Game of FAME: Automatic Detection of FAke MEmes

Bahruz Jabiyev
Northeastern University
Boston, MA
bahruz@ccs.neu.edu

Jeremiah Onaolapo
University of Vermont
Burlington, VT
jeremiah.onaolapo@uvm.edu

Gianluca Stringhini
Boston University
Boston, MA
gian@bu.edu

Engin Kirda
Northeastern University
Boston, MA
ek@ccs.neu.edu

Abstract

Memes nowadays are ubiquitous on the Web and play a major role in disinformation campaigns. It is therefore not enough to tackle only the problem of textual disinformation. The research community must also develop new techniques to address the problem of malicious memes (fake memes) that contain misattributed or fabricated quotes, for instance, in online smear campaigns that target politicians and celebrities. To address this problem, we develop a system to automatically detect fake memes; our approach leverages optical character recognition, natural language processing, image processing, and machine learning techniques to carry out this task. Our implementation, a system named FAME, relies on various features to detect visual memes that contain fake or misattributed quotes. FAME classifies memes with 84% true positive rate and 14% false positive rate. It can be used for early detection of meme-based disinformation campaigns, for instance, if deployed on online social networks or messaging applications. To the best of our knowledge, FAME is the first automatic fact-checking tool for memes.

1 Introduction

Recent developments have demonstrated a relatively new mode of information warfare: attempts were allegedly made to influence the 2016 US presidential elections, among others, via coordinated disinformation campaigns on the Internet. Hordes of fake news articles (Allcott and Gentzkow, 2017), politically-motivated images (Zannettou et al., 2019a), and targeted ads (Wakefield, 2018) on online social networks (OSNs) played major roles in the push to sway public opinion and manipulate elections.

Images play an interesting role in information warfare: Zannettou et al. (Zannettou et al., 2019a) reported that state-sponsored actors "do not only use textual content, but also take advantage of the expressive power of images and pictures." Memes—a popular Internet vehicle of information that often involves attention-grabbing images—have also been co-opted by such actors; they create and disseminate memes with biased political messages, usually via OSNs and other online communities (Zannettou et al., 2018). Figure 1 shows an example of a politically-inclined meme.

Figure 1: A quote attributed to former British prime minister, Winston Churchill, which was determined to be a misattribution by Snopes, a fact-checking organization.

Previous work has studied fake news on the Internet and developed techniques to automatically detect fake news (Zhou and Zafarani, 2018). Despite these efforts, fake news is an ongoing problem and deserves further attention: early detection of fake news is one of the open challenges identified by Zhou and Zafarani (Zhou and Zafarani, 2018). On a related note, image-based disinformation, for instance via politically-charged memes, is an understudied field. Hence, we focus our attention on this research gap: we aim to detect image-rich disinformation content in order to mitigate disinformation campaigns on the Internet.

In this paper, we address the problem of fake memes—these contain messages, fabricated or otherwise, falsely attributed to specific individuals. Such memes could be deployed against political opponents during smear campaigns, for in-

stance. Our approach leverages Optical Character Recognition (OCR), Natural Language Processing (NLP), image processing, and machine learning techniques to detect memes that contain fake or falsely-attributed content, as previously described. Our implementation, a system named FAME (a contraction of "FAke MEmes"), relies on several information feeds to carry out its task: reputable news sources, quotation websites, verified social media accounts, and public government websites. FAME achieves 84% true positive rate and 14% false positive rate.

There is a caveat associated with FAME's false positive rate: meme classification is a hard problem that involves many complex interconnected tasks, including OCR, face recognition, and NLP, each with its own limitations. In Sections 6 and 7, we discuss how these limitations contribute to false positives. We also suggest potential ways to improve future instantiations of FAME; a key recommendation is to use high-performance proprietary OCR tools rather than free OCR tools (we used a free one in this work). Similarly, using proprietary tools for the other components—for instance, NLP and face recognition—would drastically reduce FAME's false positive rate.

FAME can be deployed by various digital platforms to stem the flow of meme-based disinformation campaigns. FAME's end goal is to make the Internet safer for the general public. Our contributions are as follows.

- We identify features for the classification of fake memes; these include reputable news sources, quotation websites, verified social media accounts, and public government websites.
- We develop a novel approach for the automatic detection of fake quotes and falsely-attributed quotes in images.
- We make the source code of FAME available to the public so it can be deployed by OSNs, messaging apps, and other platforms to stem image-based disinformation campaigns. The code is publicly available on the authors' websites.
- We evaluate FAME's performance and discuss potential ways to improve it.
- We create a labeled dataset (FAME dataset) which contains 1000 fake and real quote memes, for future research into understanding and mitigating disinformation campaigns on the Internet. The dataset is publicly available on the authors' websites.

2 Background and Related Work

To help the reader understand the remainder of this paper, this section presents the three main themes that comprise the foundation of our work: fake news, memes, and fact checking.

2.1 Fake News

Fake news, according to Allcott and Gentzkow (Allcott and Gentzkow, 2017), comprises "news articles that are intentionally and verifiably false, and could mislead readers." Although fake news is not a new phenomenon, (Soll, 2016) it again came into the public spotlight during the 2016 US presidential elections, in which political actors allegedly attempted to manipulate public opinion via fake news and other methods. Unfortunately, current efforts to stem the spread of fake news have not yet recorded much success (Lee, 2016). Prior work on the detection of fake news includes (Tacchini et al., 2017; Tschiatschek et al., 2018; Zhou et al., 2015; Jin et al., 2016; Volkova et al., 2017; Liu and Wu, 2018; Ruchansky et al., 2017; Wang et al., 2018; Yang et al., 2018). Other studies on the propagation of false or malicious information include (Zannettou et al., 2019b; Zhou and Zafarani, 2018; Zannettou et al., 2017; Hine et al., 2017; Zhang et al., 2018).

2.2 Memes

According to Richard Dawkins (Dawkins, 1976), a meme—analogous to a gene—is an idea or unit of culture that is replicated and transmitted among people. Internet memes, often comprising catchy images and text, are transmitted via numerous online communities and social networks, sometimes for comedic effect, and other times with malicious intent. Internet memes are ubiquitous nowadays, and successful memes spread rapidly through various online communities (Bauckhage, 2011; Zannettou et al., 2018). Hence, memes are attractive to malicious actors who intend to carry out disinformation campaigns (Zannettou et al., 2019a). Memes often originate from fringe online communities (Zannettou et al., 2018) and then spread to the rest of the Web. For instance, 4chan, an online

message board, is reportedly the source of many popular politically-charged memes (Hine et al., 2017).

2.3 Fact Checking

Fact checking is one of the approaches that have been deployed to tackle fake news. At its core, fact checking involves comparing news content to well-established facts to ascertain if the news content under test is true or not. It can be carried out manually (by credible domain experts) or automatically (using information-processing software). Manual fact-checking, although often accurate, does not scale well, given the sheer volume of content that online communities produce daily (Zhou and Zafarani, 2018; Zannettou et al., 2018). Existing fact-checking services include Snopes, PolitiFact and FullFact which provide manual fact-checking services. Memechecker.net is another fact-checking service which focuses its efforts only on memes, while listing much fewer – less than a dozen – fact-check reports than the mentioned fact-checker organizations. Since memes play a vital role in disinformation campaigns as discussed in Section 2.2, our work aims to provide a scalable solution to the problem of fact-checking memes. In other words, we propose an automatic meme fact checker to help increase the scale of fact checking and minimize the potential psychological harm that human fact checkers encounter during their work.

3 Problem Statement

Fake quote memes comprise images which contain fake quotes, usually attributed to well-known people. They exist mainly in three forms: memes with fabricated quotes (made up), slightly-modified real quotes (slight change in the text, usually significant change in the meaning), and misattributed quotes (real quotes that actually originated from someone else not present in the meme). We focus on memes that contain fabricated or misattributed quotes.

We only address quote memes which attribute exactly one quote to one person, for technical reasons. We exclude memes that contain several persons or multiple quotes. Consider the worst-case scenario: a quote meme that contains several persons and multiple quotes. Limitations in image processing and OCR techniques prevent us from successfully matching such quotes to individual persons on the meme. Hence, as earlier mentioned, we focus on simple quote memes: one person, one quote. Figure 1 shows such an example.

Purveyors of false information via memes do not always include quotes in their memes. Sometimes, they opt for doctored images without text captions. For example, they might take a picture from a gruesome murder scene and edit it to replace the victim's face with the face of their target (say, a politician or celebrity). Such memes are out of scope in this work; they require a different approach than ours.

Our work aims to protect vulnerable online communities and digital platforms that double as sources of information, from certain types of image-based disinformation campaigns. Scenarios in which our work will be directly applicable include the following: (1) a journalist may be targeted with malicious fake quote memes in the course of reporting sensitive events, and (2) OSNs, which double as news sources for millions of people, may be contaminated with fake quote memes to defame high-profile individuals, for instance, during elections.

4 An Overview of Our Approach

Figure 2 illustrates an overview of FAME's steps in classifying meme quotes. Next, we discuss each step in detail.

4.1 Extracting Meme Text

To extract text from the input meme, we use OCR, a technique commonly used to extract text content from images and Portable Document Format (PDF) files. Factors that may affect the quality of OCR include color contrast between the text and background, font family of the text, and the amount of distortion in the text segment of the image. In Section 6, we discuss the performance of text extraction and how it affects the meme classification process.

4.2 Identifying the Subject

To identify the subject (person) to whom the quote is attributed, we use two techniques: recognition of the person's *face* from the meme and recognition of the person's *name* from the caption of the meme. To identify the person on a meme, we first perform face recognition on the meme. However, quote memes do not always display a face; instead, some include the person's name in text only. Also,

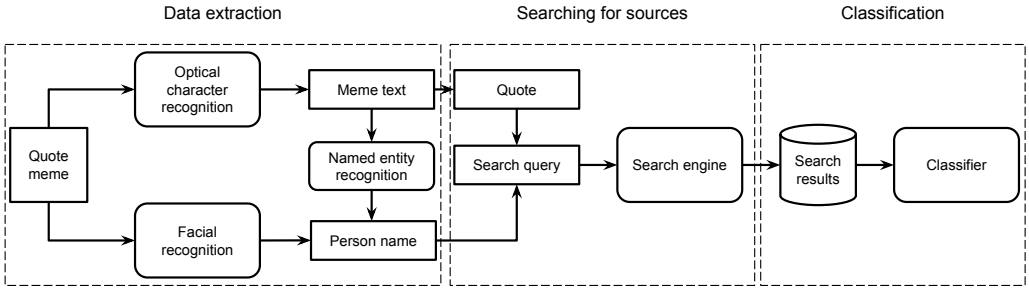

Figure 2: An overview of our meme classification pipeline.

face recognition sometimes fails. In such cases, we attempt to deduce the name of the subject from the text extracted from the meme (using OCR). For this, we leverage Named-Entity Recognition (NER), a technique for identifying names of various entities (for instance, people, organizations, and places) within a body of text.

4.3 Searching for Sources

To search for sources, we first have to obtain the meme quote in question—it has to be included in the search query. To find the quote within the text extracted using OCR, we retrieve the text segment that is enclosed in double quotation marks. However, sometimes we fail to extract the quote for two reasons: either OCR fails to recognize double quotes or the meme text does not contain double quotes. Our observations reveal that non-quote text, which sometimes appears on memes alongside quote text, might prevent search engines from retrieving sources for the quote when included in the search query. Hence, we construct search queries in two distinct ways, depending on our ability to find the quote in the OCR-extracted meme text. We discuss them next.

Success during quote extraction. When the OCR-extracted text contains a pair of double quotation marks with a body of text between them, we assume that body of text is the quote. We include the extracted quote in our search query as it is.

Failure during quote extraction. If OCR fails to recognize double quotation marks in the text or the text actually does not contain quotation marks, we split the text into sentences—knowing that at least one of them belongs to the quote segment—and construct a separate search query for each of them. Search queries which contain a sentence from the quote text will return sources for the quote (if they exist), while search queries which contain content from non-quote text will not return such sources. Additionally, we construct yet another search query using the whole OCR-extracted text.

Constructing search queries. We take the following steps to construct a search query whether we succeed or fail to identify the quote segment. First, we remove misspelled words to avoid confusing the search engine. Second, we trim the search query to its first n words—Appendix A discusses how we arrived at this—having observed that the first n words were sufficient for the search engine to recognize the quote. Third, we add the name of the subject (person on the meme) to the beginning of the search query, because it helps the search engine to return more relevant results. Finally, we submit search queries to the search engine—or only one search query if the quote segment has been successfully retrieved, as discussed previously.

Outcome. We combine the search results to create a *pool of retrieved search results* after removing duplicate results.

4.4 Identifying Relevant Search Results

Not all search results returned by the search engine will be relevant. To identify the relevant ones from the pool of retrieved search results, we test two conditions: quote condition (to ensure that a search result page includes the quote) and name condition (to ensure that a search result page contains the name of the person on the meme).

Outcome. We create a *pool of relevant search results* by applying both conditions to the pool of retrieved search results.

4.5 Classification

In this section, we discuss several features of the pool of relevant search results that serve as inputs for the meme classification task. We enter those features into a trained machine learning model to

compute the probability of the input quote meme being fake or real.

Highly-trusted news sources. These comprise a selection of news sources that have established a strong reputation, over decades, of reliable and accurate reporting, and by having high standards of reporting. We refer to them as highly-trusted sources throughout this paper. We compute the number of these sources from the pool of relevant search results.

Legitimate news sources. There is a large number of online news sources which do not necessarily carry out in-depth investigation and reporting as highly-trusted sources do, yet are known to be reliable. We call these sources legitimate news sources. We count the number of legitimate news sources in the pool of relevant search results.

Quotation websites. Compared to well-known living people today, it is harder to find quotes of well-known historical figures in news sources. Hence, we use quotation websites as sources when searching for a quote. We count the number of quotation websites in the pool of relevant search results.

Government websites. Government websites are usually reliable sources of quoted information, especially from politicians, who also happen to be common targets of fake quotes. Hence, we count the number of government websites in the pool of relevant search results.

Verified social media accounts. Finally, we check for the existence of verified social media accounts of the subject in the pool of relevant search results.

5 Prototype Implementation

In this section, we present our implementation of the quote meme classifier which we call FAME (a contraction of "FAke MEmes"). It is based on our general approach to the quote meme classification task, as discussed in Section 4.

5.1 Extracting Meme Text

To extract text from a meme, we use a free OCR API called *OCR.space*.[1] This API receives image information either via a URL that points to an image, or the image itself as a base64-encoded string, and returns the extracted text. We discuss the performance of this API in Section 6, with emphasis on how it affects the performance of our classification model. If the extracted text from a meme does not contain a sentence with at least three words, we discard that meme.

5.2 Identifying the Subject

To identify the person on a quote meme, we use two techniques: face recognition and name recognition, as mentioned in Section 4. For face recognition, we rely on the image search function that the Bing search engine provides. To this end, we craft an HTTP request, include the URL of the quote meme in it, and carry out an image search on Bing. We then retrieve the name of the person on the meme from the "Looks like" section of the resulting HTTP response. We discuss details of the performance of Bing image search in Section 6.

If face recognition fails, we run person name recognition, otherwise known as Named-Entity Recognition (NER), on OCR-extracted meme text. To this end, we use *CoreNLP*, a Natural Language Processing library, to implement NER. We discuss its performance in Section 6.

5.3 Searching for Sources

Preprocessing. In Section 4, we explained the process of constructing search queries from meme text. This process requires the removal of misspelled words as a preprocessing operation; we use a library called *pyenchant* to achieve this.

Search engine. We chose DuckDuckGo to search for sources. Arguably, using another search engine such as Google might result in better performance. However, Google blocks scripted requests and would not allow us to run as many queries as required; sometimes, in experiments, we made about 2000 requests within a few hours. DuckDuckGo sources results[2] from several partners including Bing and Yahoo. In Section 6, we show that situations in which DuckDuckGo is unable to retrieve sources that Google can fetch, are very few. To query DuckDuckGo, we craft and send HTTP requests, and use only the first two pages of search results to find sources.

5.4 Identifying Relevant Search Results

We use a Python library called *edit_distance* to implement the quote condition (see Section 4.4), which looks for the quote of interest within the page of a search result. To carry out the longest

[1] https://ocr.space/ocrapi

[2] https://help.duckduckgo.com/duckduckgo-help-pages/results/sources/

common subsequence task, we use the value *highest_match_action* for the parameter of *action_function*. Once the longest common subsequence is found, we check it against the threshold value *0.3*, which is the ratio of the length of the longest common subsequence to the length of the quote (or meme text if the quote cannot be extracted). If the length ratio is above that threshold, we add the corresponding search result to the pool of relevant search results. We discuss how we chose this threshold value in Appendix A.

To check the name condition, we use a simple regular expression to search for the identified person's name in a search result page.

5.5 Classification

To implement the FAME classifier, we use *scikit-learn*, a Python library. As we show in Section 6, Support Vector Machine (SVM) with rbf kernel yields the best results in the quote meme classification task. To extract information regarding the features mentioned in Section 4, we do the following.

Highly-trusted news sources. To implement this feature, we create a list of news sources based on the results of two separate public surveys conducted by Pew Research Center (Center, 2014) and Reynolds Journalism Institute (Kearney, 2017). This list comprises about 30 different news sources (see Appendix C). We use it to identify the number of highly-trusted sources in the pool of relevant search results, by counting how many domain names of search results match domain names in the list of highly-trusted news sources.

Legitimate news sources. Similarly, we check each search result in the pool of relevant search results against a list of legitimate news sources. This list comprises the Alexa Top 500 newspaper websites in the United States, with a slight modification; we remove highly-trusted news sources to eliminate repetition in counting.

Quotation websites. We check each search result in the pool of relevant search results against a list of quotation websites. For this list, we use Alexa Top 500 Quotation websites; it contains about 130 websites.

Government websites. To identify government websites in the pool of relevant search results, we specifically search for US government websites and use a simple regular expression which checks if the domain name of a search result ends with ".gov" or not.

Verified social media accounts. After we identify search results that point to Twitter or Facebook profiles, we carry out a scripted HTTP request to identify if they are verified or not, and also obtain the full name on the profile. If they are verified, we then check the name of the person of interest (which we extract from the input meme) against the full name on the page.

6 Evaluation

In this section, we discuss our ground truth dataset and evaluate the performance of our classification model. We also discuss the performance of specific components of FAME.

6.1 Ground Truth Dataset

We evaluate our system on a quote meme dataset which we collected ourselves, called the *FAME dataset*. It contains 1000 quote memes in total: 379 fake memes and 621 real memes.

Collection. First, we identified 20 well-known individuals (see Appendix D) who are commonly targeted by fake quote memes, by analyzing the fact-check history of three main fact-checking organizations: Snopes, FactCheck, and PolitiFact. Next, we used the DuckDuckGo search engine to collect quote memes for each of them by entering "$\{person's\_name\}$ *quote memes*" in the search field. We avoided duplicate quotes across memes during the collection process.

Labeling. To complete the ground truth dataset, we needed binary labels for the memes: "real" or "fake." We followed a set of guidelines to label the memes. First, we searched for the quote on a search engine and examined the search results which contained the quote. We examined the domain names of those search results with the help of a browser plugin that we implemented specifically for this purpose. The plugin colorizes search results of interest, with unique colors, depending on their type: "highly-trusted news source," "legitimate news source," "quotation website," "government website", or "verified social media account" (as discussed in Section 5).

We labeled memes as "real" if they met either of these conditions: (1) they had at least one search result that published the quote and was a highly-trusted news source, government website, or verified social media account that belonged to

Table 1: Performance metrics of the classification model.

Metric	*Performance*
Accuracy	85%
True positive rate (recall)	84%
True negative rate	86%
Precision	79%
F1 score	81%
False positive rate	14%
False negative rate	16%

the identified person, or (2) they had at least two search results that published the quote and both of them were either a legitimate news source or quotation website. On the other hand, we labeled memes as "fake" if they met both of these conditions: (1) no search result, of the previously discussed types, published the quote, and (2) it did not appear credible that the words on the meme were uttered or written by the identified person on the meme. The second condition is necessary because we acknowledge that the absence of reliable sources in search results does not conclusively indicate that the quote is fake; search engines sometimes fail to retrieve sources.

6.2 Classification Performance

Five-fold cross validation. To evaluate our model, we applied the cross-validation technique on the FAME dataset. It involves splitting the dataset into n parts. During each of n iterations, one part is left out for testing and the rest of the dataset is used for training. Overall metrics of the model can be evaluated by computing the average of metrics achieved during each iteration. We achieved the highest performance with SVM classifier with rbf kernel (see the performance of other classifiers in Appendix B). SVM with five-fold cross validation gives the results in Table 1; the FAME classifier achieves 84% true positive rate and 14% false positive rate. Besides these metrics, we also evaluate the time taken during the classification of a quote meme. When we run our system on a Docker container with 16 CPUs and Ubuntu installed on it, the average amount of time taken for classification is about 20 seconds, including data extraction and searching of a meme.

False positives. There are several reasons why our prototype model mistakenly classifies real quote memes as "fake." A common reason for false positives is OCR failure. When OCR drops or misspells a significant portion of the meme text, the subsequent search engine query fails to retrieve sources based on that text. This reason is responsible for one-third of the false positives. We further discuss the performance of the OCR component in Section 6.3.

Another reason is that the lists we use to categorize sources do not—and presumably cannot—include all reliable and legitimate sources of information. When our system cannot match the domain names of search results with its lists, it simply assumes a lack of sources for the quote. We owe another one-third of false positives to this reason.

False positives also arise as a result of failure of the search engine to retrieve sources. In those cases, the search engine becomes confused by either non-quote text or inadequate quality of extracted text, and therefore fails. This reason accounts for one-fifth of false positives. We also searched for those failed search queries on Google and found that in one-third of those cases, it managed to retrieve sources for the quote.

In very few cases, false positives arise as a result of mistakes in identification of the meme subject. Either face recognition or named-entity recognition may mistakenly identify some other person to be the subject. Such cases confuse the search engine and it fails to retrieve sources.

False negatives. Similarly, our system sometimes misclassifies fake quote memes as "real," for several reasons. The most common reason for false negatives is, for some fake quotes, search results contain common words just as many as to reach the threshold by chance. This happens especially when the quote is short and contains common English words. This accounts for one-fourth of false negatives.

Another common reason is that we fail to identify misattribution on some memes. Usually when a meme attributes Person A's words to Person B, sources for the quote will contain the name of Person A, not Person B. However, in some cases, sources contain both names: the real author and the falsely-attributed author. This accounts for one-fifth of false negatives.

Also, OCR sometimes drops words to an extent where only a few words from the quote remain. When we search for them, search results can easily contain some parts of those few words by chance. This in turn causes our system to reason that those search results are sources for the quote. This ac-

counts for one-sixth of false negatives.

Some sources contain fake quotes, not for publishing, but instead for debunking purposes. Our system cannot distinguish such search results and sees them as sources for the quote. One-sixth of false negatives stem from this reason.

In a few cases, a wrong segment of text is extracted as the quote. Some memes contain quotes in a way that such quotes have some segments enclosed in double quotation marks (nested quotes). Those parts tend to be short, and when searched, some search results happen to have some words in common, in the same order. This accounts for one-tenth of false negatives.

6.3 Performance of Specific Components

Text extraction API. We evaluate the performance of *OCR.Space* (a free OCR tool) on 621 real memes drawn from the FAME dataset. For 5% of real memes, search queries could not be formed because the extracted text did not contain a sentence with at least three words. In another 5% of real memes, the poor quality of extracted text caused the search engine to fail while retrieving sources for the quote; if text quality were good, the search engine would have succeeded in retrieving sources. *OCR.Space* extracted text well enough on 90% of real memes, which allowed the search engine to retrieve sources for them.

Face recognition. Bing image search correctly identified the person on 69% of all memes in our dataset. On less than 1% of all memes, it confused the person on the meme with someone else. It failed to give any result on 30% of memes.

Name recognition. *CoreNLP* NER identified the quote author's name from meme text on 57% of memes that failed face recognition—about 300 memes. It could not extract the author's name correctly from 3% of them. It could not extract any name from the text—either because the name did not exist or it was missed by NER—on 40% of them. A part of *CoreNLP*'s failure can be attributed to the failure of OCR earlier in the pipeline.

7 Concluding Discussion

We focused on a non-trivial problem and developed an approach to detect memes which contain misattributed or fabricated quotes. As we have demonstrated, meme classification is a hard task that involves many interconnected components, each with its own limitations. In Section 6, we addressed those limitations in detail, and discussed how they contributed to FAME's false positive rate. Despite this, FAME's performance shows that our approach can be reliably adopted in practice. Its performance would be even better if we had access to a proprietary OCR tool, rather than the free one we used, and had extensive lists of reliable sources. This also applies to other components of FAME, including NLP and image recognition modules; proprietary tools would perform better and boost FAME's overall performance.

Search engines play a central role in our work: we used a search engine to query the sources of quotes and examine those sources for their reliability. We also searched for the names of people in sources to ensure that the quotes had not been misattributed. Search engines are neither perfect nor the only available tools; there are many other valuable resources and databases, some of which grant free access, while others charge fees. Occasionally during the labeling process, we could not find sources for a quote, using a search engine, and could not label it as "fake" because it did not seem unreasonable that the purported author said or wrote it. However, search engines give free and quick access to large numbers of online resources with a quick search; they are therefore commonly used and highly recommended by fact-checking organizations, for instance, AFP (AFP, 2011), FactCheck (Jackson, 2008) and PolitiFact (Holan, 2014). Our work offers a framework for future research that might use other resources and databases for the identification of fake and real quotes.

Our work offers significant benefits to fact-checking organizations that rely on manual fact-checking processes by experts and cannot handle large numbers of quote memes daily. Our system will help to scale up their work. To further improve their output, they can also compile their own lists of sources that they rely on, instead of using the ones that we compiled for the FAME prototype. In addition to the time-related benefits of scaling up, our approach will also help to minimize potentially harmful content that human fact checkers will be exposed to, which will in turn reduce mental trauma, as mentioned in Section 2.3.

Finally, our approach can also be used by messaging apps and digital platforms that host quote memes. They can leverage our work to automati-

cally detect fake and real quote memes—in a reasonable amount of processing time—uploaded to their platforms. This will help in *early* stemming of disinformation campaigns, towards making the Internet safer for everyone.

Acknowledgments

We would like to thank the anonymous reviewers for their helpful comments. This work was supported by National Science Foundation under award numbers of CNS-1942610, CNS-2114407, and CNS-2127232.

References

AFP. 2011. Fact-checking : how we work. https://factcheck.afp.com/fact-checking-how-we-work.

Hunt Allcott and Matthew Gentzkow. 2017. Social media and fake news in the 2016 election. *Journal of Economic Perspectives*, 31(2):211–36.

Christian Bauckhage. 2011. Insights into internet memes. In *AAAI Conference on Weblogs and Social Media (ICWSM)*.

Pew Research Center. 2014. Political polarization & media habits.

Richard Dawkins. 1976. *The selfish gene*. Oxford University Press.

Gabriel Emile Hine, Jeremiah Onaolapo, Emiliano De Cristofaro, Nicolas Kourtellis, Ilias Leontiadis, Riginos Samaras, Gianluca Stringhini, and Jeremy Blackburn. 2017. Kek, cucks, and god emperor trump: A measurement study of 4chan's politically incorrect forum and its effects on the web. In *Eleventh International AAAI Conference on Web and Social Media*.

Angie Drobnic Holan. 2014. 7 steps to better fact-checking. https://www.politifact.com/article/2014/aug/20/7-steps-better-fact-checking/.

Brooks Jackson. 2008. Obama Quote Rumors. https://www.factcheck.org/2008/08/obama-quote-rumors/.

Zhiwei Jin, Juan Cao, Yongdong Zhang, and Jiebo Luo. 2016. News verification by exploiting conflicting social viewpoints in microblogs. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Michael W. Kearney. 2017. Trusting news project report.

Dave Lee. 2016. Facebook's fake news crisis deepens. https://www.bbc.com/news/technology-37983571.

Yang Liu and Yi-Fang Brook Wu. 2018. Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Natali Ruchansky, Sungyong Seo, and Yan Liu. 2017. CSI: A hybrid deep model for fake news detection. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*.

Jacob Soll. 2016. The Long and Brutal History of Fake News. https://www.politico.com/magazine/story/2016/12/fake-news-history-long-violent-214535/.

Eugenio Tacchini, Gabriele Ballarin, Marco L Della Vedova, Stefano Moret, and Luca de Alfaro. 2017. Some like it hoax: Automated fake news detection in social networks. *arXiv preprint arXiv:1704.07506*.

Sebastian Tschiatschek, Adish Singla, Manuel Gomez Rodriguez, Arpit Merchant, and Andreas Krause. 2018. Fake news detection in social networks via crowd signals. In *Companion of the The Web Conference 2018 on The Web Conference 2018*.

Svitlana Volkova, Kyle Shaffer, Jin Yea Jang, and Nathan Hodas. 2017. Separating facts from fiction: Linguistic models to classify suspicious and trusted news posts on twitter. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*.

Jane Wakefield. 2018. Cambridge Analytica: Can targeted online ads really change a voter's behaviour? https://www.bbc.com/news/technology-43489408.

Yaqing Wang, Fenglong Ma, Zhiwei Jin, Ye Yuan, Guangxu Xun, Kishlay Jha, Lu Su, and Jing Gao. 2018. EANN: Event adversarial neural networks for multi-modal fake news detection. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

Yang Yang, Lei Zheng, Jiawei Zhang, Qingcai Cui, Zhoujun Li, and Philip S Yu. 2018. Ti-cnn: Convolutional neural networks for fake news detection. *arXiv preprint arXiv:1806.00749*.

Savvas Zannettou, Barry Bradlyn, Emiliano De Cristofaro, Gianluca Stringhini, and Jeremy Blackburn. 2019a. Characterizing the use of images by state-sponsored troll accounts on twitter. *arXiv preprint arXiv:1901.05997*.

Savvas Zannettou, Tristan Caulfield, Jeremy Blackburn, Emiliano De Cristofaro, Michael Sirivianos, Gianluca Stringhini, and Guillermo Suarez-Tangil. 2018. On the origins of memes by means of fringe web communities. In *Proceedings of the Internet Measurement Conference 2018*.

Savvas Zannettou, Tristan Caulfield, Emiliano De Cristofaro, Nicolas Kourtelis, Ilias Leontiadis, Michael Sirivianos, Gianluca Stringhini, and Jeremy Blackburn. 2017. The web centipede: understanding how web communities influence each other through the lens of mainstream and alternative news sources. In *ACM Internet Measurement Conference (IMC)*.

Savvas Zannettou, Michael Sirivianos, Jeremy Blackburn, and Nicolas Kourtellis. 2019b. The web of false information: Rumors, fake news, hoaxes, clickbait, and various other shenanigans. *J. Data and Information Quality*, 11(3).

Jiawei Zhang, Limeng Cui, Yanjie Fu, and Fisher B Gouza. 2018. Fake news detection with deep diffusive network model. *arXiv preprint arXiv:1805.08751*.

Xing Zhou, Juan Cao, Zhiwei Jin, Fei Xie, Yu Su, Dafeng Chu, Xuehui Cao, and Junqiang Zhang. 2015. Real-time news certification system on sina weibo. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15 Companion.

Xinyi Zhou and Reza Zafarani. 2018. Fake news: A survey of research, detection methods, and opportunities. *arXiv preprint arXiv:1812.00315*.

A Setting Thresholds and Parameters

Length ratio threshold. This threshold determines if a search result meets the quote condition; it checks if a search result published the quote. We tried multiple values—between 0.25 and 0.5—for this threshold and picked 0.3 which ensures the best balance between precision and recall.[3] It also yields high accuracy. Table 2 shows changes in performance metrics relative to the performance achieved by the baseline value (0.3).

Table 2: Changes in performance metrics relative to the performance achieved by the baseline length ratio threshold (LRT). The baseline LRT is in boldface.

LRT	*Accuracy*	*Recall*	*Precision*	*F1 score*
0.25	-3%	-12%	0%	-6%
0.30	**85%**	**84%**	**79%**	**81%**
0.35	-1%	+3%	-3%	0%
0.40	-1%	+5%	-5%	0%
0.45	-2%	+8%	-7%	-1%
0.50	-4%	+8%	-10%	-2%

Optimal query length. During the labeling process, we observed that it suffices to use the first 20 words (approximately) of the quote to search for its sources. Therefore, we tried multiple values around that length—15, 20, and 25 words—and decided on 20 words: using 20 words resulted in 1% better recall than 15 words, and 1% better recall and precision than 25 words.

Window size. This parameter comes into play when we want to limit the distance between found words in search results to ensure that those found words are adjacent. Initially, we set the window size to twice the length of the quote. However, we achieved better performance—2% better recall—when we set it equal to the length of the quote.

[3] Recall is also known as the true positive rate.

B Comparison with other classifiers

We also compare our classification algorithm, SVM with rbf kernel, with other algorithms. As Table 3 shows, other classification algorithms also performed well and some surpassed our classification algorithm in some metrics. Nonetheless, we chose SVM with rbf kernel to create a balance between precision and recall, and at the same time achieve the highest accuracy and F1 score.

Table 3: Performance of other classification algorithms compared to our choice. Our choice is in boldface.

	Accuracy	*Recall*	*Precision*	*F1*
SVM (rbf)	**85%**	**84%**	**79%**	**81%**
SVM (linear)	82%	89%	71%	79%
Random Forest	84%	84%	77%	80%
KNN-3	82%	88%	71%	79%
Adaboost	84%	85%	76%	81%

C Highly-trusted Sources

- ABC News
- Associated Press
- BBC
- Bloomberg
- CBS News
- CNN
- Dallas News
- Fox News
- Google News
- Los Angeles Times
- MSNBC
- NBC News
- NPR
- PBS

- Politico
- Reuters
- The Atlantic
- The Denver Post
- The Economist
- The Guardian
- The Kansas City Star
- The New York Times
- The New Yorker
- The Seattle Times
- The Wall Street Journal
- The Washington Post
- TheBlaze
- Time
- USA Today
- Yahoo News

D Well-known Individuals

- Alexandria Ocasio-Cortez
- Barack Obama
- Ben Carson
- Bernie Sanders
- Bill Murray
- Donald Trump
- Elizabeth Warren
- Hillary Clinton
- Ilhan Omar
- Kurt Russell
- Melania Trump
- Michele Bachmann
- Michelle Obama
- Nancy Pelosi
- Ronald Reagan
- Ruth Bader Ginsburg
- Sarah Palin
- Stacey Abrams
- Ted Cruz
- Winston Churchill