.

Preslav Nakov, Alberto Barrón-Cedeño, Giovanni Da San Martino, Firoj Alam, Julia Maria Struß, Thomas Mandl, Rubén Míguez, Tommaso Caselli, Mucahid Kutlu, Wajdi Zaghouani, Chengkai Li, Shaden Shaar, Gautam Kishore Shahi, Hamdy Mubarak, Alex Nikolov, Nikolay Babulkov, Yavuz Selim Kartal, and Javier Beltrán. 2022a. The CLEF-2022 CheckThat! lab on fighting the COVID-19 infodemic and fake news detection. In *Proceedings of the 44th European Conference on IR Research: Advances in Information Retrieval*, ECIR '22, pages 416–428, Stavanger, Norway.

Preslav Nakov, David Corney, Maram Hasanain, Firoj Alam, Tamer Elsayed, Alberto Barrón-Cedeño, Paolo Papotti, Shaden Shaar, and Giovanni Da San Martino. 2021a. Automated fact-checking for assisting human fact-checkers. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence*, IJCAI '21, pages 4551–4558.

Preslav Nakov, Giovanni Da San Martino, Firoj Alam, Shaden Shaar, Hamdy Mubarak, and Nikolay Babulkov. 2022b. Overview of the CLEF-2022 CheckThat! lab task 2 on detecting previously fact-checked claims. In *Working Notes of CLEF 2022—Conference and Labs of the Evaluation Forum*, CLEF '2022, Bologna, Italy.

Preslav Nakov, Giovanni Da San Martino, Tamer Elsayed, Alberto Barrón-Cedeño, Rubén Míguez, Shaden Shaar, Firoj Alam, Fatima Haouari, Maram Hasanain, Nikolay Babulkov, Alex Nikolov, Gautam Kishore Shahi, Julia Maria Struß, and Thomas Mandl. 2021b. The CLEF-2021 CheckThat! lab on detecting check-worthy claims, previously fact-checked claims, and fake news. In *Proceedings of the 43rd European Conference on Information Retrieval*, ECIR '21, pages 639–649, Lucca, Italy.

Van-Hoang Nguyen, Kazunari Sugiyama, Preslav Nakov, and Min-Yen Kan. 2022. FANG: Leveraging social context for fake news detection using graph representation. *Commun. ACM*, 65(4):124–132.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, ACL '20, pages 4885–4901.

Albert Pritzkau. 2021. NLytics at CheckThat! 2021: Multi-class fake news detection of news articles and domain identification with RoBERTa - a baseline model. In *Proceedings of the Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum*, volume 2936 of *CEUR Workshop Proceedings*, pages 572–581, Bucharest, Romania.

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 3982–3992, Hong Kong, China.

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

Shaden Shaar, Firoj Alam, Giovanni Da San Martino, and Preslav Nakov. 2022. The role of context in detecting previously fact-checked claims. In *Findings of the Association for Computational Linguistics: NAACL-HLT 2022*, NAACL-HLT '22, pages 1619–1631, Seattle, WA, USA.

Shaden Shaar, Nikolay Babulkov, Giovanni Da San Martino, and Preslav Nakov. 2020a. That is a known lie: Detecting previously fact-checked claims. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, ACL '20, pages 3607–3618.

Shaden Shaar, Fatima Haouari, Watheq Mansour, Maram Hasanain, Nikolay Babulkov, Firoj Alam, Giovanni Da San Martino, Tamer Elsayed, and Preslav Nakov. 2021. Overview of the CLEF-2021 CheckThat! lab task 2 on detecting previously fact-checked claims in tweets and political debates. In *Working Notes of CLEF 2021—Conference and Labs of the Evaluation Forum*, CLEF '2021.

Shaden Shaar, Alex Nikolov, Nikolay Babulkov, Firoj Alam, Alberto Barrón-Cedeño, Tamer Elsayed, Maram Hasanain, Reem Suwaileh, Fatima Haouari, Giovanni Da San Martino, and Preslav Nakov. 2020b. Overview of CheckThat! 2020 English: Automatic identification and verification of claims in social media. In *Proceedings of the 11th International Conference of the CLEF Association: Experimental IR Meets Multilinguality, Multimodality, and Interaction*, CEUR Workshop Proceedings. CEUR-WS.org.

Qiang Sheng, Juan Cao, Xueyao Zhang, Xirong Li, and Lei Zhong. 2021. Article reranking by memory-enhanced key sentence matching for detecting previously fact-checked claims. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, ACL-IJCNLP '21, pages 5468–5481.

Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. Fake news detection on social media: A data mining perspective. *SIGKDD Explor. Newsl.*, 19(1):22–36.

Jiasheng Si, Deyu Zhou, Tongzhe Li, Xingyu Shi, and Yulan He. 2021. Topic-aware evidence reasoning and stance-aware aggregation for fact verification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, ACL-IJCNLP '21, pages 1612–1622.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics*, ACL '19, pages 3645–3650, Florence, Italy.

James Thorne and Andreas Vlachos. 2018. Automated fact checking: Task formulations, methods and future directions. In *Proceedings of the 27th International Conference on Computational Linguistics*, COLING '18, pages 3346–3359, Santa Fe, NM, USA.

Andreas Vlachos and Sebastian Riedel. 2014. Fact checking: Task definition and dataset construction. In *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*, pages 18–22, Baltimore, MD, USA.

Nguyen Vo and Kyumin Lee. 2018. The rise of guardians: Fact-checking URL recommendation to combat fake news. In *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '18, page 275–284, Ann Arbor, MI, USA.

Nguyen Vo and Kyumin Lee. 2020. Where are the facts? Searching for fact-checked information to alleviate the spread of fake news. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, EMNLP '20, pages 7717–7731.

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

Hai Wan, Haicheng Chen, Jianfeng Du, Weilin Luo, and Rongzhen Ye. 2021. A DQN-based approach to finding precise evidences for fact verification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, ACL-IJCNLP '21, pages 1030–1039.

Andrew Yates, Rodrigo Nogueira, and Jimmy Lin. 2021. Pretrained transformers for text ranking: BERT and beyond. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, WSDM '21, pages 1154–1156.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020a. are: Evaluating text generation with BERT. In *Proceedings of the International Conference on Learning Representations*, ICLR '20.

Wenxuan Zhang, Yang Deng, Jing Ma, and Wai Lam. 2020b. AnswerFact: Fact checking in product question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, EMNLP '20, pages 2407–2417.