CHECK-IT: Real-Time Detection of Fake News

Alexandros Kornilakis

Thesis submitted in partial fulfillment of the requirements for the

Masters' of Science degree in Computer Science and Engineering

University of Crete
School of Sciences and Engineering
Computer Science Department
Voutes University Campus, 700 13 Heraklion, Crete, Greece

Thesis Advisor: Prof. *Evangelos Markatos*

This work has been performed at the University of Crete, School of Sciences and Engineering, Computer Science Department.

The work has been supported by the Foundation for Research and Technology - Hellas (FORTH), Institute of Computer Science (ICS).

University of Crete
Computer Science Department

CHECK-IT: Real Time Detection of Fake News

Thesis submitted by
Alexandros Kornilakis
in partial fulfillment of the requirements for the
Masters' of Science degree in Computer Science

THESIS APPROVAL

Author: Alexandros Kornilakis

Committee approvals:

Evangelos Markatos
Professor, Thesis Supervisor

Polyvios Pratikakis
Assistant Professor, Committee Member

Maria Papadopouli
Professor, Committee Member

Departmental approval:

Antonios Argyros
Professor, Director of Graduate Studies

Heraklion, September 2019

CHECK-IT: Real Time Detection of Fake News

Abstract

Over the past few years, we have been witnessing the rise of misinformation on the Internet. People fall victims of fake news continuously and contribute to their propagation knowingly or inadvertently. The use of propaganda is indeed ancient, but never before has there been the technology to so effectively disseminate it. The social media engagement that has swept our lives over the past decade practically exploded the proliferation of misinformation, including the associated distribution of fake news. The 'pizzagate shooting' incident and the Cambridge Analytica scandal indicate that we should not take this rise of misinformation lightly.

Many recent efforts seek to reduce the damage caused by fake news by identifying them automatically with artificial intelligence techniques, using signals from domain flag-lists, online social networks, etc.

In this thesis, we present Check-It, a system that combines a variety of signals into a pipeline for fake news identification. Such signals include the reputation of the person (account) posting the news, the reputation of the website where the news is hosted, the linguistic features that characterize a fake news article as well as the article's content per se.

Using a deep learning approach, we combine all these features towards providing a rating that is timely and accurate. Check-It is developed as a web browser plugin with the objective of efficient and timely fake news detection while respecting user privacy. The requirements we considered when designing Check-It is GDPR compliant, highly confident identification, low response time and lightweight computation. To implement our plugin, we have used pure JavaScript frameworks, like Minhash.js and TensorFlow.js.

In this thesis, we present the design, implementation, and performance evaluation of Check-It. Experimental results show that it outperforms state-of-the-art methods on commonly used datasets while achieving an accuracy of 93%. Furthermore, we provide some directions that can guide future versions of Check-It.

Περίληψη

Τα τελευταία χρόνια, παρατηρούμε την άνοδο της παραπληροφόρησης στο Διαδίκτυο. Οι χρήστες διαρκώς πέφτουν θύματα ψευδών ειδήσεων και συμβάλλουν στη διάδοση τους, εν γνώσει τους ή μη. Η χρήση προπαγάνδας είναι ένα διόλου πρόσφατο φαινόμενο, αλλά ποτέ πριν δεν υπήρχε η κατάλληλη τεχνολογία για να διαδοθεί τόσο αποτελεσματικά. Η χρήση των κοινωνικών μέσων που έχει λάβει σημαντικές διαστάσεις στη ζωή μας την τελευταία δεκαετία, ώθησε πρακτικά την αύξηση της παραπληροφόρησης, συμπεριλαμβανομένης της σχετικής διάδοσης ψεύτικων ειδήσεων. Οι πυροβολισμοί ως αποτέλεσμα της θεωρίας συνωμοσίας «pizzagate» και το σκάνδαλο Cambridge Analytica, υποδεικνύουν πώς δεν πρέπει να λάβουμε αψήφιστα αυτή την άνοδο της παραπληροφόρησης.

Αρκετές πρόσφατες δουλειές προσπαθούν να περιορίσουν τις συνέπειες από την διάδοση των ψευδών ειδήσεων, εντοπίζοντας τις αυτόματα χρησιμοποιώντας τεχνικές τεχνητής νοημοσύνης, γνωστές λίστες, πληροφορίες από κοινωνικά δίκτυα κλπ.

Στην παρούσα μεταπτυχιακή εργασία παρουσιάζουμε το Check-It , ένα σύστημα που συνδυάζει μια ποικιλία σημάτων με στόχο την ταυτοποίηση των ψευδών ειδήσεων. Τέτοια σήματα περιλαμβάνουν τη φήμη του ατόμου (λογαριασμού) που δημοσιεύει τις ειδήσεις, τη φήμη της ιστοσελίδας όπου φιλοξενούνται τα νέα, τα γλωσσικά στοιχεία που χαρακτηρίζουν ένα ψεύτικο ειδησεογραφικό άρθρο καθώς και το ίδιο το περιεχόμενο του άρθρου.

Χρησιμοποιώντας μια προσέγγιση βαθιάς μάθησης συνδυάζουμε όλα αυτά τα χαρακτηριστικά για τον υπολογισμό μιας έγκαιρης και ακριβούς ένδειξης. Το Check-It έχει αναπτυχθεί ως ένα πλυγιν για προγράμματα περιήγησης ιστού με στόχο την αποτελεσματική και έγκαιρη ανίχνευση ψεύτικων ειδήσεων, ενώ παράλληλα σέβεται το ιδιωτικό απόρρητο των χρηστών. Οι απαιτήσεις που επιβάλαμε για τον σχεδιασμό του Check-It είναι η συμμόρφωση με το GDPR, η έλλειψη ανακριβών απαντήσεων, ο μικρός χρόνος απόκρισης και ο μικρός υπολογιστικός φόρτος. Για να υλοποιήσουμε το πλυγιν μας έχουμε χρησιμοποιήσει βιβλιοθήκες γραμμένες σε *JavaScript*, όπως οι *Minhash.js* και *TensorFlow.js*.

Στην παρούσα εργασία παρουσιάζουμε το σχεδιασμό, την υλοποίηση και την αξιολόγηση της απόδοσης του Check-It . Τα πειραματικά αποτελέσματα πάνω σε γνωστά σύνολα δεδομένων δείχνουν ότι το Check-It συνιστά βελτίωση των σύγχρονων σχετικών μεθόδων, ενώ ταυτόχρονα επιτυγχάνει ακρίβεια της τάξης του 93%. Επιπλέον, ορίζουμε ορισμένες κατευθύνσεις που μπορούν να ακολουθήσουν οι μελλοντικές εκδόσεις του Check-It .

Acknowledgements

I am grateful to my supervisor Prof. Evangelos Markatos for his guidance, for the chance to be a part of the Distributed Computing Systems Lab at FORTH-ICS and for trying to teach me what means to be a researcher. I need to express my appreciation to my colleagues for making the working hours in the DCS lab fun and interesting. Finally, I would like to thank my parents; without their support, I wouldn't be writting these lines.

This work was performed at **Distributed Computing Systems** laboratory, **Institute of Computer Science** (ICS), **Foundation for Research and Technology - Hellas** (FORTH), and is supported by the Google Digital News Initiative grant.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 History and Background

Disinformation is not a recent issue. As Guardian columnist Natalie Nougayrède has observed: "The use of propaganda is ancient, but never before has there been the technology to so effectively disseminate it" [1]. Misinformation, disinformation, and propaganda have been features of human communication since at least the Roman times [2]. However, the invention of the Gutenberg printing press in 1493 dramatically amplified the dissemination of disinformation and misinformation, and it ultimately delivered the first large-scale news hoax - 'The Great Moon Hoax' of 1835[1].

As early as 1925, when news offices started to connect via wire, the authenticity of information became a concern. Editors did not know whether the news coming in through the wire was true or not. They could try to infer authenticity based on the source of the news, but still, the concern remained: is this piece of news that just came over the wire true or not? Although the concern was there, the editors usually managed to find ways to mitigate it and reduce the intentional misinformation to the minimum possible: after all, the amount of news that came over the wire and could potentially be misinformation was not that large. Unfortunately, the "tsunami" of social media engagement that has swept our lives over the past decade practically exploded the proliferation of misinformation including the associated distribution of fake news [3].

Four are the main reasons for this explosion:

- *Speed.* Although twenty years ago, we used to talk about the daily newspaper; we do not talk about daily news anymore. News propagates at the speed of light 24/7, and thus, any mechanism to check their truthfulness needs to operate at the same speed and intensity.

- *Scale.* Social media today have billions of users - orders of magnitude more than any newspaper ever had.

[1] `http://www.tandfonline.com/doi/abs/10.1207/S15327728JMME1502_3`

- *There is no "editor" anymore.* Although newspaper editors used to perform quality control on the news about to be published, social media do not necessarily have ''editors". In several cases, they do not even have professional journalists: it is now other people (influencers, public opinion shapers, or even ordinary people) who tweet, re-tweet, and spread the news to their constituency. It is not clear that all people who spread information on social media have the same training and ethical standards of professional journalists.

- *The medium has changed.* The distributed structure of the Internet and its associated ecosystem of applications, including web sites, social media, smart apps, and even peer-to-peer systems, has created new channels for information propagation. Indeed, today most people (as many as 62% of them) receive their news from social media sites. This implies that traditional and trusted sources of information have started to lose their market foothold, which is eagerly acquired by new providers of information who have not yet established their trustworthiness.

We should not treat that explosion of misinformation lightly. In the days immediately before and after the US election, "people shared nearly as much 'fake news' as real news on Twitter"[4]. One particular 'fake news' story circulating the time of the election outlined a supposed child abuse-ring allegedly led by Hillary Clinton, running out of a pizza restaurant called Comet Ping Pong. It led one man to 'self-investigate' by firing an assault rifle inside the restaurant [5]. A week later, a YouGov poll found that the young man was not alone in believing it; nearly half of Trump's voters polled gave some credence to the rumors. By coincidence, two weeks earlier, BuzzFeed's Craig Silverman had published an article that launched the term 'Fake News' [6].

As a followup to the 2016 elections, in March 2018, a whistleblower revealed to news agencies that a Cambridge University psychology academic and 'Cambridge Analytica', has exploited a massive dataset drawn from millions of Facebook users. The company used the data to target specific sets of voters in the lead up to the USA's 2016 Presidential Election. According to undercover reporting by Channel 4, company executives boasted of using their data to target audiences with propaganda and misinformation [7].

1.2 The Fake News Ecosystem

Before proposing an approach for dealing with the Fake News problem, we should try to define it. Someone may consider fake news as false information. Yet, this viewpoint may not be precise[2]. As so, we are presenting a taxonomy proposed by Zannettou et al. [8]. According to the literature, there are 8 types of false information:

[2] `https://guides.lib.umich.edu/fakenews`

- *Fabricated.* Completely fictional stories disconnected from real facts.
- *Propaganda.* A special instance of the fabricated stories that aim to harm the interests of a particular party and usually have a political context.
- *Conspiracy Theories.* Stories that try to explain a situation or an event by invoking a conspiracy without proof.
- *Hoaxes.* News stories that contain facts that are false or inaccurate and are presented as legitimate facts.
- *Biased or one-sided.* Stories that are extremely one-sided or biased. In the political context, this type is known as Hyperpartisan news and are stories that are extremely biased towards a person/party/situation/event.
- *Rumors.* Stories whose truthfulness is ambiguous or never confirmed.
- *Clickbait.* Refers to the deliberate use of misleading headlines and thumbnails of content on the Web.
- *Satire News.* Stories that contain a lot of irony and humor.

At this thesis, we are mainly interested in fabricated stories and propaganda. Following that path, we are defining fake news as fabricated stories intending to deceive and harm. Although our approach does not ignore different types of false information, fabricated information is the most severe type.

The motivation behind the creation and spread of fake news content may vary. Trend Micro curently sees three major motivations behind fake news [9]: political, financial gain, and character assassination. More analytically, political propaganda is designed to get people to change their minds about their political beliefs or some other opinion. The most obvious financial motivation could be advertising, while character assassination by fake news could target politicians or even private individuals to cause harm.

In their study, Zannettou et al. also describe the different actors that make up the false information propagation ecosystem. Fake news is created and spread by bots, criminal organizations, activists, governments, journalists, trolls, and others.

We now know from related studies, that false information spreads faster than real information [10]. These studies point to the human predisposition in being attracted by novelty - it is known that false news carries more novelty - to explain this. It is not bots that usually spread the misinformation; This is mainly done by humans. Yet the technological processes - social media, algorithmic news curation, bots, artificial intelligence, and big data analysis - are creating echo chambers that reinforce our biases, remove incidia of trustworthiness, and are overwhelming our capacity to make sense the world.

The biggest threat of misinformation is the one that poses to our democracy. Echo chambers ringing with false news can make democracies ungovernable. We

can imagine a pluralist democracy in which populations contested elections, without ever sharing a viewpoint on what is going on in the world. Whoever won would design policies to counter what they saw as the major policy question of our times. Since these viewpoints would be isolated and different, such pluralist democracy would be deeply unstable [11].

1.3 Approach and Contribution

The focus of our work is the detection of news content that is fabricated and can be verified to be false. In this thesis, we present a plugin that fights disinformation using an automated approach. Our approach is inspired by the way we fight SPAM email messages. Indeed, to fight SPAM, computer scientists have developed SPAM filters: automated programs that scan all email messages of each user, categorize them as SPAM (trash email) or HAM (regular email) and filter the SPAM out of the user's mailboxes. As it is true with SPAM detection systems, our system has the requirement of achieving a low false-positive rate.

In this thesis, we follow the same approach: we process all information (e.g. tweets, posts, web documents, etc.) that users see online and characterize them as misinformation or not. If we find misinformation we clearly label it so that the user will be warned that he should be careful before believing this current piece of news. For experimental studies, we have developed our system as a plugin for the popular web browsers, namely Google's Chrome and Mozilla's Firefox. However, our method is general and applicable to any browser.

A key difficulty in our approach is to combine effectively a variety of signals to decide whether a piece of news is misinformation. Such signals include the reputation of the person (account) posting the news, the reputation of the web site where the news is hosted, the linguistic features that characterize a fake news article as well as the article's content per se. Using a deep learning approach, we combine all these features towards providing a rating that is timely and accurate. Another key aspect of our system is that it protects the privacy of the user (GDPR compliant) since the plugin works locally on the user's browser without the need for external communication.

We empirically evaluate our proposed method via extensive experiments on real-world datasets, demonstrating that our approach significantly improves the performance on detecting and reducing the spread of fake news and misinformation on the Web. To evaluate our approach, we have trained our model with the Fake News Corpus which includes 3 million articles labeled as fake and real. To the best of our knowledge, this is the biggest corpus in the research community.

1.4 State of the art and its Limitations

Due to the increasing interest in analyzing fake news in the Web and the development of tools to deal with fake news that had been previously identified, there is

not a satisfactory amount work in automatic fake news detection tools. Currently people do not have the tools they need in order to filter out information they are not interested in. For example, if their friends share fake news from time to time, they do not have any way to tell the social media platform "I do not want the fake news my friends (probably) inadvertently propagate. Can you filter the fake news (not my friends!) out of my social feed? Or better yet, can you label the fake news as such? I will then do the filtering out.". The main problem stems from the fact that it is difficult to develop classification algorithms to capture fake news.

Researchers in [12] studied the feasibility of using a crowdsourcing platform to identify rumours and fake news in social media. According to their research outcomes, the annotators achieve high inter-annotator agreement. In [13], authors found that fake news posts in social media are usually provoking posts (i.e., tweets) from users who raise questions about these posts. In this direction, another approach that has been proposed is the development of browser plugins, such as the B.S. Detector[3] and the FakerFact[4], which flag content from fake news sources using a constantly-updated list of known fake news sites as a reference point.

1.5 Roadmap

The rest of this thesis is organized as follows. In Section 2 we describe related work from the literature. In Section 3 we describe our approach. In Section 4 we present the functionality and UI of Check-It browser extension. In section 5 we describe our experimental setup and detail the performance of our approach. In Section 6 we discuss the limitations of this work and a present a roadmap for future work. In Section 7 we conclude this thesis.

[3] `http://bsdetecor.tech`
[4] `https://www.fakerfact.org/`

Chapter 2

Related Work

The task of fake news detection is similar to various other interesting challenges ranging from SPAM detection to rumor detection [12]. In recent years, researchers are seeking to better define and characterize misinformation and its place in the larger information ecosystem [14]. An important aspect of characterizing misinformation is to understand how people perceive the credibility of the information. People usually tend to believe the news that confirms what they already know, or what they already believe to be true [15]. News that goes contrary to their beliefs (no matter how true the news is), maybe met with high degrees of resistance. Thus, presenting people with the facts does not necessarily change their minds - several people keep on believing the fake news. To make matters worse, repeating the fake news, even in the context of refuting them, just makes them stronger. Thus, it seems that we need to explore non-obvious approaches to fight misinformation [16].

Nowadays, several pieces of fake news can be easily labeled as such. Once the news is labeled as "fake" or "most likely to be fake", people will probably be reluctant to share them further. The embarrassment of sharing fake news will deter a significant percentage of people from engaging into active sharing of such misinformation: it is just like forwarding SPAM email messages - most people would not forward SPAM. In this direction, Facebook is already partnering with fact-checking organizations. Facebook users can flag articles they suspect contain false information. These articles are then handed over to an independent evaluation centre.

When a false story is identified, rather than being removed, it is tagged with a warning that it contains fake news and appears lower down in users feeds. Recently, Facebook will provide to social scientists unprecedented access to its data so that they can investigate how the spread of fake news on social media influences elections[1]. Another initiative aiming to help citizens make informed choices ahead of the 2017 French election is the First Draft News project CrossCheck, a collaborative verification programme involving technology firms including Facebook and

[1] https://www.nature.com/articles/d41586-019-01447-5

Google. The project sees journalists from across France working together to find and verify online content, including photos, videos, memes, comment threads, and news sites. Similarly, Washington Post asked its readers to use the term "Fake News" to report this news. However, this term was used maliciously and it ended up being not so successful. Besides, some effort has also been done to detect fake news, including approaches that apply text-based methods[17] and fact-checking through knowledge graphs [18].

In this context, Google has recently released the Perspective API which is an application interface currently focused on moderating online conversations using machine learning to spot abusive, harassing, and toxic comments. Facebook trained a machine learning algorithm by having humans identify common phrases in old headlines of fake news. However, the current fact-checkers and crowdsourcing initiatives have limitations since they cannot cope with the large volume of misinformation generated online, and are usually disconnected from the Web browser, which is the medium used from users to read and share misinformation.

A few early studies tried to detect fake news based on linguistic features extracted from the text of news stories [19],[20],[21]. Recent studies have also shown that social networking features play a very important role in detecting fake news [15]. Deep neural networks have been successfully applied to fake news detection [14],[19],[20]. Technical details regarding these approaches are presented in the evaluation section. However, all the existing approaches are trying to solve the problem using only one signal of information (i.e. fact-checking web sites, linguistic features, social networking features). Most current studies on misinformation either focus on analyzing the influence of the topology of the social network on the consumption and sharing of misinformation or taking into account the linguistic characteristics. Also, most systems tend to focus on the technical and not on the human aspects of the problem (i.e., the motivations of the users when generating and spreading misinformation). Our model is inspired by SPAM detection research. Our system will assemble all sources of signal and will combine them into one signal score. The score will reflect how confident we are that the story is fake (or not) and explore relationships among news comments$'$ topicality, temporality, sentiment, virality, and quality.

There exist multiple fake news detection systems, many implemented in the form of a browser plugin. Each available plugin utilizes either flag-lists of fake news domains, fact-checking sites, or artificial intelligence models for the identification of fake news articles. Their in-browser functionality is mostly enabled via a RESTful API which the plugin invokes every time the user visits a questionable site or browsers her social feeds. Check-It differs from these common fake news detection plugins by utilizing, not a single signal (either flag-lists, fact-checking sites or artificial intelligence), but a combination of the aforementioned signals, thus maximizing its fake news identification accuracy. The main strengths of our work are the ability for real-time detection (due to lightweight computational methods), respect to the user's privacy and the low false-positive rate.

It is worth mentioning some web-based tools that aim to detect misinformation. **InVID** [22] is a browser plug-in that aims to detect user-generated fake video. **REVEAL** [23] is a Web-based service that tries to detect forged (fake) images. **TweetCred** [24] is a Web-based System for assessing the credibility of the content. **Fake Tweet Buster** [25] is a Web application that identifies tweets with fake images and users who are consistently uploading and/or promoting fake information on Twitter. **Claimbuster** [26] is an end-to-end system that uses machine learning, natural language processing, and database query techniques to aid fact-checking. Finally, **Hoaxy** [27] is a platform for the collection, detection, and analysis of online misinformation and it's related fact-checking efforts.

Chapter 3

Check-It System

Check-It satisfies a series of user-centric functional requirements revolving around the user's data privacy, as listed below: The following requirements must be taken into consideration when designing a Web browser plugin for detecting fake news on the Web:

- **Preserve User Privacy**: Check-It plugin should work locally, on the user's web browser, without the need for external communication (i.e. a RESTful API).
- **Highly Confident Identification**: Check-It labels a piece of news as fake if it is highly confident about it.
- **Low Response Time**: All the required resources, such as the flag-list and linguistic model, are efficiently loaded in the user's web browser. Also, the interconnected components of the plugin have been developed to have a low response time.
- **Lightweight Computation**: Asynchronous processing and parallelization are taken place to minimize the load of the plugin.

Thus, our main objective is: *to provide a Web browser plugin that detects efficiently and timely the fake news articles respecting the user's privacy.*

As depicted in Figure 3.1, Check-It system consists of four main components that function as a pipeline for fake news identification on the Web. The Flag-list Matcher component matches domains of news articles to Known Fake News Domains and Fact Checks; the Fact-Check Similarity component compares a piece of news against Known Fact Checked Articles labeled as fake from Fact-Checking organizations, such as Politifact[1] and Snopes[2]; the Online Social Network User Analysis component is responsible for analyzing user behavior in social networks and producing a User-Blacklist of fake news propagators; and lastly, the Linguistic

[1] `https://www.politifact.com/`
[2] `https://www.snopes.com/`

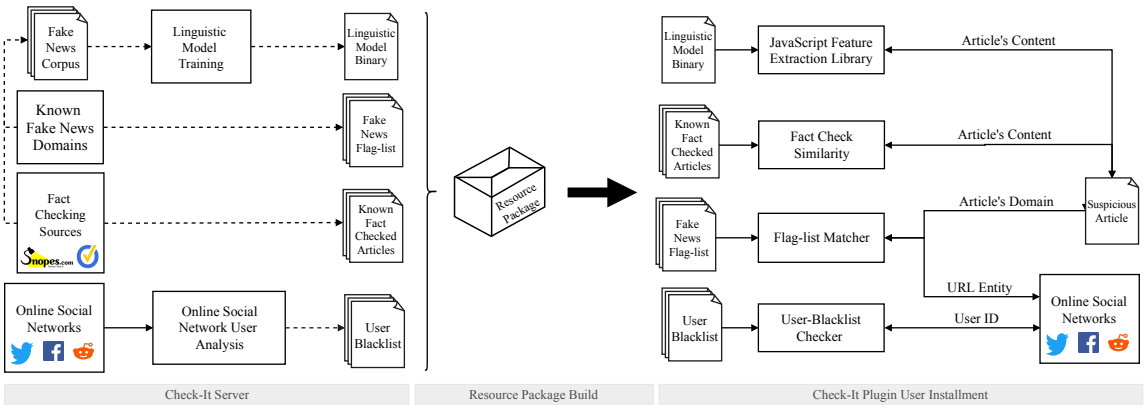

Figure 3.1: Architectural diagram for the Check-It System.

Model component, an artificial intelligence model, has been trained on linguistic features, of the Fake News Corpus, for the detection of fake news articles.

Check-It preserves the user's privacy, whilst providing the appropriate functionality and performance, by loading the required resources locally, on the user's device. These resources are combined in a Resource Package, which includes the Fake News Flag-lists, the Known Fact-Checked Articles, the User-Blacklist, and the binary-produced Linguistic Model. The Resource Package is available by the Check-It Server. The only communication between the Check-It Server and the user is during the installation of the plugin, where the required resources are downloaded and installed on the user's end (user-blacklist, fake news flag-list, known fact-checked articles, and linguistic model), and any critical updates on those resources. This provides a guarantee that Check-It is GDPR compliant.

At the Check-It Plugin User Installment, the resources are loaded within the plugin and assigned to their respective components. Besides the Fact-Check Similarity, Flag-list Matcher and User-Blacklist Checker, the Linguistic Model requires the features from the article to be extracted. To this end, the JavaScript Feature Extraction Library was developed, responsible for capturing the required features from within the article and using them as input to the Linguistic Model Binary.

The resource package is available in timely builds by the Check-It server, responsible for all the heavy lifting, including the update of the Fake News Corpus and the generation the training and exporting of the linguistic model; for the continuous operation of the Online Social Network User Analysis component and generation of the user-blacklist; and for the update of the Fake News Flag-lists, used by the Flag-list Matcher component; and the known Fact-Checked Articles for the Fact-Check Similarity component. After the resource package is built, it is then transferred to the local installment of the Check-It plugin to the user's web browser. The only communication between the server and the user is during the installation of the plugin where the required resources are downloaded

and installed on the user's end (user-blacklist, fake news flag-list, the known fact-checked articles, and linguistic model), and any critical updates on those resources. Proceeding, we present the four components in detail.

3.1 Flag-list Matcher

Some domain names are well known for spreading misinformation. Whether they do this on purpose, or for fun (like satire), the information they provide is frequently not accurate and they should not be used as trusted sources of news. Currently, there are several lists (which we call them *flag-lists*) which contain domain names known for containing and spreading misinformation. These lists are established and maintained by researchers or volunteers whose aim is to warn Internet users by "flagging" information sources of dubious credibility. The "flagging" provides some explanation for why a domain name is included in the flag-list. For example, the flag may be "fake news," which means that the site spreads misinformation, or "biased," which means that the site is known to promote a biased point of view. A non-exhaustive list of these flag-lists includes Kaggle[3], OpenSources[4] and Greek-Hoaxes[5]. A complete list of the flag-lists used by Check-It is at A.3.

Our flag-list contains domain names of websites written in English, Italian, and Greek. We have to note that our main source, opensources.co website appears to be down during the last months. Although this list currently is not updated, it is still used by popular tools like BSDetector. So, it maintains its credibility. Someone might argue that, if a list is not regularly updated, may contain false alarms - blacklisted domains that are now credible. We consider this scenario as not possible, since in the case that a domain is flagged in public lists, it permanently loses its credibility, and it is more possible to move to a different domain if the owner or the site orientation changes.

Our system has been designed to be easily configurable concerning the flag lists it takes into account. URL flag-lists and domain name checking is the simplest way for an initial, fast assessment of the trustworthiness of a news article. Unfortunately, flag-lists do not test the truthfulness of the article itself: they just comment on the reputation of the website publishing the article. In that respect, flag-lists can be very helpful as long as they identify sites that consistently engage in disinformation campaigns or in propaganda spreading, in which case they can easily flag articles hosted on dubious sites. Nevertheless, one might want to be able to reason about the credibility of articles hosted in dubious web sites. To further assess the validity of such articles we use (i) fact-checking web sites (section 3.2) and (ii) machine learning approaches (sections 3.3 and 3.4), as we describe below.

[3] `https://www.kaggle.com/mrisdal/fake-news`

[4] `https://raw.githubusercontent.com/BigMcLargeHuge/opensources`

[5] `https://raw.githubusercontent.com/Ellinika-Hoaxes`

3.2 Fact Check Similarity

Several Fact-Checking organizations are dedicated to combating propaganda, misinformation, and hoaxes circulating on the Internet. They typically employ professional journalists who invest the time to research and comment on the truthfulness of articles shared on the Web and online social media [28]. Once the truthfulness of an article is established, the findings are publicized, along with the associated information. Check-It capitalizes on fact-checking web sites, by cross-checking every article processed by its plugin against a list of fact-checking web sites, generating an informative warning when an article happens to be found listed on these web sites. Fact-checking is known as the act of checking factual assertions in a non-fictional text to determine the veracity and correctness of the factual statements in the text. Usually fact-checking is done after the text has been published and disseminated (post hoc). Post hoc fact-checking is most often followed by a written report of inaccuracies, sometimes with a visual metric from the checking organization.

Check-It capitalizes on fact-checking web sites, by cross-checking every article processed by its plugin against a list of fact-checking web sites, generating an informative warning when an article happens to be found listed on these web sites. Our list includes fact-checking sites like factcheck[6] or snopes[7]. A complete list of the fact-checking sources Check-It includes is at Table A.2.

To cross-check against the list of fact-checked articles, we used document similarity techniques. Document similarity is a metric defined over a set of documents, where the distance between them is based on the likeness of their meaning or semantic content. There are many techniques for comparing 2 different documents, many of them used by search engines. Such methods are the tf-idf[8] model, latent semantic analysis, word2vec, doc2vec, and others. When comparing documents, someone may use plagiarism detection algorithms[29]. Winnowing[30] is a plagiarism detection algorithm we have studied during our research. Using a hash function, winnowing generates several fingerprints for a document. That fingerprint is cross-checked against a corpus of fingerprints to detect a match. Another related family of techniques is fuzzy hashing. This family includes algorithms like ssdeep[31], sdhash, mvHash and others[32]. Yet, fuzzy hashing techniques are used for malware analysis rather than Web document similarity.

To meet the "Low Response Time" requirement and keep a low memory footprint, we have used Locality Sensitive Hashing techniques [33]. Locality-Sensitive Hashing (LSH) is an algorithm for solving the approximate or exact Near Neighbor Search in high-dimensional spaces. LSH is a family that contains algorithms like simhash[34], minhash[35], TLSH [36], nilsimsa [37] and others. We have limited our algorithmic choice to minhash and simhash algorithms since the latter suffers from a larger number of false positives. After that, minhash and simhash have

[6] `https://www.factcheck.org/`

[7] `https://www.snopes.com/`

[8] `http://www.tfidf.com`

been tested before in the Web document similarity domain.

Now, we had to choose the fittest candidate. So, we performed a simple experiment. We gathered a set of the 1000 most common English words[9]. Then, we generated a document containing all these words and documents containing a subset of them. Fig. 3.2 depicts the similarity score as a function of the number of common words for simhash and minhash algorithms. We have used word-grams of length 6 as features. Our observation is that simhash is a highly nonlinear function and thus offers a coarser granularity. Our experiment points at minhash since it is easier to balance the trade-off between false positives and detection rate. To find the right threshold, we performed an extra experiment. We evaluated minhash over our corpus of fact-checked articles. We chose the threshold that results in the largest detection rate while keeping a false positive rate close to 0. The threshold was at 10%.

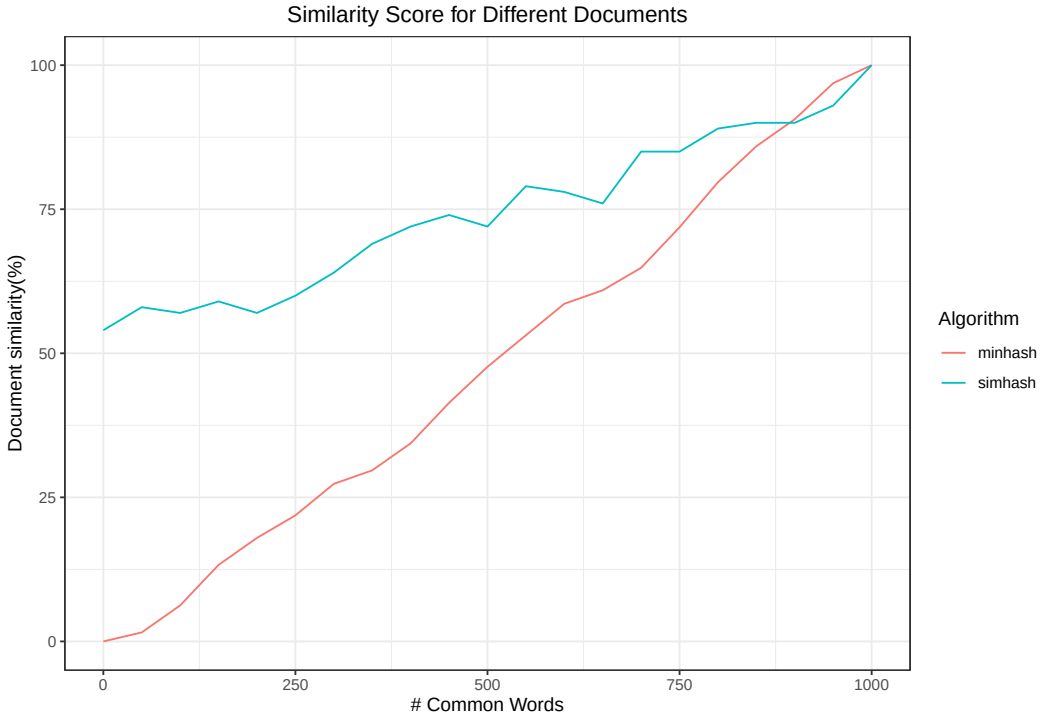

Figure 3.2: Distribution of Similarity Values for simhash and minhash

[9] https://www.ef.com/wwen/english-resources/english-vocabulary/top-1000-words/

3.3 Online Social Network User Analysis

Although perpetrators generate false content intending to harm, Online Social Networks (OSNs) provide the means for spreading it. Recent studies [16] have showed that online social networking platforms (OSN), like Twitter and Facebook, have become mechanisms for massive disinformation campaigns. Since OSNs play an important role in the propagation of fake news [16], we have incorporated another signal in the Check-It toolkit. The idea behind the OSN signal is to provide a *dynamic user-blacklist*, matching user IDs with a falsity score, indicating the likelihood of a user to post fake news articles.

The user-blacklist is dynamically generated by continuously processing OSN data and applying a DeGroot-based user probabilistic model [38] for the user falsity score calculation. DeGroot model is used since it introduces a simple mechanism of opinion propagation: every individual forms her opinion by averaging her own opinion with those of her friends. The process is repeated until all opinions converge. Although the mechanism is simple, it models sufficiently opinion diffusion and incorporates elaborate characteristics of the process [38]. Figure 3.3 presents the overall pipeline of the module and its components, which we describe in the next paragraphs.

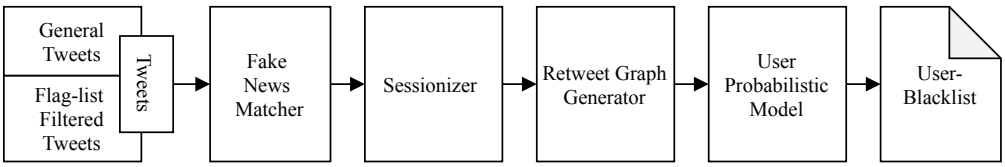

Figure 3.3: Architectural diagram for the social network signal.

The system design of Check-It facilitates integration with multiple OSN platforms. Currently, our OSN user analysis component only supports Twitter. This is due to it's massive popularity and the ease-of-access to its data stream via the Twitter Streaming API[10]. In particular, our system consumes tweets from two sources: a) tweets from the general public and b) tweets containing URLs of known fake news domains. The output of the system is a *User-Blacklist* of fake news propagators.

The Flag-list Matcher component is responsible to mark tweets that contain a URL entity and positively answer the following question: *Does the URL originate from a suspicious domain?* The tweets that have not been marked by the Flag-list Matcher are ordered in a timely manner and processed by the session-based model in groups of 1-hour sessions (Sessionizer task). A similar approach has been used in [15]. Then, each session is assigned to the Retweet Graph Generator, which is responsible for the creation of the retweet graph of the session. A retweet

[10] `https://developer.twitter.com/en/docs/tweets/filter-realtime/overview`

graph $G = (V, E)$ consists of nodes $u, v \in V$ depicting users and edges $(u, v) \in E$ representing the retweet action between users u and v. After the generation of the retweet graph, the User Probabilistic Model is applied in order to calculate the falsity score per user and produce the User-Blacklist. Initially, each user u_i is assigned with a falsity score of $p_i^{(0)} = 0$. Next, we briefly present the user probabilistic model that is based on the DeGroot's Learning Model.

Let A be the adjacency matrix of the retweet graph G. We have that $A(u,v)=1$ if u retweeted v. We create a transition matrix T by inverting the edges in A (as the influence flows from the retweeted user to the user who retweeted him or her), adding a self-loop to each of the nodes and then normalizing each row in A so it sums to 1 (meaning that each user is equally influenced by every user he or she retweets). Matrix T includes the weight a node adds on another based on the sharing of fake news articles. We then associate a falsity score $p_i^{(0)} = 1$ to every user who posted a suspicious tweet and $p_i^{(0)} = 0$ to all who did not. Lastly, we create new scores $p(t)$ using the updating rule $p(t) = T \cdot p(t-1)$. In summary, the falsity score of a user increases if that user posts or retweets a suspicious tweet (a tweet that contains a URL from the flag-list).

3.4 Linguistic Model

The previously mentioned components, namely flag-list, similarity, and social network, all share a common property; they all focus on meta-information of the news articles. In contrast, the linguistic component analyzes the actual content of the news article. Check-It extracts from an article's headline and body specific linguistic features, which have been widely used to detect fake news [39, 40, 41]. These features are used as input to a Deep Neural Network (DNN), which has been trained to predict the article's veracity. Rather than traditional machine learning, the deep learning approach was used due to the performance amplification it can achieve in the detection of fake news [14, 42] as well as in other problems addressed with artificial intelligence techniques. Next, we present an overview of the article dataset, the different linguistic features, and the DNN model.

3.4.1 Dataset Overview

Online news articles can be collected from different sources, such as news agency homepages, search engines, and social media websites. However, the manual determination of the veracity of news is a challenging task, usually requiring annotators with domain expertise. Check-It makes use of Fake News Corpus[11], an open source dataset composed of **9 million news articles**. These articles originate from a curated list of 1001 domains collected from opensources.co. The entries are divided into 12 groups: *fake news*, *satire*, *extreme bias*, *conspiracy theory*, *rumor mill*, *state news*, *junk science*, *hate news*, *clickbait*, *political*, and *credible*. In the

[11] `https://github.com/several27/FakeNewsCorpus`

scope of Check-It, we focus solely on the *fake news* and *credible* categories of the dataset, consisting of 1 million and 2 million articles respectively. As the dataset describes, fake news is considered when originating from "*sources that entirely fabricate information, disseminate deceptive content, or grossly distort actual news reports*", whereas credible are "*sources that circulate news and information in a manner consistent with traditional and ethical practices in journalism*".

3.4.2 Linguistic Features

Fake news detection on traditional news media mainly relies on news content, such as the headline and the body of an article. We compute different linguistic features that can be found in the headline and body of articles, in order to extract discriminative characteristics for the detection of fake news. These features are extracted and fed to the DNN model via the JavaScript Feature Extraction Library at Check-It plugin User Installment (Figure 3.1. We group these features into 3 broad categories: *stylistic*, *complexity* and *psychological*.

Stylistic Features: These are based on natural language processing to understand the syntax and text style of each article body and headline. Text style features include the frequency of stop-words, punctuation, quotes, negations and words that appear in all capital letters, whereas syntactical features include the frequency of Part-of-Speech tags in the text.

Complexity Features: These are based on deeper natural language processing computations aiming at capturing the overall intricacy of an article or headline. This intricacy can be computed based on several word-level metrics that include readability indexes and vocabulary richness. Specifically, we compute the Gunning Fog, SMOG Grade, and Flesh-Kincaid grade level readability indexes. Each measure computes a grade level reading score based on the number of complex words (e.g. over 3 syllables). A higher index means a document takes a higher education level to read. Moreover, we compute the Type-Token Ratio, which can be defined as the number of unique words divided by the total number of words in the article. In order to capture the vocabulary richness of the content, we also compute the number of hapax legomenon and dis legomenon, which correspond to phrase that occurs only once and twice within a context.

Psychological Features: The psychological features are based on the count of words found in expert dictionaries that are associated with different psychological processes. These dictionaries include the negative and positive opinion lexicon [43], and the moral foundation dictionary [44]. The sentiment score is computed via the AFINN sentiment lexicon [45], a list of English terms manually rated for valence. The AFINN sentiment score is defined as an integer number between -5 and +5, indicating the negative and positive score respectively.

A list of the extracted features is included at Table A.1.

3.4.3 Feature Selection

The stylistic, complexity and psychological features are extracted from both the headline and body of the articles in the dataset, summing in 534 features. Such a large number of features results in an extensive model and deem the local execution as inadequate. In addition, unnecessary features can have side-effects during the model's training, decreasing training speed, model's interpretability, and generalization performance. In order to mitigate these issues, we proceed with a feature selection process to capture the 20 most descriptive features that facilitate the classification of news articles into fake or reliable. Below, we describe the feature selection process that is applied:

1. **Missing Values**: Remove features with a high percentage of missing values e.g. 60%. Such features are not useful for the classification tasks as they do not carry any information, and can also affect the performance of the model.

2. **Single Unique Values**: Remove features with a single unique value, which have zero variance and have no contribution to the training of the model.

3. **Collinear Features**: Remove highly correlated features, which may lead to decreased generalization performance on the test set due to high variance and less model interpretability. These features are selected based on a specified correlation coefficient value (i.e., Pearson correlation coefficient).

4. **Zero Importance**: Calculate the importance of the remaining features according to a gradient boosting decision tree model, and remove features with zero importance.

5. **Low Importance**: This step builds on the feature importance calculated in step (4), and its task is to remove features with low importance as they do not contribute to the total predefined importance. Principal Components Analysis (PCA) is used, keeping only the required principal components so as to retain a certain percentage of the variance (i.e, 95%). The percentage of total importance scores accounted for is based on the same idea.

The above feature selection process resulted in removing 134 features. From the remainder, the 20 most important were selected based on their importance scores, as extracted from step (4) (Table 3.1). These include the average number of stopwords in a sentence, the ratio of uppercase letters in the headline and the AFINN sentiment score.

3.4.4 Deep Neural Network Model

Similar to the linguistic feature selection, the proposed DNN model is compliant to the functional requirements set at the beginning of the project. It is a prerequisite that the model is compatible with conventional user devices and modern

No.	Feature	Score	Type
1	Total number of lines	0.0693	Body
2	Avg. number of stop-words per sentence	0.0185	Body
3	Ratio of uppercase letters	0.0177	Headline
4	Ratio of uppercase letters	0.0152	Body
5	Avg. number of uppercase words per sentence	0.0142	Headline
6	Avg. number of characters per word	0.0141	Body
7	Ratio of alphabetic letters	0.0139	Headline
8	Number of proper nouns (NP)	0.0128	Body
9	Avg. number of sentences beginning with lowercase letter	0.0126	Body
10	Avg. AFINN sentiment score	0.0123	Body
11	Total number of characters	0.0122	Headline
12	Ratio of digits	0.0122	Body
13	Avg. number of sentences beginning with uppercase letter	0.0122	Body
14	Ratio of alphabetic letters	0.0119	Body
15	Number of genitive markers (POS)	0.0116	Body
16	Number of colon or ellipsis	0.0116	Headline
17	Total number of words beginning with uppercase letter	0.0113	Body
18	Number of colon or ellipsis	0.0102	Body
19	Avg. number of characters per word	0.0096	Headline
20	Avg. number of stop-words per sentence	0.0094	Headline

Table 3.1: Table with the 20 most important features as resulted from the feature selection process.

web browsers, as it is available as a traditional web browser plugin. Additional requirements are the low response time, lightweight computations and high confidence for the output. In order to address these challenges, the proposed DNN model adopts the cone-like structure, referred to as the bottleneck principle, and is known to perform well with numerical features [46, 47]. The structure of the model is depicted in Figure 3.4.

Before feeding the data into the DNN model, any categorical data are transformed into numerical, either via discretization or one-hot encoding, depending on the particulars of the input. As a result, each data entry is represented as a vector of numerical features. After the pre-processing, the data is used as input to the DNN model via the model's input layer.

The next layer is a Batch Normalization Layer [48] which is responsible for the normalization of the activations of the previous layer (input layer) at each

batch. Neural networks work better when the input data have zero mean and unit variance, as this enables faster learning and higher overall accuracy. A Batch Normalization Layer can achieve this by transforming and maintaining the mean and variance of its input close to zero. Next, the normalized output enters a set of fully connected layers (dense layers) that form the bottleneck. Such a bottleneck has been shown to result in automatic construction of high-level features. In our implementation, we experimented with multiple architectures, settling in a sequence of 5 layers that consist of 512, 256, 128, 64 and 32 neurons respectively. The final sequence is the one that provided the best results in our task. The units of the network are activated using the hyperbolic tangent activation function (tanh) since it is a better fit when working with standardized numerical data.

Finally, in the DNN model's classification layer, one neuron per class is used with the softmax activation function to produce the probability pair of P_{real} and P_{fake}, which correspond to the probability of the article being real or fake respectively.

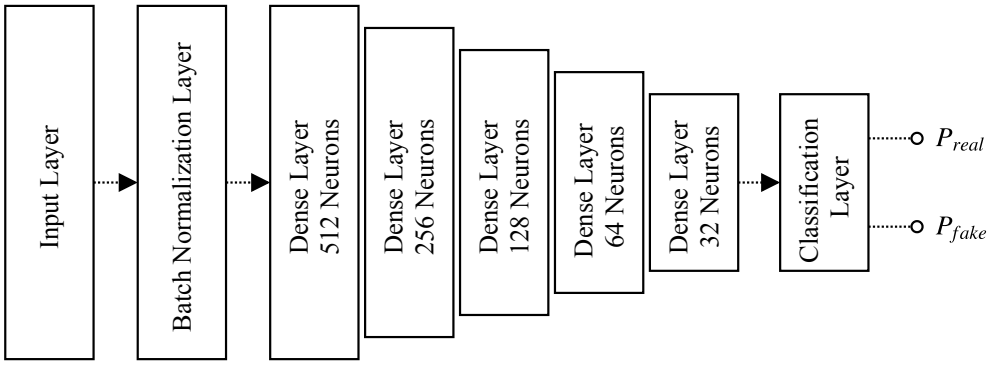

Figure 3.4: Architectural diagram for the deep neural network model used in the linguistic component.

3.5 Check-It Design Decisions and Challenges

Before presenting our implementation, we would like to share our experience with the design of Check-It , and the challenges around this implementation. At first, our decision to do all the processing at the client-side majorly influenced the implementation of the system. It forced us to optimize the execution speed. It also reinforced our decision to break the system in different components and affected the sequence of these components. This sequence, however, was also influenced by additional factors: the false positive rate and the strength of the signal.

In the scenario where the user visits a webpage, the first component we call is the flag-list matcher. The flag-list is stored as a JavaScript object (internally can be a hash table) and the lookup time is O(1)[12]. The flag-list matcher is the component with 0 false positives and false negatives and the fastest execution. If this component does not indicate that the domain publishes fake news, the "fact-check similarity" component is called. The minhashes calculated on the Check-It server are stored in a Locality Sensitive Index[49]. LSH index can achieve sub-linear time complexity, while the worst case is still O(N). After the fine-tuning of minhash, the false positive rate is close to zero.

The flag-list matcher is assigned a higher priority over the rest of the components, mainly because it constitutes a stronger signal (it flags the domain, and not only a specific article) and has a better execution speed. The lowest priority is assigned to the Linguistic Model since it achieves lower accuracy and has a somewhat slower speed due to the feature extraction process. However, we have used the faster JavaScript NLP libraries we could find to reduce this distance.

We understand that if our plugin was slow and disrupted the user's browsing experience, the users may not use it at all. Luckily, we have optimized our plugin's execution, and the imposed latency is negligible.

We note that performing linguistic analysis with client-side JavaScript is a challenge itself. We have encountered two main issues:

- **The quality of NLP libraries**. We have noticed that the results of the linguistic analysis contained inaccuracies and differed from the more stable popular python modules. To overcome this, we had to modify the library code and fix some buggy parts.

- **The extraction of article content**. To perform more precise analysis, we had to extract the actual content of the article the user is reading. We have tested several content extractors, like dom-distiller[13], Just-Read[14], unfluff [15] and others. However, we have decided to use Mozilla Readability[16] library. It is fast, fully open-source and actively maintained. However, the result is

[12] `https://v8.dev/docs`
[13] `https://chromium.googlesource.com/chromium/dom-distiller`
[14] `https://github.com/ZachSaucier/Just-Read/`
[15] `https://github.com/ageitgey/node-unfluff`
[16] `https://github.com/Mozilla/readability`

not always optimal. We had to tweak thresholds of the internal heuristics, to reduce the noise that is returned.

This Linguistic Model component is not the only component affected by the content extraction process. We have also used the Mozilla Readability library at the server-side, to create the dataset for the Fact-Check Similarity component. To verify the data quality, we had to perform some manual cleaning.

Finally, we have to note that the multi-modal architecture of Check-It has an additional benefit: The architecture is very extendable. Check-It is designed so that it trivial to register a new component and assign a priority to that. For example, a component that detects fake images and has to be called after the fact-check similarity module will automatically be given the page HTML as input.

Chapter 4

Check-It Browser Plugin

The system has been implemented as a plugin for the Google Chrome and Mozilla Firefox Web Browsers.

4.0.1 Example operation

Once loaded, the plugin appears as a small blue book in the upper right corner (Fig. 4.1). The plug in works in two ways:

- *Browsing Mode*: the plugin checks the URL that the user is accessing
- *Social Media Mode*: the plugin digs inside the page to find the URLs that are mentioned in the tweets (or other posts).

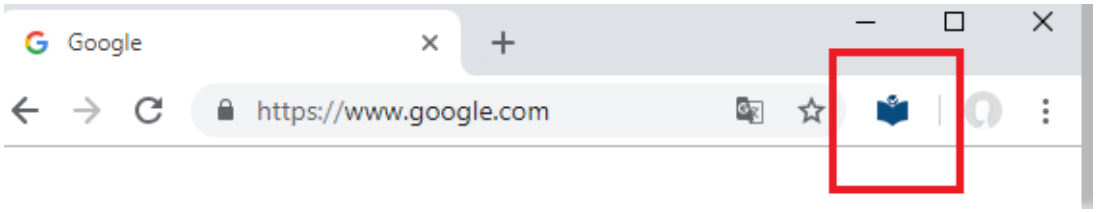

Figure 4.1: The plugin appears as a small blue book in the upper right corner.

4.0.1.1 Browsing Mode

4.0.1.1.1 Flag-Lists

In this mode, the plugin checks the URL in the address bar. It isolates the domain of the URL and checks the "reputation" of this domain. If the reputation of the domain has been flagged in the past, a popup window appears and a red exclamation mark is set in the blue book logo. Let us see the example in Fig 4.2. We see that the user visited the website www.bighairynews.com. The plugin added an exclamation mark in the blue book and created a pop-up message which in the blue background reads: "This domain appears as questionable in link: https://bit.ly/2Q5UHkQ. This means that the domain is flagged in the list https://bit.ly/2Q5UHkQ.

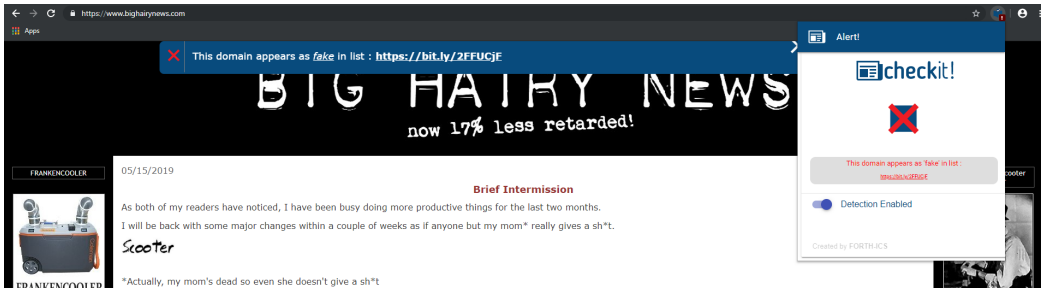

Figure 4.2: The plugin warns the user about the domain's credibility

4.0.1.1.2 Fact Check Similarity

As mentioned in section 3.2, Check-It capitalizes on fact-checking websites, by cross-checking every article processed by its plugin against a list of fact-checking websites, generating an informative warning when an article happens to be found listed on these web sites. In the following example, the plugin informs the user that the article she is reading appears to be similar to an article flagged by the snopes.com fact-checking the site as "fake".

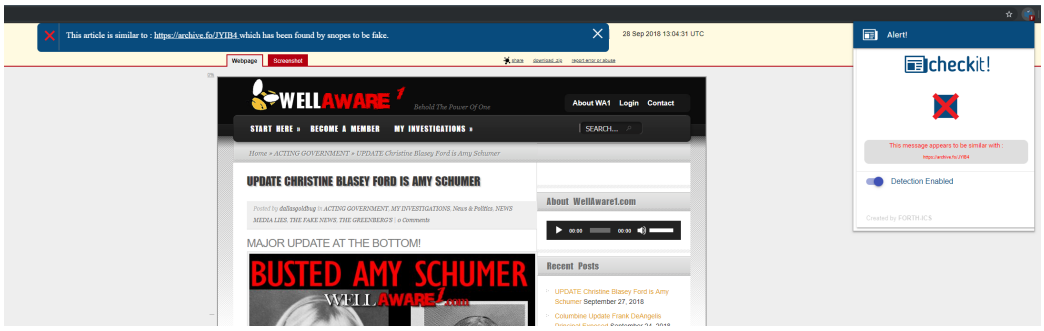

Figure 4.3: The plugin warns the user that the content is similar to a fake article

4.0.1.1.3 Linguistic Model

As mentioned in section 3.4, the linguistic component analyzes the actual content of the news article. Check-It extracts the headline and body from the article and informs the user regarding the article's veracity. Fig. 4.4 demonstrates an example where linguistic analysis of Check-It considers the article as "fake".

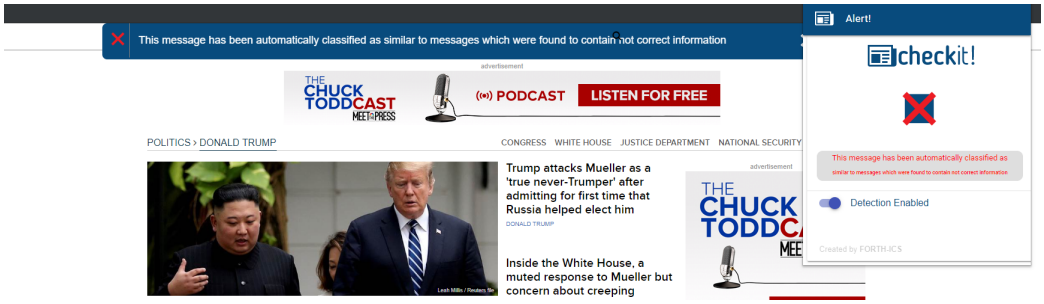

Figure 4.4: The plugin warns the user that our linguistic analysis considers the article as "fake"

4.0.1.2 Social Media mode

When the user browses a social media web page (say for example twitter), checking the domain of the URL in the address bar is of no help as the domain name will be www.twitter.com and will not be flagged. To check whether the page contains fake news or other questionable content, one needs to dig down into the page contents and find (in this case) the individual tweets. This is exactly what our plugin does in this "Social Media" mode: it parses the twitter web page, finds each tweet, extracts the URLs of each tweet and finds if the URLs are flagged. If they are, the blue book with the red exclamation mark appears next to the tweet. Let us see a specific example.

In Fig. 4.5 below, the user browses the tweets from Newstutu1. The first tweet is flagged because it has a URL that points to 100percentfedup.com. the flagging can be seen by the appearance of the "blue book" logo with the red exclamation mark in the upper right corner of the tweet. If the user clicks on this logo, she will find the reason why the tweet is flagged.

We have added support for 5 different OSNs : Twitter[1], Facebook[2], Reddit[3], 4chan[4] and VK[5].

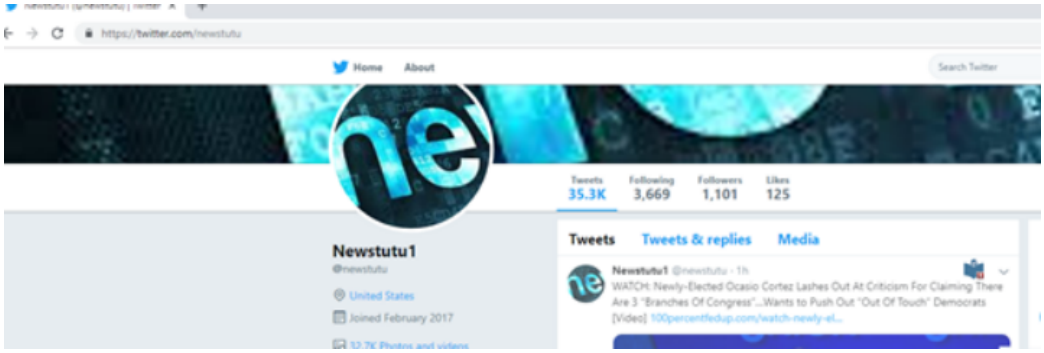

Figure 4.5: The plugin warns the user that a tweet contains a link to unreliable domain

[1] https://twitter.com
[2] https://www.facebook.com
[3] https://www.reddit.com
[4] http://www.4chan.org
[5] https://vk.com

Chapter 5

Evaluation

For the evaluation of the Check-It plugin, we focus on the linguistic model and the user-blacklist generated by the Online Social Network User Analysis component. The Fake News Flag-lists and Known Fact-Checked Articles are left out of the system evaluation since they provide us with about 100% accurate results. The task of these components is to transfer facts from the knowledge experts, such as the news site reviewers, from which the curated list of fake news domains was collected, and fact-checking organizations consisting of journalists, reporters and experts from related fields.

5.1 Linguistic Model Evaluation

For the implementation of the linguistic model presented in Section 3.4, Python Keras[1] has been used with Tensorflow[2] as back-end. The training epochs for the model have been fixed at 100 with mini-batches of 128. During training, *categorical cross-entropy* [50] has been used as loss function and *Adam* [51] as the optimization function. To prevent the model from over-fitting, a *early stopping* mechanism has been used. Early stopping is responsible for interrupting the training if the validation loss does not drop for 10 consecutive epochs. All the experiments were run in stratified 3-fold cross-validation and executed on a Virtual Machine with Ubuntu 16.4, 16 VCPUs and 32GB of RAM. For the different parameters and the sake of training time, we also used Google's Colab[3], a 12-hour free subscription to a Google Cloud VM with 13 GB of RAM and a Tesla K80 GPU. Finally, to be compatible with the user's internet browser, the model was exported with Tensorflow JS[4]. Tensorflow JS is a library for developing, training and exporting deep learning models in JavaScript, and deploying in the web browser.

Lacking state-of-the-art works that utilize the Fake News Corpus dataset, we

[1] `https://keras.io/`
[2] `https://www.tensorflow.org/`
[3] `https://colab.research.google.com/`
[4] `https://www.tensorflow.org/js`

Reference	Model	Acc.	P	R	F1
Shu et al. 2018 [20]	SVM_{LIWC}	0.610	0.602	0.561	0.555
	SVM_{RST}	0.655	0.683	0.628	0.623
Potthast et al. [52]	GRF_{STYLE}	0.550	0.520	0.525	0.520
	GRF_{TOPIC}	0.520	0.515	0.515	0.510
	ORF_{STYLE}	0.550	0.535	0.540	0.535
	ORF_{TOPIC}	0.580	0.555	0.555	0.560
Check-It Model	DNN	**0.703**	**0.713**	**0.703**	**0.700**

Table 5.1: Overall results on the comparison with the state-of-the-art for the Buzzfeed News (BF) dataset.

chose to additionally train the model with 5 other datasets and we compared our system against 3 state-of-the-art works [52, 20, 19]. Note that for a fair comparison we chose baselines that only consider news contents, similar to our approach. The selected datasets include Buzzfeed News (BF) and Politifact (PF), which are publicly available in the authors Github[5] repository. For evaluation metrics, we use accuracy, precision, recall and F1 score.

Shu et al. 2018 [20] utilize the BF and PF datasets in their work. The authors extracted news content features based on a combination of the vector space model and rhetorical structure theory (RST) [53] and the Linguistic Inquiry and Word Count (LIWC) lexicon [54], a widely used bundle of lexicons, that can extract psycholinguistic features to capture deception within the articles. These features were used to train two separate SVM classifiers, namely SVM_{RST} and SVM_{LIWC}. Furthermore, Shu et al. 2018b [19], utilize the GC and PF datasets to train several models, including an SVM, Logistic Regression (LR), Naive Bayes (NB) and a Convolutional Neural Network (CNN), focusing on the one-hot vector representation of the data. Potthast et al. [52] train 4 different Random Forest (RF) classifiers that consider the style and topic of the articles, 2 of them being generic, namely GRF_{STYLE} and GRF_{TOPIC}, and 2 of them considering the political orientation of the articles, namely ORF_{STYLE} and ORF_{TOPIC}. The authors utilize the BF dataset, having information regarding the article's political orientation.

Next, we present the overall results of the state-of-the-art and compare them with the performance of our model. Table 5.1 presents the results of the BF dataset and Table 5.2 presents the results of the PF dataset. As displayed in Tables 5.1 and 5.2, Check-It linguistic model outperforms the state-of-the-art works. Our DNN, based on the deep learning paradigm, does not depend on handcrafted features, it rather generates abstract features, able to better capture the writing style of fake news [55].

This is not the case for the GC dataset, where our model is marginally better to the LR and NB models, where it performs poorly compared to the authors' CNN

[5] `https://github.com/KaiDMML/FakeNewsNet/tree/master/dataset`

Reference	Model	Acc.	P	R	F1
Shu et al. 2018 [20]	SVM_{RST}	0.571	0.595	0.533	0.544
	SVM_{LIWC}	0.637	0.621	0.667	0.615
Shu et al. 2018b [19]	SVM	0.580	0.611	0.717	0.659
	LR	0.642	**0.757**	0.543	0.633
	NB	0.617	0.674	0.630	0.651
	CNN	0.629	0.807	0.456	0.583
Check-It Model	DNN	**0.722**	**0.725**	**0.725**	**0.722**

Table 5.2: Overall results on the comparison with the state-of-the-art for the Politifact (PF) dataset.

model. This is due to the nature of the dataset, which contains real and fake news that gossip about the relationship among celebrities. Based on the observations of the authors in [19], the real and fake articles in GC are slightly different, and for such news, it is difficult to classify them using only the news content, without the help of auxiliary information, such as social context.

Reference	Model	Acc.	P	R	F1
Shu et al. 2018b [19]	SVM	0.497	0.511	0.713	0.595
	LR	0.648	0.675	0.619	0.646
	NB	0.624	0.631	0.669	0.649
	CNN	**0.723**	**0.751**	**0.701**	**0.725**
Check-It Model	DNN	**0.647**	**0.648**	**0.647**	**0.647**

Table 5.3: Overall results on the comparison with the state-of-the-art for the GossipCop (GC) dataset.

The datasets used for this experiment were to compare our model to the existing state-of-the-art models. Training our model with datasets of a few hundred records like the above does not meet the expectations of deep learning [50]. Thus, as described in Section 3.4.1, we trained on *Fake News Corpus*, a dataset with millions of articles from domains, labeled as fake and real. **Our model can achieve an accuracy of 0.930, as well as 0.940 Precision, 0.937 Recall, and 0.937 F1 score**.

5.2 Optimization

Despite the promising results, an error margin of 0.07 still exists. For example, 7 out of 100 articles, our model labels them as fake, are in fact real. Imagine having an article from a widely known credible source like CNN, mistakenly be labeled as fake. In addition, one of our initial requirements is the "High Confidence" of our results. At this work, we assigned a proper threshold before the final labeling, to increase the confidence of the response. To reduce the error margin, we examined the number of false positives (FP) and true negatives (TN). FP is defined as the reliable articles classified as fake, whereas TN is defined as a fake article classified as real.

Figure 5.1 depicts the number of FP and TN as a function of the threshold, starting from 0.50 to 0.99 with a step of 0.01. To achieve maximum confidence, we chose the threshold to be 0.99, which resulted in 0 FP.

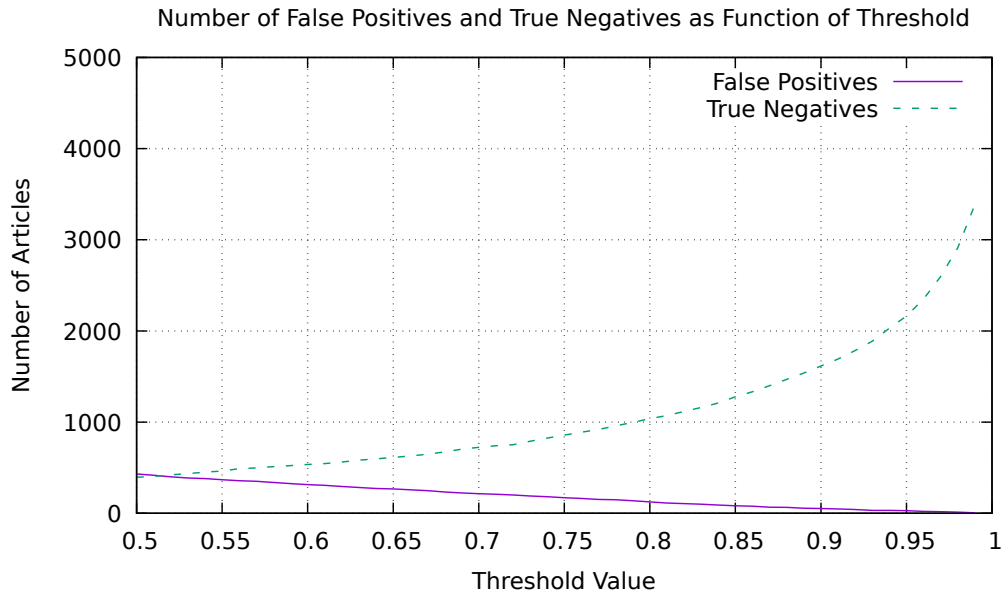

Figure 5.1: Number of False Positives and True Negatives as Function of Threshold

Chapter 6

Directions and Future Work

6.1 The big data problem

Before concluding, we will provide some directions for our future work. During this thesis, we have done similar observations with Torabi et al.[56] [57]. We consider these findings to be significant for the success of the linguistic methods for Fake News Detection. Although our system outperformed the state-of-the-art, we have reached a barrier. We believe that this barrier, that creates a hardship in achieving a better accuracy, is related to the quality of data.

The goal of the style-based (linguistic) methods is to capture the "deception style", meaning the writing style of fake news writers. Yet, there are some pitfalls with these methods. They may end up capturing the writing style of specific authors instead of the deception style. This way, even if a trained model performs well in similar data, they do not generalize. The writing style of a document is defined by the author, the topic and the editor (if there is one). So, a dataset should contain many authors, news domains and topics.

Finally, as stated in [56], Fake News is a big data problem. We need many data points, solid ground truth and balance over 4 axes: labels, topics, authors and news websites. For example, some datasets are created by downloading articles from websites in a flag-list. Although a website that is found to publish fake news, in order to maintain some credibility, it may also publish credible stories. Creating a dataset this way will add noise to the data and reduce accuracy.

To sum up, to avoid this pitfall, a quality Fake News dataset should have the following properties:

- The ground truth should be solid.
- It should cover many topics.
- It should include different authors and domains that publish articles.

We propose the creation of a fake news data evaluation framework. That framework can perform topic modeling and stylometric analysis to detect unbalanced

datasets. A dataset with many data points that also gets a large score at this evaluation will better generalize in real-world (unseen) data.

6.2 Hoax Memes

We note that currently, we are working to extend Check-It with an additional component. This component will detect not textual but visual content linked with fake news stories. This component shares similarities with the 'Fact Check Similarity' component in that it facilitates similarity preserving hashing to detect such content. We are working at detecting similarities with memes annotated as 'hoax' or a similar tag by the knowyourmeme website[1]. To identify similar memes, we are using fingerprints extracted by phash[2]. Phash is a perceptual hashing algorithm, meaning that is an algorithm that produces a snippet or fingerprint of various forms of multimedia. Phash is mathematically based on discrete cosine transformations.

[1] `https://knowyourmeme.com`
[2] `https://www.phash.org`

Chapter 7

Conclusion

In this work, we study the timely problem of fake news detection. While existing work has typically addressed the problem by focusing on either analyzing the text, the URL domain, or social networking features, we argue that it is important to incorporate all these signals.

In this thesis, we presented Check-It , a fake news detection system developed as a web browser plugin. Check-it aims to take a bold step towards detecting and reducing the spread of misinformation on the Web. To do so, it empowers its users with the tools they need to identify fake news. The major challenge of fake news detection stems from newly emerged news on which existing approaches only showed unsatisfactory performance.

To address this issue, we propose a pipeline based on a variety of signals, ranging from domain name flag-lists to deep learning approaches. Extensive experiments showcase that Check-It is effective and can outperform the state-of-the-art models.

Appendices

Appendix A

Table A.1: Linguistic Model's Feature List

No	Feature	Category	Extracted via
1	Word count	Stylistic	Text Processing
2	Words per sentence	Stylistic	Text Processing
3	Number of nouns	Stylistic	Tool/Classifier
4	Number of proper nouns	Stylistic	Tool/Classifier
5	Number of possessive pronouns	Stylistic	Tool/Classifier
6	Number of Wh-pronoun	Stylistic	Tool/Classifier
7	Number of determinants	Stylistic	Tool/Classifier
8	Number of Wh-determinants	Stylistic	Tool/Classifier
9	Number of cardinal numbers	Stylistic	Tool/Classifier
10	Number of adverbs	Stylistic	Tool/Classifier
11	Number of interjections	Stylistic	Tool/Classifier
12	Number of verbs	Stylistic	Tool/Classifier
13	Number of adjectives	Stylistic	Tool/Classifier
14	Number of past tense verbs	Stylistic	Tool/Classifier
15	Number of verb. Gerund or present particle	Stylistic	Tool/Classifier
16	Number of verb. past particle	Stylistic	Tool/Classifier
17	Number of verb. non-3rd person singular present	Stylistic	Tool/Classifier
18	Number of verb. 3rd person singular present	Stylistic	Tool/Classifier
19	Number of past tense words	Stylistic	Tool/Classifier
20	Number of future tense words	Stylistic	Tool/Classifier
21	Number of I pronouns	Stylistic	Text Processing
22	Number of we pronouns	Stylistic	Text Processing
23	Number of you pronouns	Stylistic	Text Processing
24	Number of he/she pronouns	Stylistic	Text Processing
25	Number of quantifying words	Stylistic	Dictionary
26	Number of comparison words	Stylistic	Dictionary
27	Number of exclamation marks	Stylistic	Text Processing
28	Number of negations	Stylistic	Tool/Classifier

29	Number of slang words	Stylistic	Dictionary
30	Number of swear words	Stylistic	Dictionary
31	Number of interrogatives	Stylistic	Dictionary
32	Number of ALL_CAPITAL words	Stylistic	Text Processing
33	Percentage of stop-words	Stylistic	Dictionary
34	Number of punctuation	Stylistic	Text Processing
35	Number of quotes	Stylistic	Text Processing
36	Number of verb phrases	Stylistic	Tool/Classifier
37	Gunning Fog Grade Readability Index	Complexity	Tool/Classifier
38	SMOG Readability Index	Complexity	Tool/Classifier
39	Flesch-Kincaid Grade Readability Index	Complexity	Tool/Classifier
40	Median depth of syntax tree	Complexity	Tool/Classifier
41	Median depth of noun phrase tree	Complexity	Tool/Classifier
42	Median depth of verb phrase tree	Complexity	Tool/Classifier
43	Average frequency of least common 3 words	Complexity	Dictionary
44	Average frequency of words in each document	Complexity	Dictionary
45	Type-Token Ratio	Complexity	Text Processing
46	Average length of each word	Complexity	Text Processing
47	PCFG score	Complexity	Tool/Classifier
48	Rhetorical Structure Score	Complexity	Tool/Classifier
49	Number of analytical words	Psychology	Dictionary
50	Number of insightful words	Psychology	Dictionary
51	Number of casual words	Psychology	Dictionary
52	Number of discrepancy words	Psychology	Dictionary
53	Number of tentative words	Psychology	Dictionary
54	Number of certainty words	Psychology	Dictionary
55	Number of differentiation words	Psychology	Dictionary
56	Number of affiliation words	Psychology	Dictionary
57	Number of power words	Psychology	Dictionary
58	Number of reward words	Psychology	Dictionary
59	Number of risk words	Psychology	Dictionary
60	Number of personal concern words	Psychology	Dictionary
61	Number of emotional tone words	Psychology	Dictionary
62	Number of emotion words	Psychology	Dictionary
63	Negative sentiment score	Psychology	Dictionary
64	Positive sentiment score	Psychology	Dictionary
65	Subjectivity score	Psychology	Tool/Classifier
66	# of characters	Stylistic	Text Processing
67	# of alphabetic characters / # of characters	Stylistic	Text Processing
68	# of uppercase characters /# of characters	Stylistic	Text Processing
69	# of digit characters / # of characters	Stylistic	Text Processing

70	# of white space characters / # of characters	Stylistic	Text Processing
72	Frequency of letters [26 features]	Stylistic	Text Processing
73	Frequency of special characters [21 features]	Stylistic	Text Processing
74	Happax legomena	Stylistic	Text Processing
75	Happax dislegomena	Stylistic	Text Processing
76	Yule's K measure	Stylistic	Text Processing
77	Simpson's D measure	Stylistic	Text Processing
78	Sichel's S measure	Stylistic	Text Processing
79	Brunet's W measure	Stylistic	Text Processing
80	Honore's R measure	Stylistic	Text Processing
81	Word length frequency distribution / # of words	Stylistic	Text Processing
82	Frequency of function words	Complexity	Dictionary
83	Total number of lines	Complexity	Text Processing
84	Total number of sentences	Complexity	Text Processing
85	Has quoted content (0/1)	Stylistic	Text Processing
86	Coleman-Liau Readability Index	Complexity	Tool/Classifier
87	Automated Readability Index	Complexity	Tool/Classifier
88	Dale Chall Readability Index	Complexity	Tool/Classifier
89	Linsear Readability Index	Complexity	Tool/Classifier

Table A.2: List of Fact-Checking Websites

No	Name	URL
1	FactCheck	https://www.factcheck.org/
2	Snopes	https://www.snopes.com/
3	Politifact	https://www.politifact.com/
4	Emergent	http://www.emergent.info/
5	MediaBugs	http://mediabugs.org/
6	Hoax-Slayer	http://hoax-slayer.net
7	TruthOrFiction	https://www.truthorfiction.com/

Table A.3: Flaglists incorporated in Check-It

No	Name	URL
1	Kaggle Fake-News	https://www.kaggle.com/mrisdal/fake-news
2	OpenSources	https://raw.githubusercontent.com/BigMcLargeHuge/opensources
3	Politifact	https://www.politifact.com/
4	Greek-Hoaxes	https://raw.githubusercontent.com/Ellinika-Hoaxes
5	MediaBiasFactcheck	https://mediabiasfactcheck.com
6	FakeNewsNet	https://github.com/KaiDMML/FakeNewsNet/tree/old-version/Data
7	Butac	http://butac.it/the-black-list
8	Bufale	https://www.bufale.net/the-black-list-la-lista-nera-del-web

Bibliography

[1] Natalie Nougayrède. In this age of propaganda, we must defend ourselves. here's how — natalie nougayrède, Jan 2018.

[2] A short guide to the history of 'fake news' and disinformation, Jul 2018.

[3] Miriam Fernandez and Harith Alani. Online misinformation: Challenges and future directions. In *Companion Proceedings of the The Web Conference 2018*, WWW '18, pages 595–602, Republic and Canton of Geneva, Switzerland, 2018. International World Wide Web Conferences Steering Committee.

[4] Keith Collins. People shared nearly as much fake news as real news on twitter during the election, Sep 2017.

[5] Faiz Siddiqui and Susan Svrluga. N.c. man told police he went to d.c. pizzeria with gun to investigate conspiracy theory, Dec 2016.

[6] Craig Silverman. This analysis shows how viral fake election news stories outperformed real news on facebook, Nov 2016.

[7] Georgina Lee. Qa on cambridge analytica: The allegations so far, explained.

[8] Savvas Zannettou, Michael Sirivianos, Jeremy Blackburn, and Nicolas Kourtellis. The web of false information: Rumors, fake news, hoaxes, clickbait, and various other shenanigans. *Journal of Data and Information Quality (JDIQ)*, 11(3):10, 2019.

[9] Fake news and cyber propaganda: The use and abuse of social media.

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

[11] Yochai Benkler, Robert Faris, and Hal Roberts. *Network propaganda: Manipulation, disinformation, and radicalization in American politics*. Oxford University Press, 2018.

[12] Richard McCreadie, Craig Macdonald, and Iadh Ounis. Crowdsourced rumour identification during emergencies. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15 Companion, pages 965–970, New York, NY, USA, 2015. ACM.

[13] Zhe Zhao, Paul Resnick, and Qiaozhu Mei. Enquiring minds: Early detection of rumors in social media from enquiry posts. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, pages 1395–1405, Republic and Canton of Geneva, Switzerland, 2015. International World Wide Web Conference.

[14] Natali Ruchansky, Sungyong Seo, and Yan Liu. Csi: A hybrid deep model for fake news detection. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, pages 797–806, New York, NY, USA, 2017. ACM.

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

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

[17] Niall J. Conroy, Victoria L. Rubin, and Yimin Chen. Automatic deception detection: Methods for finding fake news. In *Proceedings of the 78th ASIS&T Annual Meeting: Information Science with Impact: Research in and for the Community*, ASIST '15, pages 82:1–82:4, Silver Springs, MD, USA, 2015. American Society for Information Science.

[18] James Delingpole. Delingpole: All of recent u.s. warming has been faked by noaa, Aug 2017.

[19] Kai Shu, Deepak Mahudeswaran, Suhang Wang, Dongwon Lee, and Huan Liu. Fakenewsnet: A data repository with news content, social context and dynamic information for studying fake news on social media. *CoRR*, abs/1809.01286, 2018.

[20] Kai Shu, Suhang Wang, and Huan Liu. Exploiting tri-relationship for fake news detection. *CoRR*, abs/1712.07709, 2017.

[21] Amy X. Zhang, Aditya Ranganathan, Sarah Emlen Metz, Scott Appling, Connie Moon Sehat, Norman Gilmore, Nick B. Adams, Emmanuel Vincent, Jennifer Lee, Martin Robbins, Ed Bice, Sandro Hawke, David Karger, and An Xiao Mina. A structured response to misinformation: Defining and annotating credibility indicators in news articles. In *Companion Proceedings of the The Web Conference 2018*, WWW '18, pages 603–612, Republic and Canton of Geneva, Switzerland, 2018. International World Wide Web Conferences Steering Committee.

[22] Denis Teyssou, Jean-Michel Leung, Evlampios Apostolidis, Konstantinos Apostolidis, Symeon Papadopoulos, Markos Zampoglou, Olga Papadopoulou, and Vasileios Mezaris. The invid plug-in: web video verification on the browser. In *Proceedings of the First International Workshop on Multimedia Verification*, pages 23–30. ACM, 2017.

[23] Markos Zampoglou, Symeon Papadopoulos, Yiannis Kompatsiaris, Ruben Bouwmeester, and Jochen Spangenberg. Web and social media image forensics for news professionals. In *Tenth International AAAI Conference on Web and Social Media*, 2016.

[24] Aditi Gupta, Ponnurangam Kumaraguru, Carlos Castillo, and Patrick Meier. Tweetcred: Real-time credibility assessment of content on twitter. In *International Conference on Social Informatics*, pages 228–243. Springer, 2014.

[25] Diego Saez-Trumper. Fake tweet buster: a webtool to identify users promoting fake news ontwitter. In *Proceedings of the 25th ACM conference on Hypertext and social media*, pages 316–317. ACM, 2014.

[26] Naeemul Hassan, Gensheng Zhang, Fatma Arslan, Josue Caraballo, Damian Jimenez, Siddhant Gawsane, Shohedul Hasan, Minumol Joseph, Aaditya Kulkarni, Anil Kumar Nayak, et al. Claimbuster: the first-ever end-to-end fact-checking system. *Proceedings of the VLDB Endowment*, 10(12):1945–1948, 2017.

[27] Chengcheng Shao, Giovanni Luca Ciampaglia, Alessandro Flammini, and Filippo Menczer. Hoaxy: A platform for tracking online misinformation. In *Proceedings of the 25th international conference companion on world wide web*, pages 745–750. International World Wide Web Conferences Steering Committee, 2016.

[28] James Thorne and Andreas Vlachos. Automated fact checking: Task formulations, methods and future directions. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3346–3359, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics.

[29] Fintan Culwin and Thomas Lancaster. Plagiarism, prevention, deterrence and detection. *Available for ILT members from*, 2001.

[30] Saul Schleimer, Daniel S Wilkerson, and Alex Aiken. Winnowing: local algorithms for document fingerprinting. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 76–85. ACM, 2003.

[31] Jesse Kornblum. Identifying almost identical files using context triggered piecewise hashing. *Digital investigation*, 3:91–97, 2006.

[32] Nikolaos Sarantinos, Chafika Benzaïd, Omar Arabiat, and Ameer Al-Nemrat. Forensic malware analysis: The value of fuzzy hashing algorithms in identifying similarities. In *2016 IEEE Trustcom/BigDataSE/ISPA*, pages 1782–1787. IEEE, 2016.

[33] Aristides Gionis, Piotr Indyk, Rajeev Motwani, et al. Similarity search in high dimensions via hashing. In *Vldb*, volume 99, pages 518–529, 1999.

[34] Gurmeet Singh Manku, Arvind Jain, and Anish Das Sarma. Detecting near-duplicates for web crawling. In *Proceedings of the 16th international conference on World Wide Web*, pages 141–150. ACM, 2007.

[35] Andrei Z Broder. On the resemblance and containment of documents. In *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No. 97TB100171)*, pages 21–29. IEEE, 1997.

[36] Jonathan Oliver, Chun Cheng, and Yanggui Chen. Tlsh–a locality sensitive hash. In *2013 Fourth Cybercrime and Trustworthy Computing Workshop*, pages 7–13. IEEE, 2013.

[37] Ernesto Damiani, Sabrina De Capitani di Vimercati, Stefano Paraboschi, and Pierangela Samarati. An open digest-based technique for spam detection. *ISCA PDCS*, 2004:559–564, 2004.

[38] Morris H. DeGroot. Reaching a consensus. *Journal of the American Statistical Association*, 69:118–121, 03 1974.

[39] Benjamin D. Horne and Sibel Adali. This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news. *CoRR*, abs/1703.09398, 2017.

[40] Victoria Rubin, Niall Conroy, Yimin Chen, and Sarah Cornwell. Fake news or truth? using satirical cues to detect potentially misleading news. In *Proceedings of the Second Workshop on Computational Approaches to Deception Detection*, pages 7–17, San Diego, California, June 2016. Association for Computational Linguistics.

[41] Yafang Wang, Gerard de Melo, and Gerhard Weikum. Five shades of untruth: Finer-grained classification of fake news. *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 593–594, 2018.

[42] Yaqing Wang, Fenglong Ma, Zhiwei Jin, Ye Yuan, Guangxu Xun, Kishlay Jha, Lu Su, and Jing Gao. 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*, KDD '18, pages 849–857, New York, NY, USA, 2018. ACM.

[43] Bing Liu, Minqing Hu, and Junsheng Cheng. Opinion observer: Analyzing and comparing opinions on the web. In *Proceedings of the 14th International Conference on World Wide Web*, WWW '05, pages 342–351, New York, NY, USA, 2005. ACM.

[44] Jesse Graham, Jonathan Haidt, and Brian Nosek. Liberals and conservatives rely on different sets of moral foundations. *Journal of personality and social psychology*, 96:1029–46, 06 2009.

[45] Finn Årup Nielsen. A new evaluation of a word list for sentiment analysis in microblogs. *CoRR*, abs/1103.2903, 2011.

[46] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. *CoRR*, abs/1503.02406, 2015.

[47] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[48] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015.

[49] Aneesha Bhat. *Locality Sensitive Indexing for Efficient High-Dimensional Query Answering in the Presence of Excluded Regions.* Arizona State University, 2016.

[50] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.* MIT Press, 2016. `http://www.deeplearningbook.org`.

[51] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[52] Martin Potthast, Johannes Kiesel, Kevin Reinartz, Janek Bevendorff, and Benno Stein. A stylometric inquiry into hyperpartisan and fake news. *CoRR*, abs/1702.05638, 2017.

[53] Victoria L. Rubin and Tatiana Lukoianova. Truth and deception at the rhetorical structure level. *Journal of the Association for Information Science and Technology*, 66(5):905–917, 2015.

[54] James W Pennebaker, Ryan L Boyd, Kayla Jordan, and Kate Blackburn. The development and psychometric properties of liwc2015. Technical report, 2015.

[55] Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. Truth of varying shades: Analyzing language in fake news and political fact-checking. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2931–2937, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[56] Fatemeh Torabi Asr and Maite Taboada. Big data and quality data for fake news and misinformation detection. *Big Data & Society*, 6(1):2053951719843310, 2019.

[57] Fatemeh Torabi Asr and Maite Taboada. The data challenge in misinformation detection: Source reputation vs. content veracity. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 10–15, 2018.