The Kauwa-Kaate Fake News Detection System: Demo

Abhishek Bagade
IIT Bombay

Ashwini Pale
IIT Bombay

Shreyans Sheth*
CMU

Megha Agarwal
IIT Bombay

Soumen Chakrabarti
IIT Bombay

Kameswari Chebrolu
IIT Bombay

S. Sudarshan†
IIT Bombay

ABSTRACT

Fake news spread via social media is a major problem today. It is not easy with current-generation tools to check if a particular article is genuine or contains fake news. While there are many Web sites today that debunk viral fake news, checking if a particular article has been debunked or is true is not easy for an end-user. Search engines like Google do not make it easy to check a complete article since they limit the number of query keywords. In this paper, we outline the architecture of the Kauwa-Kaate system for fact-checking articles. Queried articles are searched against articles crawled from fact-checking sites, as well as against articles crawled from trusted news sites. Our system supports querying based on text as well as on images and video; the latter features are very important since many fake news articles are based on images and videos. We also describe the user interfaces which we will use to demonstrate the Kauwa-Kaate system.

KEYWORDS

fake news, fact checking, information retrieval

ACM Reference Format:
Abhishek Bagade, Ashwini Pale, Shreyans Sheth, Megha Agarwal, Soumen Chakrabarti, Kameswari Chebrolu, and S. Sudarshan. 2020. The Kauwa-Kaate Fake News Detection System: Demo. In *7th ACM IKDD CoDS and 25th COMAD (CoDS COMAD 2020), January 5–7, 2020, Hyderabad, India.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3371158.3371402

1 INTRODUCTION

There is an ongoing epidemic of fake news today, spread in particular by social media. Fake news spread via Facebook has been implicated in influencing the results of elections, and has lead to ethnic cleansing. Closer home in India, fake news spread via WhatsApp has led to many malign effects, ranging from misleading voters and inflaming religious passions, all the way to lynchings of innocent people on false suspicion of being kidnappers.

There have been a number of attempts to curb the spread of fake news. A major contribution in this regard is being made by fact-checking sites such as AltNews, BoomLive, Check4Spam, factly.in, and SMHoaxSlayer. These sites look for fake news that is spreading virally, and debunk them. More recently, many main-stream newspapers and magazines such as Times of India, The Hindu, and India Today, among others, have sections on their websites devoted to debunking fake news. Today, these sites play a major role in helping people check if a particular article (news article or social media forward) is genuine or fake.

Despite the presence of a large number of such sites, social media users in India continue to forward fake news without checking for correctness. A significant factor in this regard is that it is not easy for an individual to check if an article they have received has been debunked or not. Search engines have limits on number of query keywords, and searching with an entire document may not give any meaningful result. In particular, search engines may return not only irrelevant results, but are quite likely to lead users to websites that publish fake news; many such sites look like authentic news sites to the casual user, and may convince the user of the genuineness of fake news. Users who are aware of fact-checking websites can do a search restricted to a single domain, but such search does not scale across multiple fact-checking sites.

Further, forwards often have images which cannot be searched directly; while search engines do support reverse image search, doing so is not easy, especially from mobile phones. Forwards with a combination of text and images make the search task even harder.

The Kauwa Kaate project at IIT Bombay was started in 2018 to address the fake news problem in various ways:

- For end users, the system provides multiple user-friendly interfaces for checking the genuineness of an article. Users can forward articles to a WhatsApp number, or can share the articles with a mobile app, or can copy-paste the text/images into a Web interface. The system checks the submitted articles and images against a repository of fact-checked articles and other articles from trusted news sources, and returns the results in the form of matching articles similar to a search engine result.
- For fact-checking sites, the system provides a way to find what articles users submit for fact-checking, and the number of requests across time. This can help sites prioritize what articles they fact-check. Currently this task is done manually, with humans monitoring a WhatsApp number where users

*Work done at IIT Bombay
†Contact author. Email: sudarsha@cse.iitb.ac.in

CoDS COMAD 2020, January 5–7, 2020, Hyderabad, India

ACM ISBN 978-1-4503-7738-6/20/01.
https://doi.org/10.1145/3371158.3371402

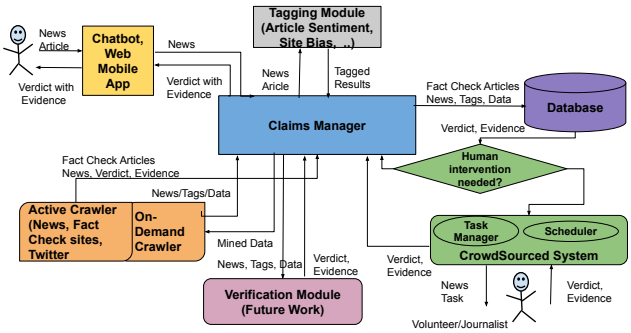

Figure 1: Architecture of the Kauwa-Kaate System

can submit fact-check requests, and by monitoring social media feeds.

- For researchers (including ourselves), our curated and cleaned up crawl of fact-checking and trusted news sites can help provide data for further research. For example, we are exploring how to estimate site bias and trustworthiness using our crawl along with a Twitter crawl.

There have been a number of papers and systems that attempt to automate fact-checking. However, when such sites are used to check even simple facts their limitations become obvious. One problem is that they are designed to check specific claims, not an entire article. A second is that even in the context of checking a single claim, they perform poorly, for various reasons including limitations in their ability to comprehend natural language. And finally, they are generally unable to give a clear explanation for their decision, beyond sharing snippets from web sites which in our experience does not work very well. We therefore believe that fact-checking is, for now, a human activity, although it can be supported by computers. Our goal is to help users easily check if a particular article has been fact-checked, or has appeared in a trusted news site.

2 ARCHITECTURE

Figure 1 shows the high-level architecture of the Kauwa-Kaate system.

A key part of the architecture is the crawling, indexing of articles from fact-checking sites and trusted news sites, and their subsequent use for querying, given an article to be fact-checked. We crawl a number of popular fact-checking websites (including those mentioned in the Introduction, and more) as well as trusted news sites (such as the English sites Times of India and The Hindu and the Hindi site NavBharat Times) every three hours. Since the crawled pages contain extraneous information such as advertisements and links to other articles, in addition to the actual textual content, we scrape the article content using a set of site-specific scrapers. (We tried using generic scrapers, but found that their output quality was not very good.) Our scrapers also extract meta-data that most news sites provide along with the web pages. We then index the extracted textual content and meta-data of the crawled articles using Solr.

Given an article to be fact-checked, we run a query on the index using the entire body of the article; in contrast, commercial search engines use only the first 32 or so words of the article, which we found often results in irrelevant articles being returned.

More importantly, fake news is often characterized by the combination of image and text; taken individually, neither may be fake, but a picture from context A may be falsely be claimed to be from context B. Thus, we need to look up a combination of text and image, to check if an article is fake. We describe how we handle images later in the paper.

When we fail to find an article on a fact-checking site or a trusted news site, we may find the article in a non-trusted website. But trust in a website is a somewhat subjective matter. We believe that a better approach is to establish the bias of a site (or an author) using a variety of information including not just the site content, but links to the site and tweets that link to the site or article. We are currently exploring several alternatives for judging the bias of a site on a particular topic, which will help the user verify the credibility of a given article.

Fake News API & Web front-end: We allow access to our system using a REST API for querying the image and text system separately. The final API combines these. As proof of concept, we have a simple Web client which returns the result to users against queries.

2.1 Handling Images

We index images in the crawled documents in two ways: (i) using hashing on the image content to support exact match, and (ii) by extracting and indexing image signatures (using open-source software) to support approximate indexing. When a forward/article being fact-checked has exactly the same image as that in a fact-checking site, a hash on the image suffices to find a match. However, often the forwards alter the size of the image, so hashes will not match exactly. Further, many fact-checking sites alter the image, for example by stamping images with their site logo. We thus also need approximate match on images. To implement this we used the image match library from https://github.com/EdjoLabs/image-match, based on Wong et al. [9], which creates multiple signature strings from the image, and indexes them using Elastic Search.

A further problem that we encountered is that fact-checking sites often merge multiple images into a single image, and even approximate indexing fails on such merged images. To handle this problem, we are currently working on using image segmentation algorithms to find lines that can meaningfully divide an image into parts. We then index the image segments separately, along with the original image.

Yet another issue that we encountered is that users often share screenshots containing textual information. Just searching on the image does not get any matches since the fact-checking website may not have that screenshot. To handle this problem, we run OCR on images in articles to be fact-checked, using the Tesseract library. We can then perform textual search on the returned text.

When a query contains text and images, we run independent searches on the text and on the images, and then merge the returned results to find the best matches.

If an image is not found in our crawled data, it can be quite useful to run a reverse-image search to find the source of the image or similar images. Fake news with text context not matching the image context can often be detected by such reverse-image search. If the reverse-image search shows the image is from a trusted site, with a context different from that in the article being fact-checked, that makes it likely that the article claims are fake. Therefore, along with our results we give a link allowing users to run a reverse-image search on the original image.

2.2 Handling Videos

Videos are a commonly used component of fake news. Most often videos are clipped to hide context, or used in an entirely different context, claiming to be at a particular location/time, or involving particular people, when in fact they are from a different location/time, or involve other people. Human fact-checkers use various mechanisms to locate the original source of a video, to debunk a claim related to the video. (The InVID project provides some tools to support this task.)

For Kauwa-Kaate, the key issue is how to match a video in a social media forward to a video in a fact-checking website. If the video is shared with no changes, a hash of the video content is good enough to find matches. However, this often does not work since the video may be resized, compressed, or edited in other ways when forwarded. Approximate matching of videos is therefore a requirement for Kauwa-Kaate. While there are commercial tools for this task such as the Tecxipio reverse video search API, we could not find any open-source software for this task. We use two approaches for this task. For longer videos, we use scene-change detection techniques to create a signature based on the lengths of different scenes in the video. This signature can be used to match existing videos by looking for common subsequences with the corresponding signature of the existing video. However, this approach fails for short videos. For such videos, we extract frames at key points (before/after scene change, and at regular intervals), and index the frames as events. By extracting similar frames in a query video, we can match existing videos by using the approximate image matching framework.

An evaluation of the effectiveness of approximate image and video matching is beyond the scope of this paper, but anecdotally, our techniques work reasonably well.

2.3 Crawling Pipelines

One of the major tasks in our system is to crawl and collect relevant news articles and store them for efficient retrieval and search. We use the Scrapy Framework [5] for crawling and scraping the relevant news articles to extract clean data. The Scrapy framework provides sophisticated features for efficient crawling and works well for fetching data across multiple sources. The scraped data from crawlers is processed through multiple data pipelines which perform the following operations.

(1) **Validation Pipeline:** Performs basic validation checks on the extracted data and eliminates blank articles.
(2) **Stat Aggregation Pipeline:** Collects various statistics about the crawler run and the number of articles crawled and inserts into a Database.
(3) **WritetoFile Pipeline:** Dumps the crawled data in JSON files. The files are organized by the site name and the topics it belongs to.
(4) **ImageInsert Pipeline:** Downloads all the images in the articles crawled and indexes it along with metadata in Elastic search back-end.
(5) **IndexSolr Pipeline:** Indexes the crawled article in Apache Solr which we use for efficient retrieval and search operations.
(6) **NLU Pipeline:** Extracts and stores various information about the article body using IBM Watson NLU [1] API.
(7) **OCR Pipeline:** Uses the Tesseract OCR engine [7] to extract text from images crawled in both English and Hindi language. The extracted text is indexed into Apache solr for efficient search and retrieval.
(8) **Video Pipeline:** Extracts signatures from the videos present in the articles and indexes the signatures in Solr for search and retrieval.

2.4 Front End

Our Web and mobile app front ends provide user-friendly UI wrappers on top of the REST API supported by our backend. Since image/video download and upload can be quite slow on mobile phones, we optimize the handling of images and videos as follows. Instead of sending the actual image/video, we first send only a hash of the content to the backend. If there is no match based on the hash, we send the actual image/video to the backend for further processing to get approximate matches.

2.5 Ongoing Work

Ongoing work includes development of interfaces for supporting human fact-checkers. For example, based on queries that our system gets which are not found in fact-checking or news sites, we can determine which articles should get higher priority for debunking. We can also allow route articles to human fact-checkers who are knowledgeable in a particular area, and we can allow them to enter fact-checking results (original article, and a link to the debunking article) on our system.

We are working on better support for the case when a query has an image and text, but we do not find the image/text in our crawled data. An effective way for humans to check such queries is to perform reverse image search to find the original context of the image, and see if it matches the text. We are working on automating this task partially, by using the Google Vision API to perform a reverse image search, and see if the context of the image in the search result (which is provided by the Vision API) matches the query text. If not, we can flag the text as not matching the image context. Giving higher importance to trusted sites, and to older articles (which are likely to be the original source for the image) are important tasks in this context.

Determining site and author bias is another important of ongoing work. It is easy to find twitter handles corresponding to political parties and their key members. We are working on using article links in tweets from such handles with known biases to determine site bias, and in particular to identify politically partisan

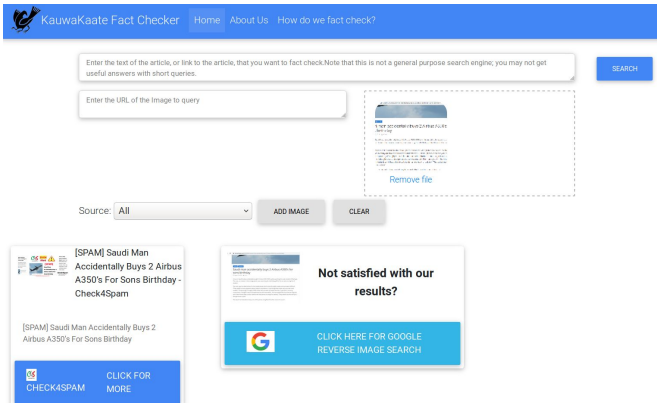

Figure 2: Web Interface Showing Image Search

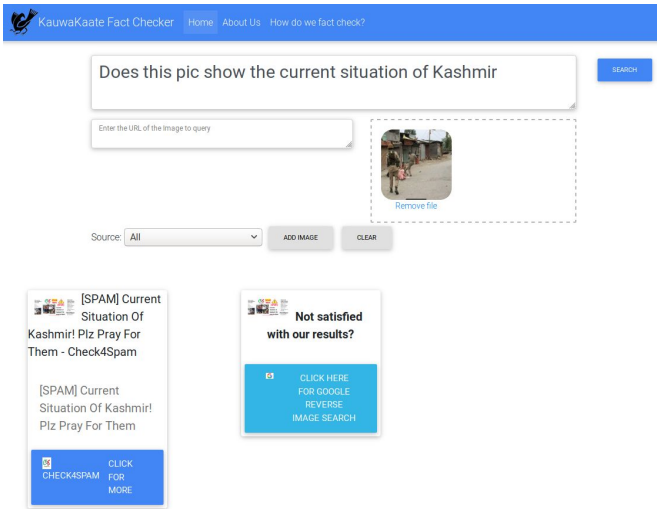

Figure 3: Web Interface Showing Image + Text Search

sites based on the tweets (rather than based on possibly biased human judgement).

3 DEMO DETAILS

Figures 2 shows the result (using a web interface) on a query based on an image which is a screenshot containing text (the actual text is too small to be visible in the figure, but is clear in the original screenshot). Our system runs OCR to extract the text and then runs the textual query to find the result, which may not have been found using pure image search. Figure 3 shows the result (using our web interface) on a query that includes a picture and text. The system also supports queries that only include text.

Figure 4 shows the mobile app and WhatsApp interfaces. The mobile app, Kauwa Kaate Fact Checker, is currently available on the Android AppStore only, and supports the same features as the Web app, but makes it easy to share images directly from WhatsApp to our fact-checking app. Users can directly forward their text/images to a WhatsApp number. Since WhatsApp does not allow programmatic access to content received on WhatsApp, a rooted device

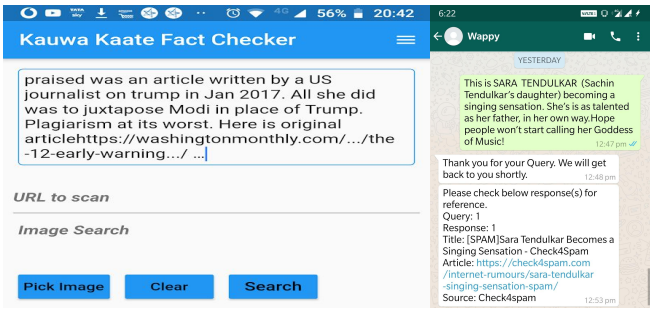

Figure 4: Mobile and WhatsApp Interfaces

had to be used to allow a backend program to retrieve messages received on WhatsApp; the program then uses our API to retrieve results, and sends them back via WhatsApp. Details of how this is done are out of the scope of this paper.

We will invite viewers to try out their own queries on our system, or to try out a variety of queries based on actual fake news that we have received, or retrieved from fact-checking sites, involving text, images, and videos.

4 RELATED WORK

Several systems use end to end machine learning or probabilistic models to give an estimated certainty of the truth of a given text snippet. Examples include CredEye [3], ClaimEval [4] and [8]. They focus on specific snippets, and do not attempt full-article verification. Further they do not deal with images, which are today a key part of fact-checking. And most systems based on automated approaches either do not provide explanations for their conclusions, or offer rather unsatisfactory explanations [6, 10]. Google provides a fact-checking service at https://toolbox.google.com/factcheck/explorer, which indexes crawled articles that have meta-data tags indicating that they are fact-checking articles. However, this system only supports search on a topic or a person and does not support (as far as we can see) searching by article content, or by images. The site nokiye.com supports searching of article text across fact-checking sites, while altnews.in provides a mobile app supporting image search on articles in the altnews.in site. A recent tutorial on fake news detection with a database focus, is provided by [2].

5 CONCLUSIONS

We have developed the Kauwa-Kaate system to help users fact-check articles that they have (typically) received via social-media forwards. Our system solves a practical problem, and provides several user-friendly interfaces. We are currently working on features to support human fact-checkers, improved image/video handling, and on determining site and author bias. Fact-checking in the health domain is another important problem, given the large number of health related forwards that give wrong information.

Acknowledgments: Work partially supported by Govt. of India DST Imprint Grant IMP/2018/001682. The WhatsApp interfaces were developed by Aman Jindal and Shiva Kulshreshta. Ashish Mithole and Jagadeesha Kanihal contributed to some aspects of the system.

REFERENCES

[1] IBM corporation. 2018. Natural language processing for advanced text analysis. https://www.ibm.com/watson/services/natural-language-understanding/

[2] Laks V. S. Lakshmanan, Michael Simpson, and Saravanan Thirumuruganathan. 2019. Combating Fake News: A Data Management and Mining Perspective. *PVLDB* 12, 12 (2019), 1990–1993.

[3] Kashyap Popat, Subhabrata Mukherjee, Jannik StrÃűtgen, and Gerhard Weikum. 2018. CredEye: A Credibility Lens for Analyzing and Explaining Misinformation (demo).

[4] Mehdi Samadi, Partha Talukdar, Manuela Veloso, and Manuel Blum. 2016. ClaimEval: Integrated and Flexible Framework for Claim Evaluation Using Credibility of Sources. In *AAAI*. 222–228.

[5] ScrapingHub. 2018. Scrapy: A framework for extracting the data you need from websites. https://scrapy.org/

[6] Kai Shu, Limeng Cui, Suhang Wang, Dongwon Lee, , and Huan Liu. 2019. dEFEND: Explainable Fake News Detection. In *Conference on Knowledge Discovery and Data Mining (KDD)*.

[7] R. Smith. 2007. An Overview of the Tesseract OCR Engine. In *Proceedings of the Ninth International Conference on Document Analysis and Recognition - Volume 02 (ICDAR '07)*. 629–633.

[8] Xuezhi Wang, Cong Yu, Simon Baumgartner, and Flip Korn. 2018. Relevant Document Discovery for Fact-Checking Articles. In *ACM WWW*. 525–533. https://doi.org/10.1145/3184558.3188723

[9] H. C. Wong, M. Bern, and D. Goldberg. 2002. An Image Signature for any Kind of Image. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*.

[10] Fan Yang, Shiva K. Pentyala, Sina Mohseni, Mengnan Du, Hao Yuan, Rhema Linder, Eric D. Ragan, Shuiwang Ji, and Xia (Ben) Hu. 2019. XFake: Explainable Fake News Detector with Visualizations. In *World Wide Web Conference (WWW)*.