Related Fact Checks

Sreya Guha

Castilleja High School

Abstract. The emergence of "Fake News" and misinformation via online news and social media has spurred an interest in computational tools to combat this phenomenon. In this paper we present a new "Related Fact Checks" service, which can help a reader critically evaluate an article and make a judgment on its veracity by bringing up fact checks that are relevant to the article. We describe the core technical problems that need to be solved in building a "Related Fact Checks" service, and present results from an evaluation of an implementation.

1 Introduction

Fake news, defined by the New York Times as "a made-up story with an intention to deceive", is one of the most serious challenges facing the news industry today. Fake news rose to prominence in the 2016 United States Presidential election, with over 50% of the voting population being exposed to fake news during the election cycle [1]. Instances of fake news have even lead to disastrous consequences such as the shooting at Comet Ping Pong in 2016, which originated from now-debunked news of a sex trafficking ring involving high ranking political officials [2].

The rising influence of fake news poses a clear threat to ethical journalism and the future of democracy. While there are no easy solutions, countering the spread of online fake news lies with the purview of computer science. Currently, many are investigating how computational techniques can be used to ameliorate the spread and popularity of fake news.

While automatically fact checking articles remains the ultimate goal of many researchers, it is now recognized that this is a very complex task that requires not just extremely sophisticated text understanding, but also a level of common sense and world knowledge that computational tools currently do not have. This is further complicated by the fact that the veracity of articles cannot be defined in a binary sense of "fake" or "not fake". Stories and news are often more complex and do not completely fit one of the classifications. For example, in a 2017 interview with the Wall Street Journal, President Trump claimed that Korea used to be a part of China. Numerous sites such as dailybanter.com and QZ.com declared his claim to be false. However, according to the Washington Post, President Trump's claim - while not capturing the entire story - was not completely false. The subtlety that is captured in a fact checking article cannot be replicated by today's programs. We see this in how many fact checkers have a scale of accuracy rather than a simplistic binary distinction. Politifact [3] for example, assigns each claim one of 6 rulings that range from "True" to "Pants on Fire". The Washington Post [4] assigns each claim one of 5 rulings, ranging from "Geppetto Checkmark" (True) to "4 Pinnochios" (False).

Due to the rise of fake news, fact checkers are becoming more prevalent. According to the Duke Reporter's Lab [5], there are currently 116 active fact checking websites in the world. These include national ones such as PolitiFact and Snopes as well as local ones. In 2015, Schema.org [6], working with the Duke Reporters Lab, released vocabulary, including a new type, $ClaimReview$ for adding Semantic markup to fact checks, so as to make them more portable and accessible. As of June 2017, there are more than 7,000 fact checks with ClaimReview markup.

Fact checks have started playing an important role in the news ecosystem. However, there is an piece of functionality, that would substantially improve the utility of fact checks, that is still missing. Fact checking websites do not link to articles circulating incorrect claims just as these articles do not link to the pages that fact check them. This missing link reduces the potential impact of fact checking sites. Though fact check websites are popular and accessed by many, fake news spreads quickly through social media. Without any explicit link between the fake news article itself and the fact checks, the fake news gains traction.

In this paper we propose a new "Related Fact Checks" service. Consider the following scenario: a user is reading an article which makes claims that they would like to verify. Currently, they would have to go to Bing, Google or another search engine and do a set of searches constructed out of the terms/entities in the article, hoping to find a relevant fact check. Imagine instead, a service that would retrieve relevant fact checks, if any. This service would not decide if the claims were true or false. Rather, it would provide a short list of relevant fact checks available and let the reader make an informed decision. This could take the form of a website or a browser extension. This service would be the link between the fake news and fact checks. In this paper, we describe the core technical problem such a service needs to solve, propose a set of techniques for solving this problem and present initial results from an implementation of these techniques.

Though a "Related Fact Checks" service needs to solve many problems related to infrastructure and user interface, the central problem of the service is finding the most relevant fact checks, which is the main topic of this paper. Though this problem may appear to simply be a search problem where the documents being searched are the fact checks and the query being issued is the article the user is reading (regarding whose claims they want fact checks on), the relevance of a fact check to an article is more complicated than a traditional Web search. The entire document or page the user is looking at is the query. In addition, most articles do not have a relevant fact check associated with them and the user should not be given fact checks that are below a certain threshold of relevance, even if they are the most relevant. Therefore, the core problem is a hybrid ranking and classification problem.

The question of the relevance of a fact check to a given article is also more nuanced because of the following phenomenon: there are long running themes in fake news (e.g., anti-vaccine, anti-climate, etc). Even if there is no fact check that deals with the precise claim made by an article, providing the user with fact checks for claims that are along the same theme can help the user more critically interpret the article they are reading.

Outline of paper We first review background literature. Then, we describe the methodology used to create the dataset which includes fact checks, stories covering the claims reviewed by these fact checks and a broad set of articles mentioning some of the entities (such as specific people, places, etc.) mentioned in these fact checks. We then describe a progression of techniques for solving the problem of identifying the relevant fact checks, if any, for a given article and analyze the accuracy of these techniques.

2 Background

The topic of computational treatments of fake news has received a substantial amount of interest in the research community recently. Conroy, Rubin, et. al. [7] provides a survey of the landscape of veracity (or deception) assessment methods, their major classes and goals, all with the aim of proposing a hybrid approach to system design. As they describe, there are two major categories of methods: (1) Linguistic Approaches in which the content of deceptive messages is extracted and analyzed to associate language patterns with deception; (2) Network Approaches in which network information, such as message metadata or structured knowledge network queries can be harnessed to provide aggregate deception measures. Chiu, Gokcen et. al. [8] examine different ways of classifying fake and real articles based on Support Vector Machines. The authors make use of Topic Modeling for classification.

There are a number of 'Fake News Challenges' that have started over the last year, including fakenewschallenge.org [9] and a challenge by Kaggle [10]. Our approach, in contrast, does not attempt to tell the user whether an article is true, or even to what degree it is true. Instead we aim to supplement the article the user is reading with fact checks which might enable the user to more critically interpret the article.

Much of the work on the spread of fake news has focused on its dissemination through social media. Jin and Dougherty [11] apply epidemiological models to information diffusion on Twitter. This paper is the first to employ the SIEZ model to Twitter data and shows the success of this method in capturing the spread of information on Twitter. Tacchini and Ballarin [12] shows that Facebook posts can be determined to be hoaxes or real based on the number of likes. The authors use two classification techniques; one is based on logistic regression while the other on a novel adaptation of boolean crowd sourcing algorithms. Gupta, Lamba, et. al. [13] show how a small number of users were responsible for a large number of retweets of fake images of Hurricane Sandy. Gupta, et. al. [14] used regression analysis to identify the important features which predict credibility. The authors used machine learning to create an algorithm which ranked tweets based on credibility sources.

3 Methodology

We now describe our methodology. We first describe how we built a corpus of fact checks and corpus of articles to test our different techniques. We then discuss the objectives of the "Related Fact Checks" service, which helps us decide whether a fact check is relevant or not to a given article. We then discuss the algorithmic techniques that we used to build a prototype of this service.

3.1 Collecting Fact Checks

According to the Duke Reporter's Lab, there are about 116 sources providing fact checks today [5]. Fact check articles usually contain a lot of text. In order to determine the relevance of a fact check article, we need to identify the primary claim that is being investigated. For this, we rely on the Schema.org ClaimReview, which gives us the claim that is being reviewed. A majority of the sites that provide fact checks now carry this markup on their pages. We crawled the pages on these websites and extracted the ClaimReview markup. A number of the fact checks are duplicated on many pages on the originating site. After removing duplicates, we built a corpus of 5350 fact checks. For each fact check, in addition to the URL, the markup gave us the title of the fact check, the specific claim that was reviewed, the date of the fact, and the rating (or how the claim was judged by the article).

In addition to the corpus of fact checks, we need a corpus of pages for which a user might want to pull up relevant fact checks. We need this both for training/tuning our algorithms and for evaluating them. Constructing such a corpus is complicated by the fact that the vast majority of pages cover topics that are completely unrelated to topics touched on by fact checks. In order to be useful, both for tuning the algorithms and for accurately evaluating the utility of the service, we need a set of pages that deal with subjects that are related to the fact checks but in which the story may or may not have relevant fact checks. For example, consider the page for religion in Sweden in the 17th century on the Swedish government site. Though this page deals with many of the subjects that are the central themes of a flurry of recent fake news (such as Sweden, different religions and the evolution of religious practices in Sweden), the fact checks investigating this recent flurry of fake news are not relevant to the content of this page. In contrast, these facts might be very relevant and useful to provide to a reader who is looking at a page on CNN or BBC on the current crime rate in Sweden. In another example, consider the results for a Google search on "Sweden refugee policy". Some of the results are government policy documents that don't really make any "claim". They are simply statements about the Swedish government's policies. Others are related to various claims made by politicians and media about the impact of refugees on terror and crime in Sweden and are very related to fact checks from many organizations. In contrast, the pages that show the results for the query "Swedish mathematicians" are completely unrelated to the fact checks in our corpus, even those that mention Sweden. Our goal is to collect a corpus of articles that fall in the former categories and avoid those in the last one.

We bootstrap the process of collecting our corpus as follows. The markup for each fact check gives us an entity of type "ClaimReview", which has the property "claimReviewed" (see table 1 for some sample values of the claimReviewed fields). We issue these strings as queries to Google, collect the first 20 results from each query and collate these result by the site. We review a random sample of pages from each of the sites with most pages and manually identify those sites that repeatedly touched on stories that had been fact checked. These are the sites that tended to carry articles that met the criterion laid out earlier. Examples of such sites include New York Times, CNN and Breitbart but exclude the website of the National Oceanic and Atmospheric Administration (which does appear in these results). We create a Google custom search from these sites. We

construct shorter queries from the claimReview strings (by removing stop words, extracting named entities, etc.) and issue them to this Google Custom Search Engine. We collect 5,000 result pages that we were able to crawl. This forms the corpus for our tuning and testing.

A damaged nuclear reactor at Fukushima Daiichi is about to fall into the ocean.
Australia is the first country to begin microchipping its citizens.
There are “no-go zones” in Sweden where the police can’t enter.
A federal judge ruling in a defamation suit declared that CNN was “fake news.” of concentration camps in the U.S.

Table 1: Sample of some of the claims reviewed in the fact checks

3.2 Relevance

Before describing the techniques used to determine whether a fact check is relevant to a particular story, we discuss the question of how we might define relevance in the context of retrieving fact checks for a given page.

If we look at the strict definition of relevance, we should look specifically at the claims made by the page and present the user with only the fact checks that analyze these particular claims. However, fake news is easily generated and fact check sites cannot verify every single dubious claim. Instead, they must choose a select few, leaving many fake stories without fact checks. When we look closer at fake news articles, we find that there are certain long running themes or story lines, with clusters of stories around each theme. Each story within a specific theme may have a different claim but will still relate to the theme.

For example, one prominent theme amongst fake news articles is that vaccines are harmful. There are many specific stories that fall into the genre of anti-vaccine stories, including: ’CDC raided by FBI to seize data on vaccines’, ’If a vial of vaccine is broken, the building must be evacuated’, ’HPV vaccine causes death of 32 year old’, ... There appear to be at least hundreds of stories related to vaccines, if not more.

Another recurring theme in fake news relates to climate change. There are wide range of stories related to this theme, ranging from those about the climate, such as ’Carbon dioxide is not a primary contributor to the global warming that we see’ to those that are related to the political side of climate change, such as ’California legislators have made it illegal for anyone to deny climate change, under threat of jail time’.

Fact checking is laborious and expensive and it is not possible to get all of these articles fact checked. Further, some of these stories (such as the one about a 32 year old woman dying from the HPV vaccine) are so vague and incomplete that it would be virtually impossible to show that this was not true.

When there is a fact check that is very specific to the claim being made in the article, clearly, it is important that we bring up that fact check. However, even the case where

there is no fact check that investigates the specific claim made by the article, if the article falls into a theme where other stories have been fact checked, it would be useful to provide the user with some of those fact checks. These related fact checks will hopefully enable the user to more critically interpret the article that they are looking at. For example, there is a story circulated by the Daily Sheeple with the headlines 'BOMBSHELL: CDC Commits New Vaccine-Autism Crime - Won't Allow Whistleblower to Testify'. As of July 2017, this specific story does not have a fact check. However, stories that are related to this general theme, such as the one which claimed that the FBI raided the CDC to seize data about vaccines, do have fact checks. If the user can be shown that this new story is just the next twist on a long running series of stories, many/most of which have been shown to be not true, the user will be in a better position to judge the veracity of the article currently being read.

We recognize two distinct levels of relevance of an article to a fact check. The first level are those which specifically match the claim. The second level of relevance is where the fact check does not address the specific claim, but addresses claims that are about the same theme and related to the claim made by the article. We are interested in identifying fact checks in both levels of relevance.

3.3 Finding Relevant Factchecks

Given an article, we would like to find relevant fact checks if they exist. This problem bears many similarities with traditional information retrieval and many of the techniques developed in that field can be applied here. However, there are some important distinctions that we need to guide the development of our approach. Specifically,

1. In most of the widely used information retrieval systems (such as web search), we have a large corpus of documents with a query that is comparatively much shorter than the documents. In contrast, in this case, we have a document and a set of fact checks. We have the central claim addressed in the fact check (through the semantic markup on the page), but we don't have (a concise description of) the claim made by the article. In some cases, the title of the article accurately reflects the main claim, but in many others the title is either generic or 'click bait', i.e., optimized for getting users to click on links mentioning the article.
2. In typical search, the goal is to retrieve some number of the most relevant documents, even if some of them are not particularly relevant to the query. I.e., typically, there is not a hard cutoff for the relevance, where we don't show documents that fall below this threshold. In the context of a fact check retrieval service, we do need such a cutoff. Showing the user fact checks whose relevance is extremely low is likely to make the user ignore all fact checks in the future and should hence be avoided.
3. As discussed in the previous section, we are interested not just in fact checks that address the specific claim in the article, but also those that address other claims that fall in the general theme of the article's claim, if there is one. This requires a corpus level semantic model of the themes that occur.

We used a set of techniques to determine the relevance of a fact check to an article. In the results section, we present the performance of various combinations techniques.

3.4 Vector Space Model

We start with the traditional vector space model of documents and cosine similarity between the Term Frequency Inverse Document Frequency (TFIDF) vectors corresponding to articles and fact checks to determine relevance. While term frequency is easy to compute between the article that the user is looking at and the fact check, inverse document frequency is defined against a specific corpus and can be computed in different ways in our application:

- We could compute it from a large sample of randomly chosen web pages. Lists of term frequencies computed from such corpora are available from many sources. In this work, we used the list from [15]. However, given that the bulk of the articles that require fact checks are drawn from a very small subset of all the topics that web pages cover, we found that these lists, at least by themselves, did not yield good results.
- We could regard the text of the fact checks and corpus collected (as described earlier) as being representative of all articles requiring fact checks and compute term frequencies from this. This manages to capture some of the idiosyncrasies of articles that require fact checks, but the relatively small size of the corpus fails to identify some terms that are quite common but happen to not occur that frequently in this set of articles.
- The claimReviewed property of a fact check is a concise summary of the claim checked by a fact check. In a moderate proportion of cases, the title of the article is similarly a good reflection of the claim made by the article. So, another alternative is to compute the corpus frequency of a term by using the claimReviewed field of fact checks and titles of articles in our corpus.

In practice, each of these approaches has its strengths. We calculated term frequencies using all three approaches and computed a final term frequency as a weighted product of three.

For the term frequency, similarly, we can do the matching purely on the article/fact check titles and claimReviewed or do the matching on the whole document. We implemented both approaches and present the results in the next section.

3.5 Capturing themes with Topic Modeling

While the basic TFIDF model is easy to implement, it does not capture the phenomenon of recurrent themes in fake stories. Our goal is to create a model for each of themes and use that as a feature in matching stories to fact checks. More specifically, if a fact check and story are about the same theme(s), then the fact check should be considered more relevant to the story. We note that the fact checks themselves reflect the themes. For example, we find a number of fact checks corresponding to the theme of vaccines being harmful, another set corresponding to the theme of climate change being a hoax and so on. So, we try to identify the major themes by analyzing the fact checks.

We use Topic Modeling on the corpus of fact checks to identify themes. Starting with Blei, Jordan, et. al. [16] there has been substantial work on identifying a set of

"topics" which can be combined in different proportions to generate the articles (modeled as a bag of words) in a corpus. Topic Modeling has become an effective tool for the discovery of underlying semantic themes in document corpora. In this work, we use Topic Modeling as a tool for identifying recurring themes in fake news stories. Since the corpus of fact checks reflects the topics covered in fake stories, we can use our fact check corpus to generate a set of topics. There are several widely used packages that can be used to generate topics. We use the ldaModel class in the Gensim [17] tool.

In Topic Modeling, each article is modeled as being composed of a set of topics in different proportions. Each topic is a bag of words. Some of these topics correspond to the structure of the language and include words found in most documents. Such topics, which are part of most documents, don't help much with thematic understanding. We preprocess our corpus to remove all words that appeared in more than half the documents (naturally eliminating stopwords, etc.). With Topic Modeling, one of the input parameters is the number of topics desired, which could be a function of the application. In our case, after some experimentation, we settled on 300 topics.

Once we derive our model, we can use it to assign topics to any document — article or fact check. Given a bag of words from the document — words from the title, claim reviewed or the whole content of the article — the model can be run to give us a set of topics (and relative proportions) corresponding to that bag of words. The number of topics (and their proportions) shared by an article and fact check can be viewed as a measure of their similarity.

3.6 Semantically meaningful topics

Each article is modeled as being composed of a set of topics, in different proportions. Each topic can be thought of as capturing some aspect of the document. By removing the most commonly occurring words, we eliminated the topics that capture the structure of language, including pronouns, articles, etc. Some of the topics generated (from the more unique words) capture stylistic aspects of particular publishers. For examples, a fact check from Washington Post has certain stylistic elements that are not found in a fact from a publication such as Snopes. We do find that a small number of topics capture the kind of themes we are aiming to capture. Clearly, an article and fact check matching on one of these topics is more salient than their matching on one of the other topics.

We had two raters go through the topics and identify those that were clearly associated with particular themes and picked the topics on which the raters agreed. These were our 'Thematic Topics'. We computed two measures of similarity based on topics — one based on the number of topics (thematic or not) that the article and candidate fact check shared and another based on the number of thematic topics shared.

Table 2 lists some of the topics identified as being thematically meaningful. Table 3 lists some of the topics that were not identified as being thematic. Please note that the actual terms in these topics are stemmed, but for the sake of readability, we have used one of the unstemmed versions of the words in these tables.

In summary, we have the following features and their variations: (1) tfidf, on document titles or content (2) topics, evenly weighted or more weight for thematic topics. In addition to testing the performance of the individual features, we also evaluated a linear combination of these features, with coefficients hand tuned on a set of 20 sample pages.

Birther controversy	obama, certif, hawaii, kenya, birther, 2008, obama, blumenthal, citizenship, ...
Vaccines	vaccine, infant, flu, cdc, monument, hpv, gardasil, pediatr, autism,
Climate	climate, carbon, emission, pollute, epa, co2, fossil, earth, atmosphere, ...
LGBTQ	gender, bathroom, transgender, gay, discriminate, lgbt, ...
Refugee crisis	refugee, immigrant, vetting, resettle, syria, migrant, iraq, ...

Table 2: Sample of thematic (semantically meaningful) Topics

Topic 28	cash, citation, trait, miracl, addict, 2015, 135, undesir, eo, liquor, ...
Topic 32	deadbeat, pure, clue, five, exist, none, pig, insert, hud, poppi, ...
Topic 44	“none”, “peanut”, “morgan”, “lynch”, “allergi”, “hr”, “ingredi”, “trace”,
Topic 45	none, mail, satan, injury, ration, singer, sir, temple, microphone,

Table 3: Sample of topics that don’t seem to correspond to themes in the fact checks

3.7 Evaluation

We randomly picked 100 articles from the corpus described earlier and evaluated the performance of the different features and their combination. For each article, we computed the five of the most related fact checks whose score was above a certain threshold. In some cases, there were less than five fact checks above the threshold. For each article and fact check pair, we assigned one of the following scores.

1. On Claim: The fact check addresses (one of) the main story of the article.
2. On Theme: The fact check does not address the main story of the article, but addresses other stories that are in the same storyline or theme of the article. Most importantly, these fact checks would help the reader understand the article and place it in context.
3. Irrelevant: The fact check is irrelevant to the article. Presenting the reader with this fact check is likely to make them less likely to ask for relevant fact checks in the future due to it simply being a distraction.

For example, given an article published by the Observer claiming “The Clinton Foundation Shuts Down Clinton Global Initiative”, the fact check from Snopes checking the claim “The Clinton Foundation is shutting down due to lack of donations” would receive a 1. A fact check from PolitiFact about the salaries of the Clintons for the Clinton Foundation would receive a 2 as it could help the user place the article in context. Lastly, fact checks that have no association with the topic such as those concerning ObamaCare would receive a 3.

Most pages on the web don’t have a fact check associated with them. However, because of the way we constructed our corpus of pages to train and test our algorithms, most but not all of the pages have either an “On Claim” fact check or “On Theme” fact check. In the ideal case, we should retrieve at least one “On Claim” fact check, if there are any, and one or more “On Theme” fact checks, if there are any, and no “Irrelevant” fact checks.

4 Results & Discussion

Table 4 gives the results from the evaluation of the 100 randomly chosen stories for the different scoring techniques. Note that the total number of results for different techniques vary. That is because only results whose score is above a certain threshold are returned. Each of the scoring techniques has its own threshold.

Table 5 is a measure of recall and gives us the fraction of pages for which there is an "On Claim" result which was retrieved.

Num	Scoring Type	On Claim	On Theme	Irrelevant
1	TFIDF on Title	91 (24%)	99 (23%)	202 (52%)
2	TFIDF on Page Content	128 (25%)	188 (30%)	219 (43%)
3	Unweighted topic match	67 (10%)	153 (25%)	310 (63%)
4	Weighted topic match	65 (11 %)	182 (35 %)	257 (53 %)
5	1 and 4	81 (15 %)	201 (36 %)	266 (48 %)
6	2 and 4	130 (23 %)	232 (41 %)	198 (35 %)

Table 4: Number of results (and fractions) in each category for each scoring type

Num	Scoring Type	On Claim Recall
1	TFIDF on Title	.72
2	TFIDF on Page Content	.85
3	Unweighted topic match	.62
4	Weighted topic match	.55
5	1 and 4	.60
6	2 and 4	.85

Table 5: Fraction of articles that have 'On Claim' fact checks for which at least one of these fact checks was retrieved.

From table 5 we can see that with a hybrid approach, where we combine TFIDF on the content together with thematic topics, we are able to retrieve both "On Claim" and "On Theme" results a very high fraction of the time. We can also see that the fraction of pages for which at least one "On Claim" result is retrieved, when there is one, is quite high. However, we note that the number of "Irrelevant" results is still quite high.

Surprisingly, just using topics, with and without extra weights for topics corresponding to themes, performs as well as TFIDF on the content of article for retrieving 'On Topic' as content, but not on "On Claim". Just using topics doesn't do as well on "On Claim", because though topics tend to be a good distillation of the overall content, they do poorly on capturing the precise semantics of a particular claim. Weighted topics gives fewer off-topic results, since some of the topics tend to capture aspects of documents that have more to do with the writing style of different publications than with the content of the story.

While content-based matching can be effective, "On Theme" improves with addition of topics. Not surprisingly, matching based on topics is better than TFIDF at capturing the theme of articles.

Just matching on titles (and claimReviewed) does better than expected, likely because a number of titles do provide a good synopsis of the article. However we do see a

category of poor retrieval that arise out of the fact that many fake articles tend to to use titles that are targeted more at attracting clicks than conveying a summary of the article, e.g., articles with titles like "You won't believe who just endorsed ...". In these cases, the title is less effective than the content in matching fact checks.

Titles are small, enabling lightweight service that may even be bundled into the browser extension so that this service can be provided without loss of privacy. Title matching can be improved by the use of word similarity metrics. For example, with strict matching, the words 'immigrant' and 'refugee' will not match. However, the two words are more similar to each other (and almost replaceable in the language of article titles) than most other words. We are currently investigating the use of embedding techniques (such as Word2Vec) for doing this.

5 Conclusion

In this paper, we introduced a new service for helping users make better judgments about articles they read, especially when they make claims that might seem fake. Given the scale and scope of fake news, it is clear that we need computational tools to combat this phenomenon. The problem is more nuanced than simply training classifiers that classify articles as being fake or true. Not only are there many stories that occupy a gray zone, but also the goal should be to give the user more context so that they can more critically read an article. The emergence of a number of fact checking organizations and their extensive adoption of Schema.org schemas offers an opportunity for a new kind of service that serves these needs.

We introduce the concept of a "Related Facts Checks" service that enables a user to get a set of fact checks that may be relevant to an article that they are reading. We describe how such a service may be built, discuss alternative approaches and present results from an early implementation of this service. As demonstrated by the evaluations, even our early implementation offers results that we believe will be helpful to a user.

The work presented here is just a first step and opens up many new directions. In particular, our implementation uses a number of parameters/weights for combining different techniques. In the initial implementation discussed in this paper, these weights were manually adjusted. This was done in part because we had very few labeled examples and the use of machine learning techniques would have lead to over fitting. In future, as we get more training data, we hope to explore the use of machine learning to automatically tune the system. The approach described in this paper made significant use of Topic Modeling. Today's Topic Modeling tools generate many topics for a given corpus. We showed how identifying and giving greater weight to semantically or thematically more meaningful topics improves the performance of the system. In this work, we identified such topics manually, which poses a challenge as the number and scope of fact checks increases. Another direction of future work is to automatically identify which of the topics satisfy this condition.

Much of the effort behind this work was in curating the corpus of fact checks and articles. We believe that the availability of open dataset, preferably one that allows others to contribute to it, will greatly facilitate research on this important topic. In that spirit, we plan to make the data collected by us available on GitHub.

6 Acknowledgements

I would like to thank Dave Story for feedback on drafts of the paper. I would like to thank K. Mahesh for help with accessing the Schema.org markup data. I would also like to thank Dr. Christy Story, Kyle Barriger, Dave Lowell and Ann Greyson.

References

1. Hunt Allcott and Matthew Gentzkow. Social media and fake news in the 2016 election. Technical report, National Bureau of Economic Research, 2017.
2. Dissecting the PizzaGate Conspiracy Theories, 2016.
3. About politifact. *at politifact.com/about/.*
4. Fact checker — washington post. *at www.washingtonpost.com/news/fact-checker/.*
5. Duke reporters lab. *at reporterslab.org*.
6. Schema.org. *at schema.org*.
7. Niall J Conroy, Victoria L Rubin, and Yimin Chen. Automatic deception detection: Methods for finding fake news. *Proceedings of the Association for Information Science and Technology*, 52(1):1–4, 2015.
8. Justin Chiu, Ajda Gokcen, Wenyi Wang, and Xiaohua Yan. Classification of fake and real articles based on support vector machines. 2013.
9. Fake news challenge. *at fakenewschallenge.org/.*
10. Getting real about fake news. *at www.kaggle.com/mrisdal/fake-news*.
11. Fang Jin, Edward Dougherty, Parang Saraf, Yang Cao, and Naren Ramakrishnan. Epidemiological modeling of news and rumors on twitter.
12. Eugenio Tacchini, Gabriele Ballarin, Marco L. Della Vedova, Stefano Moret, and Luca de Alfaro. Some like it hoax: Automated fake news detection in social networks. *CoRR*, abs/1704.07506, 2017.
13. Aditi Gupta, Hemank Lamba, P Kumaraguru, and Anupam Joshi. Faking sandy: Characterizing and identifying fake images on twitter during hurricane sandy.
14. Aditi Gupta and P Kumaraguru. Credibility ranking of tweets during high impact events. In *Proceedings of the 1st Workshop on Privacy and Security in Online Social Media*, PSOSM '12, pages 2:2–2:8, New York, NY, USA, 2012. ACM.
15. Common english terms. *at wordfrequency.info*.
16. David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.
17. gensim: Topic modeling for humans. *at radimrehurek.com/gensim/.*