Relevant Document Discovery for Fact-Checking Articles

Xuezhi Wang, Cong Yu, Simon Baumgartner, Flip Korn
Google Research NYC
{xuezhiw,congyu,simonba,flip}@google.com

ABSTRACT

With the support of major search platforms such as Google and Bing, fact-checking articles, which can be identified by their adoption of the schema.org ClaimReview structured markup, have gained widespread recognition for their role in the fight against digital misinformation. A claim-relevant document is an online document that addresses, and potentially expresses a stance towards, some claim. The claim-relevance discovery problem, then, is to find claim-relevant documents. Depending on the verdict from the fact check, claim-relevance discovery can help identify online misinformation.

In this paper, we provide an initial approach to the claim-relevance discovery problem by leveraging various information retrieval and machine learning techniques. The system consists of three phases. First, we retrieve candidate documents based on various features in the fact-checking article. Second, we apply a relevance classifier to filter away documents that do not address the claim. Third, we apply a language feature based classifier to distinguish documents with different stances towards the claim. We experimentally demonstrate that our solution achieves solid results on a large-scale dataset and beats state-of-the-art baselines. Finally, we highlight a rich set of case studies to demonstrate the myriad of remaining challenges and that this problem is far from being solved.

CCS CONCEPTS

• **Information systems** → *Data mining*; *Clustering and classification*; • **Computing methodologies** → *Classification and regression trees*;

KEYWORDS

Fact Checking; Digital Misinformation; Claim-Relevance Discovery

1 INTRODUCTION

Fact checking[1] is a type of journalism where a journalist examines claims published by others for the veracity and accuracy of those claims. The claims can range from a statement made by a politician to a story reported by another publisher to a rumor circulating in social networks. The goal of fact checking is to provide a verdict on whether the claim is true, false, or mixed, based on a methodology agreed upon by the fact checking community [16]. Fact checking provides context for users to understand information better and is key to the fight against digital disinformation and misinformation and to the overall trust and credibility of journalism [14]. While fact checking has been around since early 2000s, it came to a broader public consciousness in 2016.

The adoption of the open standard ClaimReview markup[2] by many fact checking organizations has enabled major search engines, including Google and Bing, to provide additional support for fact checking content across their products [4, 12, 17], because the markup makes it easy to correctly identify fact checking articles via structured data; the support from search engines in turn spurred the growth of the fact checking community globally [19]. The data encoded within ClaimReview markup provides three key fields on top of the content of the fact-checking articles, which are: 1) *claim*, the statement that is being fact checked; 2) *claimant*, the person or organization making the claim; and 3) *verdict*, the conclusion on the veracity of the claim according to the fact checker. This alleviates the need to extract those fields from the text, a long standing challenge in information extraction which can now be bypassed thanks to the contributions of fact checkers.

What the structured data fields cannot reliably provide, however, are the documents across the Web that are relevant to the claim(s) which are examined in the fact-checking article. This is not due to the limitation of the markup: the fact checker can indeed provide the URL(s) of the page(s) repeating the claim[3]. However, this is rarely done for several reasons. First, many fact checkers are concerned about the spread of fringe content and thus reluctant to provide the URLs for fear that they might be misused. Second, a popular claim may be discussed by hundreds of different articles and it is impossible for the fact checker to find all such articles. Third, a relevant document may be created long after the fact-checking article was written and it is infeasible for fact checkers to routinely go back to all the fact-checking articles they have written and update the list.

Meanwhile, identifying claim-relevant documents is extremely useful. It provides a means to identify potential misinformation when the stance towards the claim in the claim-relevant document differs from the verdict of a fact-checking article.[4] Note that the claim-relevance discovery problem does not require the literal or precise claim to appear in the document but rather aims to find ones that seem to align in spirit. Thus, our goal is not to identify articles that are being debunked by the fact-checking article, but rather to identify as comprehensive a set of claim-relevant documents as possible and classify their stances toward the given claim.

Figure 1 illustrates a fact-checking article by Snopes examining the claim, “Recently discovered satellite photos from Google

[1] In this paper, we focus on *post hoc* fact checking, which is different from *ante hoc* fact checking that a publisher does to ensure their own stories are factually accurate.

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

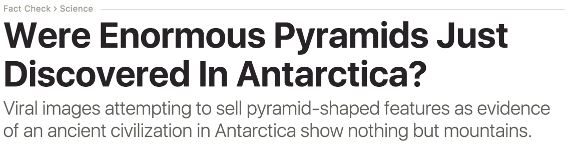
[3] https://developers.google.com/search/docs/data-types/factcheck#creativework

[4] Here, we assume the fact checking organizations that produce those fact-checking articles are high quality and reputable publishers. Evaluating the credibility of the fact checking organizations is beyond the scope of this paper.

WWW'18 Companion, April 23–27, 2018, Lyon, France

ACM ISBN 978-1-4503-5640-4/18/04.
https://doi.org/10.1145/3184558.3188723

Earth show that an ancient civilization built pyramids in Antarctica 100 million years ago,"[5] with the title of the article being "Were Enormous Pyramids Just Discovered In Antarctica?" and the verdict being "false." A quick Google search reveals that at least ten documents are relevant to this claim and express positive stance. Figure 2 illustrates one such example[6] amid many. And there are a similar number of documents that, like Snopes, contradict[7] the claim or discuss it without passing judgment.[8]

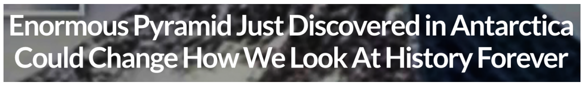


Fact Check › Science

Were Enormous Pyramids Just Discovered In Antarctica?

Viral images attempting to sell pyramid-shaped features as evidence of an ancient civilization in Antarctica show nothing but mountains.

Figure 1: A fact-checking article with a claim.



Enormous Pyramid Just Discovered in Antarctica Could Change How We Look At History Forever

Figure 2: Example claim-relevant document w.r.t. the claim in Figure 1.

These examples illustrate several challenges for the claim-relevance discovery problem, which is formalized in Section 3. First, it is not clear how to find candidate documents, especially when the claim can be expressed in multiple ways. We propose methods to craft well-designed queries that capture the claim and leverage a search engine to identify candidates. Second, while modern search engines do a good job of returning lexically related results, not all of the results actually address the claim statement because search engines do not have semantic understanding of it. Indeed, our experiments show that most of the top-100 returned results tend to be irrelevant to the claim, despite our best efforts to formulate the right queries. To solve this challenge, we design a classification model to predict, given a claim and a candidate document, whether the document is relevant to the claim. Finally, the most difficult challenge is how to tell apart which stance each document, among the relevant ones, has towards the claim. This is similar to the so-called "stance detection" problem from the Fake News Challenge (FNC) [26] and explored in [21].

To tackle these three challenges, we propose a three-stage system consisting of: 1) a candidate generation phase that retrieves candidate documents based on signals generated from the fact-checking article and its claim (Section 4); 2) a relevance classification phase that filters away documents that are unlikely to be relevant to the claim (Section 5); and 3) a stance classification phase that predicts the stance a relevant document has towards the claim (Section 6).

Finally, in Section 7, we experimentally demonstrate that: 1) we are able to automatically find a comprehensive candidate dataset of related documents with 80% recall compared to intensive manual effort; 2) we build a relevance classifier that achieves 81.7% accuracy, surpassing the winning method from FNC (trained on our data) by almost 5%; and 3) we build a stance classifier with 91.6% accuracy, outperforming the winning method from FNC by more than 6%.

2 RELATED WORK

Research related to fact checking fall into a few general categories. First, a rich body of works on information veracity in the context of information extraction, especially in the big data era where multiple sources can provide conflicting information, starting with [32]. The key research topics include determining what is the correct information and which sources are more credible. The information being studied in these works, however, are simple facts represented by knowledge base triples and very different from the general claims with complex semantics we study in this paper. Second, semi-automated fact checking is an emerging research field aimed at producting tools to assist journalists in performing fact checking more efficiently [11, 15, 30, 31]. Some of the topics include identifying true but misleading statements, using query perturbation techniques against a given knowledge source, and predicting which statements are worthy of fact checking based on features generated from just the statements themselves. Our work is complementary to this thread. Third, rumor detection on microblogs has also received a lot of attention [27], where the focus has been leveraging the social graph structure to detect the emergence of misinformation in social networks like Twitter. Many of those techniques can be considered as lead generation techniques for fact checkers. Finally, [28] proposed an algorithm to make connections between news articles to help users form a better understanding of the stories, which is a general goal we share. Their specific problem, however, is different in that the connections they try to make between the news articles do not have to form a supporting or contradicting relationship and can be tangential.

Text similarity techniques are widely used in text classification, sentiment analysis, document clustering, etc. For our task specifically, text similarity is crucial in determining whether a document we found is actually relevant to the claim in the fact-checking article. Traditional word-based models (bag-of-words or bag-of-n-grams) suffer from data sparsity and high dimensionality. They usually fail to capture the semantics of texts, hence two pieces of texts with few overlapping words will have low similarity even if some of the words share similar semantic meanings. Recent works on text embeddings [1, 18, 22, 23] prove to be highly effective in identifying the similarity between short texts. In [22], the authors computed continuous vector representations of words from very large datasets and show that the neural network based language models significantly outperform N-gram models. In [18], the authors propose Paragraph Vector that learns continuous distributed vector representations for pieces of texts, which can capture both the word ordering and the semantics in texts. We adopt some of those techniques in our relevance classification phase.

The problem of *stance detection*, or *stance classification*, has multiple definitions in the NLP literature including targeted entity sentiment in [24], agreement with a controversial topic in [3] and agreement with a rumor in the Fake News Challenge [26]. The last of these is closest to our problem, and is similar to the textual

[5] https://www.snopes.com/enormous-pyramids-just-discovered-in-antarctica/

[6] https://goo.gl/qQbM84

[7] https://goo.gl/K3wy1u

[8] https://goo.gl/SuhkQD

entailment problem in natural language inference. For this task, [5] collected a large dataset (SNLI) [6] with 570k human-written English sentence pairs. Each sentence pair includes a premise and a hypothesis with labels entailment, contradiction, and neutral. Many models have also been developed to advance the state-of-the-art results on natural language inference and text understanding, including feature-based models [5], sentence encoding-based models [7], and general neural network models [8, 25, 29]. Among those, [25] built an attention model to decompose the problem into sub-problems that can be solved separately, and achieved 86.8% accuracy on the SNLI dataset with almost an order of magnitude fewer parameters than previous work. More recently, [8] built an enhanced LSTM model and achieves 88.6% accuracy on the SNLI dataset. [13] introduces an Densely Interactive Inference Network (DIIN) to achieve high-level text understanding by hierarchically extracting semantic features from interaction space, and achieves state-of-the-art results on the SNLI dataset with 88.9% accuracy.

3 DEFINITIONS AND OVERALL SYSTEM

Definition 3.1. A **fact-checking article** is defined as an article that examines a single factual assertion, which we refer to as the **claim**, and produces a **verdict**, which judges the veracity of the claim according to the research done by the authors.

Fact-checking articles are typically written by professional journalists and fact checkers. In this paper, we are only interested in those that are annotated with the ClaimReview markup[9].

Definition 3.2. Given a fact-checking article, a **related document** is a document that bears some topical or lexical similarity to the fact-checking article.

We rely on Google Search to fetch related documents based on queries that we craft from the fact-checking articles. All relevant documents (Definition 3.3) are selected from the set of related documents.

Definition 3.3. Given a fact-checking article with claim c, a **claim-relevant document** is a related document that addresses c.

Not all documents related to a fact-checking article are relevant. The difference between the two is best illustrated through an example. Consider a fact-checking article with the claim "Recently discovered satellite photos from Google Earth show that an ancient civilization built pyramids in Antarctica 100 million years ago."[10] and a related document discussing "Pyramid-like structures spotted in the Egyptian desert."[11] Although these two articles are similar, the related document is *not* relevant to the specific claim since it addresses pyramids in the desert rather than those found in Antarctica. A relevant document can address the claim in many different ways: some merely report the claim or discuss it without passing judgment ("It has been claimed by XYZ that ..."). Most, however, aim to either support or contradict the claim.

Definition 3.4. Given a claim c, a **contradicting document** is defined as a relevant document that contradicts c, and a **supporting document** is defined as a relevant document that supports c.

Both contradicting and supporting are terms that describe the stance that a document has towards the *claim*, and not (necessarily) with respect to the fact check *verdict*. Importantly, we note that the literal or precise claim is not required to appear in a claim-supporting document; rather, the document aligns with the spirit of the claim. Among the relevant documents, contradicting documents are particularly interesting: when the fact check considers the claim to be false, the contradicting documents represent documents that refute the claim, which could potentially provide more context or evidence for users to understand the whole story better.

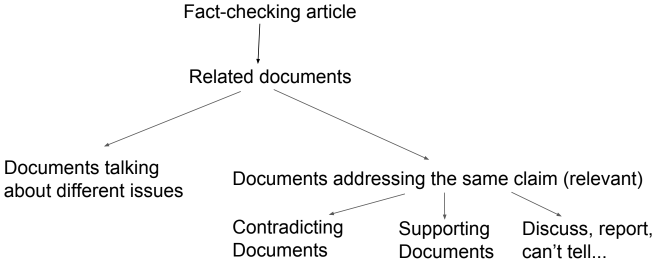

Fact-checking article
Related documents
Documents talking about different issues
Documents addressing the same claim (relevant)
Contradicting Documents
Supporting Documents
Discuss, report, can't tell...

Figure 3: System Overview: finding claim-relevant documents to fact-checking articles

3.1 Overall System

Our overall system works as follows. We start from a set of fact-checking articles. For each article, we craft a set of queries and use a search engine to find a set of related articles. We then build a binary classifier to predict whether a related document is a relevant document. Finally, among all the relevant documents, we classify the stance of each relevant document w.r.t. the fact-checking article and its claim. An overview of the system is shown in Figure. 3. The next three sections dive into details about each of those components.

4 CANDIDATE GENERATION

The goal of Candidate Generation is to find as comprehensive a set of *related* documents as possible so that the full set of *relevant* documents can be discovered within this candidate set. We adopt two main mechanisms for discovering related documents: navigation and search.

4.1 Candidate Generation via Navigation

Intuitively, we start by including outgoing links and source articles cited in the fact-checking article as candidate documents, assuming they are related by nature, which they almost always are. However, most of those are not *relevant*. For example, for the fact-checking article on "Pyramids discovered in Antarctica,"[10], while some outgoing links are relevant, including a supporting document[12] and a contradicting document[13], most of them point to related but irrelevant documents, such as documents on "antarctic climate evolution" and "continental drift."[14]

[9] http://schema.org/ClaimReview
[10] https://www.snopes.com/enormous-pyramids-just-discovered-in-antarctica/
[11] https://goo.gl/etTKik
[12] https://goo.gl/qQbM84
[13] https://goo.gl/B6SHDJ
[14] https://goo.gl/zA9tN4, https://goo.gl/bGMU6P

4.2 Candidate Generation via Search

Recognizing the limited coverage through navigation, we turn to search. The key challenge in generating candidates via search is to formulate the right set of queries: we want the queries to be as specific as possible without losing potentially relevant documents. The three main categories of queries we adopt include:

1. *Texts from the title of the fact-checking article and the claim of the ClaimReview markup*. This category is an obvious choice: the article title typically summarizes the fact check and the claim text summarizes the claim being fact checked.
2. *Title and claim text transformed with entity annotations*. In this category, we perform entity resolution [20] on the text and transform the text into an alternative form that contains only the inferred entities. For example, given the claim "A video documents that the shootings at Sandy Hook Elementary School were a staged hoax," we will generate the query "video documents shootings Sandy Hook Elementary School hoax," which is a simple concatenation of the five entities (video, documents, shooting, Sandy Hook Elementary School, hoax) discovered in the text. The main motivation behind this is to extract important information only from long texts.
3. *Click graph queries [9]* associated with the fact-checking article, which are popular search queries that led to the click on the fact-checking article. We collect up to 50 most popular click graph queries, which turns out to be a very useful query source as we show empirically in Section 7.

We issue each query to Google search and collect top-100 results. Duplicates across different query result sets of the same fact-checking article are removed. Combining both navigation and search, for each fact-checking article, we generate about $2,400$ related candidate documents for further processing. As we detail in Section 7, we are able to achieve 80% recall over the golden data produced by skilled workers performing open-ended research.

5 RELEVANCE CLASSIFICATION

The focus of the Candidate Generation component is recall, namely, identifying as many related documents as possible among which it is hoped are all the relevant documents. It is the goal of the Relevance Classification component to prune away irrelevant documents. Based on a labelled corpus of $8,000$ (fact-checking article, related document) pairs as described in Section 7, we build a classification model $M(f, d) \rightarrow \{relevant, irrelevant\}$, where f, d are the fact-checking article and related document, respectively, to predict relevance.

5.1 Evidence

The fundamental task is to find out how similar the related document is to the claim. In designing features for the classification model, we collect evidence from both the fact-checking article and the related document and build features based on the combination of those evidence. We find that sentence-level similarities often provide strong evidence to help us determine document-level relevance.

Evidence from the fact-checking article:

- Claim (c): the text of claim as provided by the fact-checker is a very good summary of what we are looking for in a relevant document. (Unfortunately, no such structured data exists for the related documents.)
- Article title (f_t)
- Article headline (f_h): some fact checkers (e.g., Snopes) provide a more detailed headline summarizing the fact check
- Selected sentences (f_{s_i}): we further select sentences in the fact-checking article that are similar to the claim text, i.e., $sim(c, f_{s_i}) > \theta$
- Entities annotated on the claim text with associated confidence score a la [20]

Note for f_{s_i}, we specifically do not include all sentences from the fact-checking article because many sentences are not about the claim itself.

Evidence from the related document:

- Document title (d_t)
- Document headline (d_h), if there is one
- Sentences (d_{s_i}): all sentences from the related document
- Paragraphs (d_{p_i}): all paragraphs from the related document
- Entities annotated on the entire content with a confidence score

Note that we use all sentences from the related document because, unlike in the case of fact-checking article, we don't have a good selection mechanism to determine which sentences are more useful. We also use paragraphs, which are groups of sentences, because we believe a paragraph can sometimes provide a more comprehensive set of key information needed to be matched to the claim in the fact-checking article.

5.2 Features

For each pair of fact-checking article and related document, we extract the following features:

Entity similarity: Annotated entities with confidence scores are collected from the claim text of the fact-checking article and the entire content of the related document. Let $\{e_1, e_2, \ldots, e_K\}$ be the union of the entities extracted from both sides, we then represent the entities from each side as a vector of confidence scores aligned with the entity vector, $\{c_1, c_2, \ldots, c_K\}$ (c_i is 0 if the entity is not found on this side).

The entity similarity feature is computed as the cosine similarity between the two entity confidence score vectors.

Core text similarity: $sim(c, d_t)$ and $sim(c, d_h)$

Claim-to-sentence similarity: $max_i(sim(c, d_{s_i}))$

Claim-to-paragraph similarity: $max_i(sim(c, d_{p_i}))$

Sentence similarity: $max_{i,j}(sim(f_i, d_{s_j}))$, where f_i iterates over f_t, f_h, f_{s_i}.

Content similarity: $sim(c, d)$ and $sim(f, d)$, i.e., the similarities between the entire content of the related document to either the claim or the entire content of the fact-checking article

For each pair of texts, $sim(t_1, t_2)$ is computed as the cosine similarity between the text embeddings. For each piece of text, the text embedding is computed by taking a weighted sum over the word embeddings and phrase embeddings (by connecting two tokens), where the weight is computed based on the inverted word/phrase frequency (i.e., less-frequent words/phrases get larger weights). The word/phrase embeddings are pre-trained vectors on the Google News dataset [22, 23].

Publication order: We also extract the publish date from both the fact-checking article and the related document and compute the absolute difference (in days) between the two dates. This captures that fact-checking articles tend to appear around the same time when a claim is published to counter misinformation.

5.3 Model

We build a gradient boosted decision tree model [10], which combines all the features described above and predicts whether a related document is relevant to the fact-checking article. We select the hyper parameters (max depth, learning rate, number of estimators etc.) for the GBDT model based on 10-fold cross validation.

6 STANCE CLASSIFICATION

In this phase, we build a model, $M(f, d) \rightarrow \{contradict, support\}$, where f, d are the fact-checking article and relevant document, respectively, to classify d into the following two categories, given the fact-checking article and its claim:

- **Contradicting document**: a relevant document that contradicts the claim.
- **Supporting document**: a relevant document that supports the claim;

While we acknowledge the existence of a "neither / discuss" category[15], the number of such relevant documents is small within the fact checking domain and thus it is not the main focus of this paper. Based on a crowd-sourced labeling experiment, we found that roughly 80% of the relevant documents fall into contradict or support (see Section 7); this is quite different from the dataset used in [26], where 66% of the relevant documents fall into the "discuss" category. Upon inspection of non-contradict/support documents, we found most to contain user-generated content, such as those from forums, social network posts and video[16], since these documents include opposing statements by different users and are therefore often too confusing or subjective, without a single narrative, to classify correctly by human raters.

6.1 Overall Intuition

Similarity is not always a good proxy for stance detection: two texts may have much lexical overlap yet simply adding the word "not" in the right place completely changes the stance status. Therefore, our main intuition in building the stance classification model is to determine whether a relevant document contains statements that are contradictory to the claim. Technically, for each relevant document, we would like to identify key contradicting patterns in contexts that are similar to the claim. This intuition is rooted in our observation that documents spreading misinformation often attempt to report a false claim in a non-rigorous manner unlike the usual evidence-based discourse that journalists use in backing up a report.

In designing features for contradiction, we noticed that for many fact checkers (e.g., Snopes), when the verdict is "false" (or "mostly false") the headline of the fact-checking article is often a perfect example of a statement that contradicts the claim. For example, the article[17] that fact checks the claim "Country star Willie Nelson has died" has the headline, "Country music legend Willie Nelson is not dead; he's just the target of a recirculated celebrity death hoax," which directly contradicts the claim.

6.2 Vocabulary for Contradiction

Based on the above observation, we decided to construct a relatively small lexicon that can indicate contradicting discourse. We collected around $3.1k$ (claim, contradicting statement) pairs from fact-checking articles where the headline can be easily extracted. We aggregate uni-grams and bi-grams from the contradicting statements and build a 900-dim vocabulary using the grams with highest frequency. Stopwords were removed except those expressing negation like *no* and *not*. The most-frequent uni-grams with contradicting intent include *fake*, *purportedly*, *hoax*, and *rumor* and bi-grams include *made up*, *fact check*, *not true*, and *no evidence*. Note there are other uni-grams and bi-grams appearing as frequently with no contradicting intent such as *Web*, *story* and *reported*, which are likely to be given less weight in the classifier by learning from examples from both supporting and contradicting documents.

6.3 Classifier

Given a relevant document and a claim, we first collect *key textual evidence* from the relevant document for subsequent feature generation. As described in Section 5, textual evidence from the relevant document include title (d_t), headline (d_h) and important sentences (d_{s_i}) and we use all of those. We then prune away any text whose similarity with the claim is less than a predefined threshold[18]. We call those that remain *key texts*.

For each key text, we further construct *key components* by concatenating the sentence with its surrounding text (one sentence each before and after the key text). This is mainly because in a lot of cases a contradicting document will first state the claim and then followed by simple contradicting statements, e.g., "Does ... really happen? No, it turns out to be false."

We then extract uni-grams and bi-grams from the union of the key components, based on the contradiction vocabulary constructed in Section 6.2. Both the uni-grams from the vocabulary and those extracted from the key components are stemmed for better matching. The final feature is a vector of n-gram weights over the contradiction vocabulary where each weight is the frequency of a particular vocabulary word that appeared in the key components.

Finally, we build a gradient boosted decision tree model based on those n-gram features to predict if a relevant document is a supporting document or a contradicting document. Similar to the relevance classifier, we select hyper-parameters (max depth, learning rate, number of estimators, etc.) for the GBDT model based on 10-fold cross validation.

[15] An example would be an article about how a politician made a specific claim without passing judgment on the claim itself (though many sources tend to call it out if the claim is obviously false).

[16] Our techniques cannot yet analyze video content. We do, however, recognize that digital misinformation is spread through online videos and intend to tackle this challenge as future work.

[17] https://www.snopes.com/inboxer/hoaxes/willienelson.asp

[18] The threshold we use is 0.8, which is chosen to be as high as possible while ensuring enough evidence.

6.4 Examples

Here we highlight some examples to demonstrate the importance of identifying key texts in the relevant document, using a fact-checking article which debunks the claim that "Shaving makes your hair grow back in thicker, faster, and fuller"[19]. Examples of contradicting documents where the contradicting information (bolded) is explicitly expressed in titles or headlines: (1) "The **Debunker**: Does Hair Grow in Thicker After You Shave?"[20]; (2) "Shaving myths, **busted**"[21]. Examples where the title or headline ("Does shaving make hair grow back thicker?") does not convey much information and key sentences and key components from main texts are necessary for contradiction detection: "Given the facts above, it can safely be said that shaving actually does **NOT** make hair thicker and grow faster,"[22] and "If you shave your legs, underarms or any other part of your body, it may appear that your hair grows back thicker and coarser. **But it doesn't.**".[23]

While our current model works reasonably well, as our experiments in Section 7 show, we do note that further improvements will be necessary and challenging. Specifically, there are more complex patterns like "*How on earth...?*", "*rules out..*", "*...? Come on*", "*..., or was it?*", etc., and patterns that only make sense in specific context such as *reunited with* vs. *not friends anymore*, *not acting maliciously* vs *had intent to do harm*, etc. Even more long tail patterns include particular usages of quotes and question marks, sarcastic languages, etc. We present some of those hard cases Section 7.3.3.

We believe to build a model that is capable of capturing the more complex and long tail contradicting semantics, we will need to collect a dataset with comprehensive contradicting language patterns and this is part of our future work.

7 EXPERIMENTS

7.1 Datasets

We used the following datasets for evaluation:

Unlabeled Corpus: This corpus is programmatically created based on the ClaimReview markup. We crawled the Web for all the documents containing exactly one ClaimReview in JSON or Microdata format. Being an open standard, it is to be expected that there are misuses of the markup. We therefore perform the following filtering to ensure we have a corpus of valid and high quality fact checks. First, we ignore any invalid markup where the ClaimReview fails to parse or is missing any of the three key fields (claimant, claim, verdict). Second, we ignore any syndicated or plagiarized fact check where the article containing the markup points to a fact-checking article from a domain that is different. Third, we perform deduplication such that if there are two identical fact checks (same fact checker, claim, and verdict), only one is retained. This can happen when the fact checker publishes a periodic roundup article of prior fact checks. Last, we leverage the IFCN list of signatories[24] and retain fact checks from only those considered by the fact checking community as highly reputable. After all the filtering, there were 14, 731 fact-checking articles in this corpus.

Relevance-labeled Corpus: Using the candidate generation algorithm we developed in Section 4, we found roughly 2, 400 related candidate documents per fact-checking article from the Unlabeled Corpus for a total of 33.5M (fact-checking article, candidate document) pairs. We then sampled from these pairs to determine relevance labels using the Google crowd-sourcing platform where the workers are ordinary Web users recruited from across the English-speaking world. Given the fact-checking article and its associated claim, each worker is asked to answer the question "*Does the candidate document address the claim?*" Each pair is rated by 3 workers and each worker can rate up to 6 pairs. A pair is considered as positive (i.e., the candidate document is relevant to the claim) if a majority of the workers answer "yes" to the question; otherwise, it is considered as negative. We then subsampled to achieve a balance of positive and negative examples. In total, 8, 000 relevance-labeled pairs are collected for training and evaluating the relevance classification described in Section 5.

Stance-labeled Corpus: We further randomly sampled 1, 200 from the positive pairs of this corpus for crowd-sourced stance labeling using the same platform via the question, "*Does the candidate document (1) support the claim (2) contradict the claim (3) neither (4) can't tell?*." Again, each pair is rated by 3 workers and each worker can rate up to 6 pairs. Each category must be agreed to by at least two workers before it is assigned as the label for the pair, otherwise, the pair is labelled as unknown. For about 12% of pairs, the workers did not achieve agreement and those pairs are removed. This corpus is used for training and evaluating stance classification described in Section 6.

Manual Corpus: The labeled corpora above are useful for evaluating the performance of the classifiers but cannot be used to measure the performance of the candidate generation. For that, we randomly sampled 450 fact-checking articles from the Unlabeled Corpus and trained more skilled (and expensive) crowd-source workers to perform open-ended research tasks with the sampled articles for the goal of discovering as many supporting documents as they could find using whatever means they deemed as useful. Most of the researchers used a search engine with creatively crafted queries and then followed the links on the search results. Each relevant (fact-check article, supporting document) pair was further examined by two human researchers for agreement before acceptance. In total we obtain around 4,000 fact-checking article to supporting-document pairs.

7.2 Results

7.2.1 Recall of Candidate Generation. We use the Manual Corpus as described above to evaluate the coverage of our candidate generation algorithm. A simple baseline method is to use (the top-100) search query results based on claim text and/or title, as well as outgoing links and cited-sources from the fact-check article. This yielded recall less than 20%. Table 1 shows how various proposed query generation methods compare to the baseline, where each line is additive of all previous lines. Overall we achieved recall 80%. Note here the upper bound in practice is about 90% since 10% of the

[19] https://www.snopes.com/oldwives/hairgrow.asp
[20] https://goo.gl/rB3zS4
[21] https://goo.gl/zpa4po
[22] https://goo.gl/KFCJZH
[23] https://goo.gl/ExRfze
[24] https://www.poynter.org/international-fact-checking-network-fact-checkers-code-principles

Method	Recall
claim-text, title, outgoing-links and cited-sources	<20%
+ top-10 click-graph queries	53.8%
+ top-50 click-graph queries	74.8%
+ annotated entities from claim-text/title	80.0%

Table 1: Candidate Generation: recall analysis

URLs have pages that are no longer accessible, which is common for online documents spreading misinformation.

7.2.2 Performance of Relevance Classification. Based on the Relevance-labeled Corpus, our proposed relevance classifier achieved 81.7% ± 1.8% accuracy averaged over 10-fold random train/test split on the entire dataset and handily beat the majority class label baseline, which has 50% accuracy due to the balance in the positive/negative labels. We also note that simply taking all related documents generated by the candidate generation phase achieves much lower accuracy/precision because the majority of related documents are not relevant.

Table 2 shows the performance comparison between our classifier and the various baseline classifiers. In particular, we compare with the winning model of the Fake News Challenge [2]. Since the code made available[25] does not allow the CNN part to be re-trained, we trained only the GBDT part on our dataset and achieved 77.2% accuracy. It is worth noting that if we apply the FNC winner model trained on the FNC dataset, the accuracy drops to 51.3%. This highlights the significant differences in the characteristics between our dataset and the FNC dataset. Regarding running time, our feature extraction operates independently on each fact-checking article and related-document pair, which is scalable for both training and testing on large datasets. The FNC model uses a joint model on textual features from all training and test data and might therefore not be as scalable. In addition, their model needs to be re-trained for prediction on new test examples, while our model does not need to be re-trained for prediction.

We examined mis-classification cases and found they were generally due to the following:

- Incorrect extractions resulting in poor titles, incorrect publication dates, etc.
- UGC pages such as online video pages, social network posts and forum pages.

The impact of these resulted in either wrong text being used as feature or not enough text to enable good prediction.

7.2.3 Performance of Stance Classification. To ensure we have enough data for training the model, we augmented the Stance-labeled Corpus with $4k$ (fact-check article, supporting document) pairs from the Manual Corpus and $3.1k$ (fact-checking article, contradicting document) pairs automatically generated by pairing claims taken from fact-checking articles having "false / mostly false" verdicts with the fact-check articles themselves as relevant documents. Overall, we have a labeled dataset with 8, 422 pairs, among which 57% are supporting documents and 43% are contradicting documents.

Table 3 shows a comparison of our model and a few baselines and we achieve the best accuracy (91.6%) among all the methods. The main comparisons are made against the DIIN model [13], which achieves state-of-the-art accuracy 88.9% on the SNLI dataset [6]. In our dataset we do not have any pairs with neutral relationships while one third of the SNLI dataset are neutral. Thus, for fair comparison, if DIIN predicts neutral, the pair is not counted when computing accuracy. For claim-to-document pairs (original task), DIIN achieves accuracy 53.8%. Since the DIIN model was designed for pairs of short texts, we further cleaned the dataset to best approximate the sentence pairs with known entailment/contradiction relationships: 1) for entailment we picked the sentence with highest similarity to the claim from supporting documents (this could contain errors since the most-similar sentence might not entail the claim); 2) for contradiction we picked the (claim, headline) pairs from the Snopes articles since we know they directly contradict each other. The cleaned dataset has 2, 972 entailment pairs and 3, 137 contradiction pairs. Among all 6, 109 pairs, DIIN predicted 4, 446 as "neutral" (which are wrong), 786 as "entailment" (561 of which are correct), and 877 as "contradiction" (629 of which are correct). The accuracy is 71.6% (if "neutral" predictions are counted the accuracy drops to 19%). We further compared our method with the FNC model trained on our dataset, where the accuracy is 85.0%. If we apply their model trained on the FNC dataset, the accuracy drops to 44.3% (the pairs where the model predicts "unrelated" or "discuss" are ignored, if they are counted the accuracy drops further to 25.1%).

7.3 Case Studies

7.3.1 Coverage of Candidate Generation. Here are the two main reasons for coverage loss:

Bad query construction: In some cases the article title or the claim text does not serve as a good summary of what being fact checked or lose important information entailed in the fact-checking article. In addition, in some cases there are better candidates in the fact-checking article that serve as better queries (e.g., quotes). For example, for the fact-checking article talking about "Steve Jobs deathbed speech"[26], the exact quote "I reached the pinnacle of success in the business world ..." turns out to be a much better query candidate.

Unfocused fact check: The fact-checking article discusses too many issues and there is a discrepancy on the focus between the raters and the summaries we generated.

7.3.2 Relevance Classification. Some relevant document may include the claim as a small fraction of the article content, e.g., this document[27] "6 rock stars people swore were dead hoax" has "Willie Nelson death hoax" as one of the examples (with two paragraphs). Overall the article does not share high similarity with the claim text (and the title is not indicative). This again demonstrates the importance of finding key sentences in the document and using the similarity between claim text and key sentences in the related document as one of the features in the classifier.

In addition, the percentage of relevant documents in the related document set varies greatly depending on the claim and topics

[25] The code is from https://github.com/Cisco-Talos/fnc-1/

[26] https://www.snopes.com/steve-jobs-deathbed-speech/

[27] https://goo.gl/ARj7Nu

Method	Accuracy	Precision	Recall
Majority class label	50%	50%	100%
Entity resolution	68.8%	70.5%	64.9%
Text similarity (over embeddings, claim to title)	68.7%	70.3%	65.3%
Text similarity (over embeddings, claim to article)	67.3%	66.9%	68.4%
Text similarity (over embeddings, article to article)	71.5%	71.3%	72.3%
FNC winner trained on FNC data	51.3%	5.5%	61.4%
FNC winner trained on our data	77.2%	67.9%	84.9%
Our model	**81.7%**	80.9%	83.1%

Table 2: Relevance classification: comparison between the proposed approach with baselines

Method	Accuracy	Precision	Recall
Majority class label	57%	57%	100%
DIIN from claim to document	53.8%	62.4%	40.4%
DIIN from claim to sentence	71.6%	71.4%	69.3%
FNC winner trained on FNC data	44.3%	99.8%	44.3%
FNC winner trained on our data	85.0%	84.9%	89.3%
Our model	**91.6%**	95.9%	86.5%

Table 3: Stance classification: comparison between proposed approach with baselines

in the claim. For certain claims/topics the precision on the search result can be quite low since there are many other results on very related but different issues. For example, for the fact-checking article with the claim "The Internal Revenue Service (IRS) is contacting taxpayers via e-mail to convey important tax return information, complete paperwork for a refund, or request payment on a balance owed"[28], using queries like "IRS scam" will yield many search results involving all kinds of scams related to IRS. The specific issue discussed in this claim is for scams through "email" while many related documents are discussing other channels like "phone."

7.3.3 Stance Classification. We list some tricky cases here, where *background and/or domain knowledge* is needed to understand the whole story. In this fact-checking article[29] where the claim "Andy Murray is the first person ever to win two Olympic tennis gold medals" is debunked, the fact is that "Murray is the first to win two Olympic gold medals in singles tennis, but 19 other men and women have won multiple gold medals in all forms of Olympic tennis." Here is a contradicting document[30] that talks about "Andy Murray... just reminded a reporter that women's tennis is still tennis," and it requires some background knowledge on this event as well as some reasoning in order to know it contradicts the claim.

We also show a case where the contradiction is expressed with sarcasm. The fact-checking article[31] checks the claim, "Placing a raw, cut onion in contact with your foot overnight 'purifies your blood,' removes 'toxins', and heals your body." Here is a contradicting article[32] that debunks the same claim, without explicitly expressing contradiction in the document content. Instead, the author discusses "six more things you will find by putting onions in socks" where all six things are stated in obviously ridiculous and funny ways, to imply that the author does not believe the claim. These documents are extremely hard cases and even the most state-of-the-art NLP techniques are not able to handle them very well.

8 CONCLUSION AND FUTURE WORK

In this paper we build an end-to-end system for automatic relevant document discovery and claim-relevance classification for fact-checking articles. Given a fact-checking article, we generate a comprehensive candidate set of related documents by automatically constructing queries using information from the fact-checking document. This is followed by building a classifier that determines whether a related document is relevant (i.e. it addresses the same claim as the one checked in the fact-checking article) and distinguishes between the stances of the relevant document towards the claim. Finally we build a classifier to determine whether a relevant document expresses a potential stance.

As future work we plan to include non-textual data as features. One source of relevance classification error are documents with insufficient text information. For example, for videos we could leverage the transcript using speech recognition techniques and for social network pages we could utilize the image/graphic information encoded.

REFERENCES

[1] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A Simple but Tough-to-Beat Baseline for Sentence Embeddings. In *ICLR 2017*.

[2] Sean Baird, Doug Sibley, and Yuxi Pan. 2017. *Talos Targets Disinformation with Fake News Challenge Victory.* https://blog.talosintelligence.com/2017/06/talos-fake-news-challenge.html

[3] Roy Bar-Haim, Indrajit Bhattacharya, Francesco Dinuzzo, Amrita Saha, and Noam Slonim. 2017. Stance Classification of Context-Dependent Claims. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 1: Long Papers.* 251–261.

[4] Microsoft Bing. 2017. *Bing adds Fact Check label in SERP to support the ClaimReview markup.* https://blogs.bing.com/Webmaster-Blog/September-2017/Bing-adds-Fact-Check-label-in-SERP-to-support-the-ClaimReview-markup.

[5] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *2015 Conference on Empirical Methods in Natural Language Processing (EMNLP).*

[6] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015.* 632–642. http://aclweb.org/anthology/D/D15/D15-1075.pdf

[28] https://www.snopes.com/irs-scam-season/

[29] https://goo.gl/4TgTkc

[30] https://goo.gl/AQ1qVS

[31] http://www.snopes.com/onion-in-your-sock-cure/

[32] https://goo.gl/nswXR2

[7] Samuel R Bowman, Jon Gauthier, Abhinav Rastogi, Raghav Gupta, Christopher D Manning, and Christopher Potts. 2016. A Fast Unified Model for Parsing and Sentence Understanding. In *ACL 2016*.

[8] Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced LSTM for Natural Language Inference. In *ACL 2017*.

[9] Nick Craswell and Martin Szummer. 2007. Random Walks on the Click Graph. In *SIGIR*.

[10] Jerome H. Friedman. 2001. Greedy function approximation: A gradient boosting machine. In *The Annals of Statistics, Volume 29, Number 5 (2001), 1189-1232*.

[11] FullFact. 2016. *The State of Automated Factchecking*.

[12] Richard Gingras. 2016. *Labeling fact-check articles in Google News*. https://www.blog.google/topics/journalism-news/labeling-fact-check-articles-google-news/.

[13] Yichen Gong, Heng Luo, and Jian Zhang. 2017. Natural Language Inference over Interaction Space. In *arXiv:1709.04348*.

[14] Alan Greenblatt. 2017. *The Future of Fact-Checking: Moving ahead in political accountability journalism*. https://www.americanpressinstitute.org/publications/reports/white-papers/future-of-fact-checking/single-page/.

[15] Naeemul Hassan, Bill Adair, James T. Hamilton, Chengkai Li, Mark Tremayne, Jun Yang, and Cong Yu. 2015. The Quest to Automate Fact-Checking. In *Proceedings of the 2015 Computation+Journalism Symposium*.

[16] IFCN. [n. d.]. *International Fact-Checking Network fact-checkers' code of principles*. https://www.poynter.org/international-fact-checking-network-fact-checkers-code-principles.

[17] Justin Kosslyn and Cong Yu. 2017. *Fact Check now available in Google Search and News around the world*. https://www.blog.google/products/search/fact-check-now-available-google-search-and-news-around-world/.

[18] Quoc Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. In *ICML 2014*.

[19] Michelle Ye Hee Lee. 2017. *Fighting falsehoods around the world: A dispatch on the growing global fact-checking movement*. https://www.washingtonpost.com/news/fact-checker/wp/2017/07/14/fighting-falsehoods-around-the-world-a-dispatch-on-the-global-fact-checking-movement/.

[20] Pablo N. Mendes, Max Jakob, Andres Garcia-Silva, and Christian Bizer. 2011. DBpedia Spotlight: Shedding Light on the Web of Documents. In *Proceedings of the 7th International Conference on Semantic Systems (I-Semantics)*.

[21] Marcelo Mendoza, Barbara Poblete, and Carlos Castillo. 2010. Twitter Under Crisis: Can we trust what we RT?. In *Proceedings of the first workshop on social media analytics*. ACM, 71–79.

[22] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *Proceedings of Workshop at ICLR, 2013*.

[23] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality.. In *NIPS 2013*.

[24] Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiao-Dan Zhu, and Colin Cherry. 2016. SemEval-2016 Task 6: Detecting Stance in Tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016*. 31–41. http://aclweb.org/anthology/S/S16/S16-1003.pdf

[25] Ankur P. Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A Decomposable Attention Model for Natural Language Inference. In *EMNLP*.

[26] Dean Pomerleau and Delip Rao. [n. d.]. *Fake News Challenge*. http://www.fakenewschallenge.org/

[27] Vahed Qazvinian, Emily Rosengren, Dragomir R. Radev, and Qiaozhu Mei. 2011. Rumor has it: Identifying Misinformation in Microblogs. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*. 1589–1599. http://www.aclweb.org/anthology/D11-1147

[28] Dafna Shahaf and Carlos Guestrin. 2010. Connecting the dots between news articles. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 623–632.

[29] Shuohang Wang and Jing Jiang. 2016. Learning Natural Language Inference with LSTM. In *HLT-NAACL*.

[30] You Wu, Pankaj K. Agarwal, Chengkai Li, Jun Yang, and Cong Yu. 2014. Toward Computational Fact-Checking. *PVLDB* 7, 7 (2014), 589–600. http://www.vldb.org/pvldb/vol7/p589-wu.pdf

[31] You Wu, Pankaj K. Agarwal, Chengkai Li, Jun Yang, and Cong Yu. 2017. Computational Fact Checking through Query Perturbations. *ACM Trans. Database Syst.* 42, 1 (2017), 4:1–4:41. https://doi.org/10.1145/2996453

[32] Xiaoxin Yin, Jiawei Han, and Philip S. Yu. 2007. Truth discovery with multiple conflicting information providers on the web. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, California, USA, August 12-15, 2007*. 1048–1052. https://doi.org/10.1145/1281192.1281309