Check-It: A plugin for Detecting and Reducing the Spread of Fake News and Misinformation on the Web

Demetris Paschalides, Chrysovalantis Christodoulou, Rafael Andreou, George Pallis, Marios D. Dikaiakos
{dpasch01,cchris47,randre07,gpallis,mdd}@cs.ucy.ac.cy
Computer Science Department, University of Cyprus
Nicosia, Cyprus

Alexandros Kornilakis, Evangelos Markatos
{kornilak,markatos}@ics.forth.gr
Department of Computer Science, University of Crete
Heraklion, Crete, Greece

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. 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 work, we present Check-It, a system that combines a variety of signals into a pipeline for fake news identification. Check-It is developed as a web browser plugin with the objective of efficient and timely fake news detection, while respecting user privacy. In this paper, 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.

CCS CONCEPTS

• **Networks → Online social networks**; • **Computing methodologies → Lexical semantics**; Information extraction.

KEYWORDS

Web of Trust, Fake News Detection, News Content

ACM Reference Format:
Demetris Paschalides, Chrysovalantis Christodoulou, Rafael Andreou, George Pallis, Marios D. Dikaiakos and Alexandros Kornilakis, Evangelos Markatos. 2019. Check-It: A plugin for Detecting and Reducing the Spread of Fake News and Misinformation on the Web. In *IEEE/WIC/ACM International Conference on Web Intelligence (WI '19), October14–17, 2019, Thessaloniki, Greece.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3350546.3352534

1 INTRODUCTION

Misinformation is not a recent issue. When news offices started to connect to each other via wire, the authenticity of information became a concern. Editors did not really know whether the news coming in through the wire was credible. They 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 [2].

In recent years, researchers are seeking to better define and characterize misinformation and its place in the larger information ecosystem [16]. An important aspect of characterizing misinformation is to understand how people perceive the credibility of information [21]. Contradiction of beliefs and repetition are some major characteristics of fake news that make the problem complex and challenging, indicating that more exploration is needed [7].

Social media companies are already partnering with fact-checking organizations and adopting crowd-sourcing techniques to detect fake news in social media. For instance, the First Draft News project CrossCheck[1] is a collaborative verification program involving technology firms including Facebook and Google, aiming to help citizens make informed choices. Similarly, the Washington Post asked its readers to use the term 'Fake News' for reporting the fake news on the website. In addition, some effort has been done to detect fake news, including approaches that apply text-based methods[1] and fact-checking through knowledge graphs [11]. However, the current fact-checkers and crowdsourcing initiatives have limitations since they cannot cope with the high volume of misinformation generated online.

Despite the increasing interest in analyzing fake news in the Web and the need for tools to deal with them [7], there has been little work in automatic fake news detection tools. The main challenge stems from the fact that it is difficult to develop classification algorithms to capture fake news. Researchers in [9] 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 [23], 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, an approach that has been proposed is the development of browser plugins, such as the B.S. Detector[2] and the NewsGuard[3], which flag content from fake news sources using a constantly-updated list of known fake news sites as a reference point. However, they try to solve the problem using only one signal of information (i.e., fact-check, linguistic, social network). In

WI '19, October 14–17, 2019, Thessaloniki, Greece
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6934-3/19/10…$15.00
https://doi.org/10.1145/3350546.3352534

[1] https://firstdraftnews.org/project/crosscheck/
[2] http://bsdetecor.tech
[3] http://www.newsguardtech.com/

contrast, our approach combines intelligently the variety of signals, making it able to calculate the credibility of a piece of news and successfully warn the reader. A 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. Our main objective is to provide a Web browser plugin that detects efficiently and timely the fake news articles respecting the user's privacy. We empirically evaluate our proposed method via experiments on real-world datasets from Twitter and news articles, demonstrating that our approach significantly improves the performance on detecting and reducing the spread of fake news and misinformation on the Web.

The rest of this work is organized as follows. In Section 2, we present the Check-It plugin and its components, in Section 3 we describe our experimental setup and the evaluation of the performance of our approach, and finally, in Section 4, we conclude this paper.

2 CHECK-IT SYSTEM

Check-It satisfies a series of user-centric functional requirements: **i) Preserve User Privacy**: Check-It plugin should work locally, on the user's web browser, without the need of external communication (i.e. a RESTful API). **ii) Highly Confident Identification**: Check-It labels a piece of news as fake if it is highly confident about it. **iii) Low Response Time**: All the required resources are efficiently loaded in the user's web browser. **iv) Lightweight Computation**: Asynchronous processing and parallelization is taken place so as to minimize the load of the plugin.

As depicted in Figure 1, Check-It system consists of four main components that function as a pipeline for fake news identification on the web: a) The Flag-list Matcher component matches domains of news articles to Known Fake News Domains and Fact Checks; b) the Fact Check Similarity component which compares a piece of news against Known Fact Checked Articles labelled as fake from Fact Checking organizations, such as Politifact[4] and Snopes[5]; c) the Online Social Network User Analysis component which is responsible for analyzing user behavior in social networks and producing a User-Blacklist of fake news propagators; and d) the Linguistic Model component, an artificial intelligence model,which 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, made 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 Resource Package is downloaded and extracted on the user's end, and any critical updates on any of the resources.

At the Check-It Plugin User Installment, the resources are loaded within the plugin, and assigned to their respective components. The Linguistic Model requires the features from the article's to be extracted. To this end, the JavaScript Feature Extraction Library was developed, responsible to capture the required features from within the article, and use them as input to the Linguistic Model Binary.

2.1 Flag-list Matcher

Some domain names are well known for spreading misinformation. Currently, there are several lists maintained by researchers, (referred to as *flag-lists*) containing domain names known for spreading misinformation. These lists are maintained by researchers or volunteers. Our system uses a non-exhaustive list of the flag-lists that includes Kaggle[6], OpenSources[7] and Greek-Hoaxes[8]. URL flag-lists and domain name checking are the simplest way for an initial assessment of the trustworthiness of a news article. Unfortunately, flag-lists do not test the truthfulness of the article itself, 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 2.2) and (ii) machine learning approaches (Section 2.3 and 2.4).

2.2 Fact Check Similarity

A number of 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 on online social media [19]. 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.

2.3 Online Social Network User Analysis

Since OSNs play an important role in the propagation of fake news [7], 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 model [4] for the user falsity calculation. Figure 2 presents the pipeline of the module and its components.

The system design of Check-It facilitates integration with multiple OSN platforms. Currently, we only support Twitter due its massive popularity and the ease-of-access to its data stream via the Twitter Streaming API[9]. 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

[4] https://www.politifact.com/
[5] https://www.snopes.com/
[6] https://www.kaggle.com/mrisdal/fake-news
[7] https://raw.githubusercontent.com/BigMcLargeHuge/opensources
[8] https://raw.githubusercontent.com/Ellinika-Hoaxes/Greek-Hoaxes-Detector
[9] https://developer.twitter.com/en/docs/tweets/filter-realtime/overview

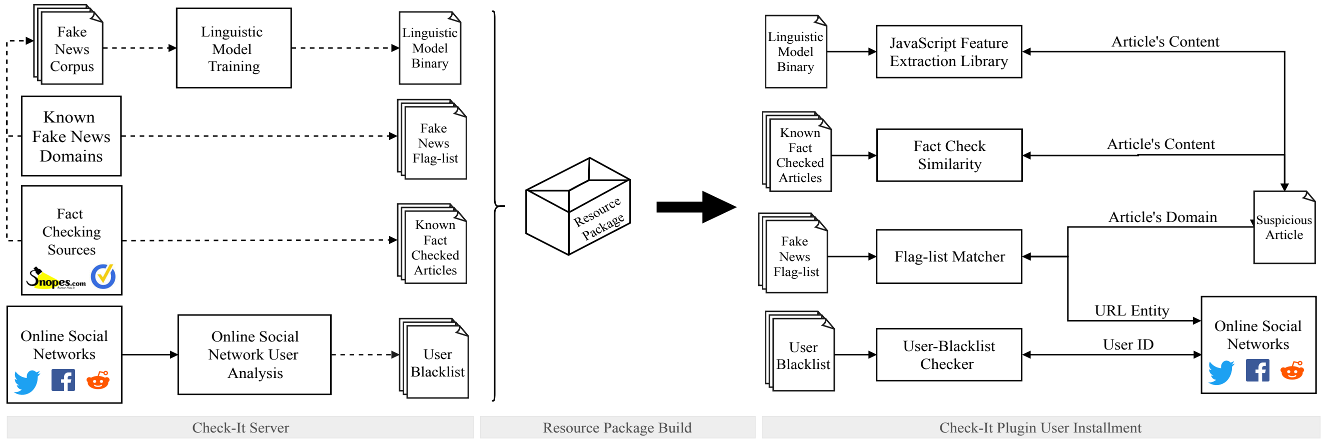

Figure 1: Architectural diagram for the Check-It System.

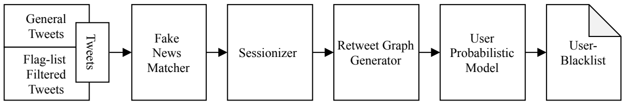

Figure 2: Architectural diagram for the social network signal.

model in groups of 1-hour sessions (Sessionizer task) [21]. 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 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 $p_i^{(0)} = 0$. 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).

2.4 Linguistic Model

Check-It incorporates textual features extracted from the headline and body of an article which have been widely used to detect fake news [6, 22]. A Deep Neural Network (DNN) is trained on the extracted features to predict the article's veracity. Next, we present an overview of the article dataset, the different linguistic features, and the DNN model.

2.4.1 Dataset Overview. Check-It makes use of Fake News Corpus[10], 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 categories labels: *fake news*, *satire*, *extreme bias*, *conspiracy theory*, *rumor mill*, *state news*, *junk science*, *hate news*, *clickbait*, *political*, and *credible*. In our approach, we focus only on the *fake news* and *credible* consisting of 1 million and 2 million articles respectively.

2.4.2 Linguistic Features. We compute different linguistic features of 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 1). We group these features into 3 categories: *stylistic*, *complexity* and *psychological*. **Stylistic Features** include the frequency of stopwords, punctuation, quotes, negations and words that appear in all capital letters and the frequency of Part-of-Speech tags in the text. **Complexity Features** include word-level metrics such as readability indexes and vocabulary richness. For readability indexes we used Gunning Fog, SMOG Grade and Flesh-Kincaid. For the vocabulary richness we computed the Type-Token Ratio, and the number of hapax legomenon and dis legomenon. **Psychological Features** include the count of words found in expert dictionaries that are associated with different psychological processes. These dictionaries include the negative and positive opinion lexicon [8], and the moral foundation dictionary [3], as well as the sentiment score, computed via the AFINN sentiment lexicon [10].

2.4.3 Deep Neural Network Model. The proposed DNN model adopts the cone-like structure, referred to as the bottleneck principle, and is known to perform well with numerical features [5, 20]. The features are converted into numerical vectors and used as input to a dense neural network composed of a sequence of 5 layers that consist of 512, 256, 128, 64 and 32 neurons respectively. The inner-neurons are activated using the hyperbolic tangent activation function (tanh). Finally, the classification layer consists of one neuron per class with the softmax activation function.

3 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 reside on the expert knowledge of fact-checkers.

[10] https://github.com/several27/FakeNewsCorpus

Reference	Model	Acc.	P	R	F1
Shu et al. 2018 [18]	SVM_{LIWC}	0.610	0.602	0.561	0.555
	SVM_{RST}	0.655	0.683	0.628	0.623
Potthast et al. [13]	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*	**0.715**	**0.719**	**0.715**	**0.714**

Table 1: Overall results on the comparison with the state-of-the-art for the Buzzfeed News (BF) dataset.

Reference	Model	Acc.	P	R	F1
Shu et al. 2018 [18]	SVM_{RST}	0.571	0.595	0.533	0.544
	SVM_{LIWC}	0.637	0.621	0.667	0.615
Shu et al. 2018b [17]	*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*	**0.728**	**0.734**	**0.727**	**0.725**

Table 2: Overall results on the comparison with the state-of-the-art for the Politifact (PF) dataset.

3.1 Linguistic Model Evaluation

To evaluate the performance of our approach, we compared it against three state-of-the-art works [13, 17, 18]. The datasets used in these works include the Buzzfeed News (*BF*) and the Politifact (*PF*), which are publicly available [11].

In [18], Shu et al. apply two separate SVM classifiers, namely SVM_{RST} and SVM_{LIWC}, into both the *BF* and *PF* datasets. The authors extracted news content features based on a combination of the vector space model and rhetorical structure theory (RST) [15] and the Linguistic Inquiry and Word Count (LIWC) lexicon [12]. Furthermore, in [17], Shu et al. apply different classifiers only to the *PF* dataset. The classifiers used in this work include an SVM, a Logistic Regression (LR), a Naive Bayes (NB) and a Convolutional Neural Network (CNN), focusing on one-hot vector representation of the data. Lastly, in [13] the authors train four different Random Forest (RF) classifiers to the *BF* dataset since it contains information of the article's political orientation. The features of the four classifiers are extracted from the style and topic of the articles. Two of the classifiers consider the political orientation of the articles, ORF_{STYLE} and ORF_{TOPIC}, whereas the other two are generic, namely GRF_{STYLE} and GRF_{TOPIC}.

Note that for a fair comparison, from each dataset, we chose baselines that only consider news contents, similar to our approach. The training for all the datasets, for all the experiments was run in a stratified 3-fold cross validation.

As displayed in Tables 1 and 2, Check-It linguistic model outperforms all the three state-of-the-art works on both datasets. This is due to the fact that our DNN, based on the deep learning paradigm, is able to better capture the writing style of fake news [14].

We additionally trained our model on *Fake News Corpus*, described in Section 2.4.1. The experimental results also show the excellent performance of our linguistic model. Our model achieved an accuracy of **0.930**, as well as **0.940** Precision, **0.937** Recall, and **0.937** F1 score.

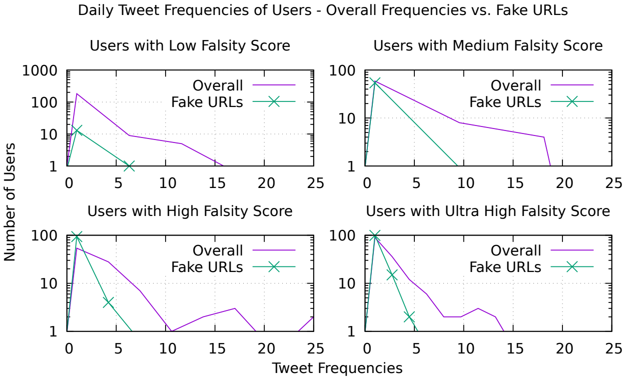

Figure 3: Tweeting frequencies of users assigned with "Low", "Medium", "High" and "Ultra High" falsity scores.

3.2 OSN User Analysis Evaluation

The task of Online Social Network User Analysis component is to build the User-Blacklist that includes the users who disseminate misinformation through social media. Our evaluation took place from October 31_{st} 2018 to December 2_{nd} 2018, with a total of 150M tweets, from which the 30M contained URLs from known fake news domains. In terms of user accounts, we processed a total of 8.1M unique users. The users are categorized into buckets of low [0-0.25), medium [0.25-0.50), high [0.50-0.75) and ultra high [0.75-1.0) falsity scores. As we move from a low score to an ultra high score, the probability of a user to disseminate a fake article is increased.

Taking into account the daily tweeting frequencies of the users in each group, we compared them with the frequencies of tweets containing URLs of known fake news domains (fake URLs). Figure 3 depicts the frequencies of all the tweets and the tweets containing fake URLs. We see that users falling into the low falsity group have a large overall tweeting frequency with low frequency of tweets with fake URLs. Medium and high falsity groups present a rise on the frequency of tweets containing fake URLs. Users with ultra high falsity score seem to have lower overall frequency, but rather high frequency of tweets with fake URLs. Thus, the User-Blacklist consists of the users from the ultra high falsity group.

4 CONCLUSION

In this paper, 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. The major challenge of fake news detection stems from newly emerged news on which existing approaches only showed unsatisfactory performance. In order 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. An extended version of this work can be found at: $https://arxiv.org/abs/$1905.04260.

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

ACKNOWLEDGMENTS

This work is supported by the Google Digital News Initiative grant.

REFERENCES

[1] Niall J. Conroy, Victoria L. Rubin, and Yimin Chen. 2015. 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)*. Article 82, 4 pages.

[2] Miriam Fernandez and Harith Alani. 2018. Online Misinformation: Challenges and Future Directions. In *Companion Proceedings of the The Web Conference 2018 (WWW '18)*. 595–602.

[3] Jesse Graham, Jonathan Haidt, and Brian Nosek. 2009. Liberals and Conservatives Rely on Different Sets of Moral Foundations. *Journal of personality and social psychology* 96 (06 2009), 1029–46. https://doi.org/10.1037/a0015141

[4] Morris H. DeGroot. 1974. Reaching a Consensus. *J. Amer. Statist. Assoc.* 69 (03 1974), 118–121. https://doi.org/10.2307/2285509

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. *CoRR* abs/1512.03385 (2015).

[6] Benjamin D. Horne and Sibel Adali. 2017. 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).

[7] 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. 2018. The science of fake news. 359, 6380 (2018), 1094–1096.

[8] Bing Liu, Minqing Hu, and Junsheng Cheng. 2005. Opinion Observer: Analyzing and Comparing Opinions on the Web. In *Proceedings of the 14th International Conference on World Wide Web (WWW '05)*. 342–351.

[9] Richard McCreadie, Craig Macdonald, and Iadh Ounis. 2015. Crowdsourced Rumour Identification During Emergencies. In *Proceedings of the 24th International Conference on World Wide Web (WWW '15 Companion)*. 965–970.

[10] Finn Årup Nielsen. 2011. A new Evaluation of a word list for sentiment analysis in microblogs. *CoRR* abs/1103.2903 (2011).

[11] Jeff Z. Pan, Siyana Pavlova, Chenxi Li, Ningxi Li, Yangmei Li, and Jinshuo Liu. 2018. Content Based Fake News Detection Using Knowledge Graphs. Springer Verlag, 669–683. https://doi.org/10.1007/978-3-030-00671-6_39

[12] James W Pennebaker, Ryan L Boyd, Kayla Jordan, and Kate Blackburn. 2015. *The development and psychometric properties of LIWC2015.* Technical Report.

[13] Martin Potthast, Johannes Kiesel, Kevin Reinartz, Janek Bevendorff, and Benno Stein. 2017. A Stylometric Inquiry into Hyperpartisan and Fake News. *CoRR* abs/1702.05638 (2017).

[14] Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. 2017. 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*. Copenhagen, Denmark.

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

[16] Natali Ruchansky, Sungyong Seo, and Yan Liu. 2017. CSI: A Hybrid Deep Model for Fake News Detection. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM '17)*. 797–806.

[17] Kai Shu, Deepak Mahudeswaran, Suhang Wang, Dongwon Lee, and Huan Liu. 2018. FakeNewsNet: A Data Repository with News Content, Social Context and Dynamic Information for Studying Fake News on Social Media. *CoRR* abs/1809.01286 (2018). arXiv:1809.01286 http://arxiv.org/abs/1809.01286

[18] Kai Shu, Suhang Wang, and Huan Liu. 2017. Exploiting Tri-Relationship for Fake News Detection. *CoRR* abs/1712.07709 (2017).

[19] James Thorne and Andreas Vlachos. 2018. Automated Fact Checking: Task Formulations, Methods and Future Directions. In *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, Santa Fe, New Mexico, USA, 3346–3359.

[20] Naftali Tishby and Noga Zaslavsky. 2015. Deep Learning and the Information Bottleneck Principle. *CoRR* abs/1503.02406 (2015).

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

[22] Yafang Wang, Gerard de Melo, and Gerhard Weikum. 2018. Five Shades of Untruth: Finer-Grained Classification of Fake News. *IEEE/ACM International Conference on ASONAM* (2018), 593–594.

[23] Zhe Zhao, Paul Resnick, and Qiaozhu Mei. 2015. 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)*. International World Wide Web Conference, Republic and Canton of Geneva, Switzerland, 1395–1405.