# Supporting verification of news articles with automated search for semantically similar articles

Vishwani Gupta[a], Katharina Beckh[a,b], Sven Giesselbach[a,b], Dennis Wegener[a] and Tim Wirtz[a,c]

[a]*Fraunhofer IAIS*
[b]*Competence Center for Machine Learning Rhine-Ruhr*
[c]*Fraunhofer Center for Machine Learnning*

## Abstract

Fake information poses one of the major threats for society in the 21st century. Identifying misinformation has become a key challenge due to the amount of fake news that is published daily. Yet, no approach is established that addresses the dynamics and versatility of *fake news editorials*. Instead of classifying content, we propose an evidence retrieval approach to handle fake news. The learning task is formulated as an unsupervised machine learning problem. For validation purpose, we provide the user with a set of news articles from *reliable news sources* supporting the hypothesis of the news article in query and the final decision is left to the user. Technically we propose a two-step process: (i) *Aggregation*-step: With information extracted from the given text we query for similar content from reliable news sources. (ii) *Refining*-step: We narrow the supporting evidence down by measuring the semantic distance of the text with the collection from step (i). The distance is calculated based on Word2Vec and the Word Mover's Distance. In our experiments, only content that is below a certain distance threshold is considered as supporting evidence. We find that our approach is agnostic to concept drifts, i.e. the machine learning task is independent of the hypotheses in a text. This makes it highly adaptable in times where fake news is as diverse as classical news is. Our pipeline offers the possibility for further analysis in the future, such as investigating bias and differences in news reporting.

## Keywords
fake news, document similarity, word mover's distance, news verification

## 1. Introduction

Although its negative influence and its weaponizing usage is known for ages [1], *fake news* (in non-political context also known as *false news* [2]) and its negative impact was globally recognized, during the U.S. elections in 2016, as one of the major challenges for the society of the 21st century [3, 4]. Importantly, it was used to promote both political campaigns in the election. Beside the promotion of political campaigns [5, 6], fake news occurs with various purposes or

due to various circumstances, e.g. to destabilize governments in third countries, accidentally due to unconscious misinterpretation of facts [7] or as a worthwhile revenue stream based on advertisement [8, 9, 10].

Following [7] the term *fake news* is used twice (a) to discredit and downgrade media and journalism; and (b) to summarize various forms of wrong, misguided, or fabricated information. Throughout this manuscript we are speaking about (b) when discussing fake news. Fake news articles, as just described, are to a large extend published, maintained, circulated and promoted in social media [11, 12, 13, 14]. On a high level, two strategies of potential interventions have been highlighted [15], (i) empowering of individuals to evaluate and assess fake news and (ii) structural changes preventing exposure of fake news to individuals. Most likely, machine learning based intervention strategies can be categorized into the second class of strategies. However, our intent is to propose an approach, while mainly based on state-of-the-art machine learning methodology, that can be categorized into the first class. Because we believe that the most sustainable strategy to fight the impact of fake news is to empower individuals to evaluate and assess fake news, we propose to assess content with evidence from reliable news sources supporting the hypotheses in the articles. We leave the final decision to the user which helps to improve acceptance because no actual censorship is happening. However, a quantitative statistical evaluation is still possible by simply adding a threshold from cross-validation experiments on top of the mechanism.

In summary our contribution can be structured into the following aspects:

- Modular system for the comparison of news articles from various sources.
- Unsupervised approach for verification of a queried article and its content.
- Automatic querying for supporting articles using News API.
- An intuitive user interface which allows to individualize the collection of reliable sources and to receive visual feedback for the queried article.

The outline of the paper is as follows: Section 2 summarizes the main related work, highlighting prior approaches towards verification. In section 3, we outline the relevant machine learning building blocks of our approach. Section 4 introduces our system for news verification and describes the workflow, architecture and user interface. The deployment of the solution architecture is described in section 5 as well as a discussion of the approach in general and its advantages in section 6. Finally, in section 7 we summarize the approach.

## 2. Related Work

Following the line of argumentation of [16], approaches to identify fake news can be structured into four major categories: knowledge-based, style-based, propagation-based and source-based Propagation-based analyses are concerned with how fake news spread online which is mostly formulated as a binary classification problem. The input can be either a news cascade [17] or a self-defined graph [18]. For style-based analysis, the writing style is assessed according to malicious intent. Perez et al. [19] point out stylistic biases that exists in text in order to automate fake news detection. Source-based approaches assess the credibility of a news source [20, 21] while knowledge-based approaches compare news content with known facts [22]. According to

the scheme from Zhou et al. [16], our proposed approach combines two categories of fake news detection: source-based and knowledge-based analysis. For both, we highlight prior work.

The most prevalent source-based approach is to rate news sources on their credibility. Traditional source-based approaches are Web ranking algorithms which rely on website credibility to improve search results for user queries [23]. Two current resources for news publisher credibility are MediaBias/FactCheck [24] and NewsGuard [20], a browser extension that displays ratings of news websites. The ratings are manually curated by journalists. Since the ratings go through a manual review process the list of rated websites is prone to be incomplete and quickly outdated. Therefore, recent efforts aim for automating source reliability ratings. Based on *expert*-features including for example web-traffic, the existence of a verified Twitter account or textual information, the authors of [21] classify the news sources in a supervised manner using a Support Vector Machine. Another approach to evaluate credibility of the knowledge is proposed by Esteves et al. [25]. The proposed approaches are based on supervised learning to automatically extract source reputation cues and to compute a credibility factor. A further approach that also taps into style-based methods is to analyse text and metadata in the article. Rashkin et al. [26] assess the reliability of entire news articles by predicting whether the document originates from a website classified as hoax, satire or propaganda by comparing the language of real news with those three categories to find linguistic characteristics of untrustworthy text. Wang et al. showed that significant improvements can be achieved for fine-grained fake news detection when meta-data is combined with text [27].

Knowledge-based approaches mostly tackle the process of fact-checking. Several fact-checking organizations such as CORRECTIV [28], PolitiFact [29] and Snopes [30] operate by manually verifying claims (see [16] for more expert-based fact-checking websites). A drawback of manual verification is that it may reach readers too late. An approach tackling this issue was recently published [22] where the authors approached fact-checking with machine learning methods and focuse on claim verification. Another related approach is presented in [31] where the authors introduce a Fact Extraction and VERification (FEVER) Shared Task. The aim is to classify whether a claim is factual or not by retrieving evidence from Wikipedia. Both works treat the task as a classification problem, and a critical challenge with this approach is that we can not guarantee that the system is able to give suggestions to very recent claims. An approach geared towards misinformation detection for social media treating exactly this challenge is presented in [32] by including a retrieval step. While the authors still include classification as a second step, i.e. for stance detection, we completely omit any supervised task and focus on retrieval and an expert-knowledge-based scoring.

## 3. Building Blocks of the Approach

As presented in more detail in section 4 we propose an evidence retrieval approach to handle fake news instead of classifying content. The learning task is formulated as an unsupervised machine learning problem. The evidence supporting the hypothesis of the queried article is gathered from a collection of reliable news sources which we provide to the user. It is individualized by selecting an arbitrary number of sources out of a curated list of reliable news sources for evidence-gathering purposes. Technically, we propose a two-step process:

1. *Aggregation*-step: Extract information from the given article and query for similar content from reliable sources
2. *Refining*-step: Narrow the supporting evidence down by calculating the semantic distance of the text with the collection that was retrieved in step 1.

In the following subsection we briefly introduce the most relevant machine learning concepts, forming the basis of the proposed approach. To calculate the semantic distance of news articles we rely on distributed word embeddings and the Word Mover's distance.

### 3.1. Word Embedding

Mikolov et al. [33] proposed the Word2Vec algorithm to learn vector representations of words. The method is based on the distributional hypothesis [34, 35] that words get their meaning from the context in which they appear. Mikolov et al. propose two different variations of the Word2Vec algorithm, both typically trained on large text corpora. The Continuous Bag-of-Words Model (CBOW) and the Continuous Skip-gram Model (skip-gram) which predict target words from source context words and source context words from target words, respectively. Specifically, they propose a shallow neural network architecture, which trains continuous word vectors representations to maximize the log probability of neighboring words in a corpus. For a given sequence of words $w_1, w_2, ..., w_N$, it models the probability of this particular sequence as follows

$$\frac{1}{N} \sum_{n=1}^{N} \sum_{j \in nb(n)} \log p(w_j \mid w_n) \tag{1}$$

Here, $nb(n)$ is the set of neighboring words of the word $w_n$. The unsupervised training is done by optimizing the maximum likelihood of a corpus of sentences (sequences of words) such that the word embeddings capture the semantic information of words and relations between them, given a particular context. In their original work [33], the authors approximated the objective above by more efficiently trainable objectives.

A flaw of Word2Vec is its inability to infer continuous representations for words not seen during training. Especially in domains such as news, new vocabulary can emerge rapidly. A simple way to account for that is to incorporate morphological information about words in the text representations. Bojanowski et al. [36] proposed *fastText*, an extension of the skip-gram model, which learns word representations by including sub-word information. This is achieved by not only representing words with vectors but also the sub-word parts they consist of, bag of character *n*-grams. Word vector representations are built as the sum of their sub-word and their own representation.

In this work, we experimented with two embedding models, Word2Vec and fastText embeddings. Although there are by far more than two approaches available in the literature (also more advanced approaches like Transformers [37]), see [38, 39] for comprehensive reviews, we focus on those because they can be efficiently implemented on standard hardware and are well-established in the NLP community. Nevertheless, there is freedom in experimenting with other word embeddings as it only requires a change of the distance threshold. Hence, the approach can be easily adapted to support news verification for different languages.

### 3.2. Word Mover's Distance

Earth mover's distance (EMD), also known as the Wasserstein distance, is a distance measure between two probability distribution. Kusner et al. [40] proposed a version of EMD applicable to language models, the Word mover's distance (WMD) which evaluates the distance between two documents represented in a continuous space using word embeddings such as the afore-mentioned Word2Vec and fastText embeddings. For any two documents *A* and *B*, WMD is defined as the minimum cost of transforming document *A* into document *B*. Each document is represented by the relative frequencies of its words relative to the total number of words of the document, i.e., for the *j*th word in the document,

$$d_{A,j} = count(j)/\mid A \mid \tag{2}$$

where $\mid A \mid$ is the total word count of document A and $count(j)$ is number of occurrences of the word with vocabulary index $j$. The $j$th word is represented by its corresponding word embedding, say $\mathbf{v}_j \in \mathbb{R}^n$. The $n$-dimensional word embeddings are obtained from a pre-trained model, e.g. Word2Vec or fastText. The distance between two words can easily be measured using Euclidean distance,

$$\delta(i,j) = \|\mathbf{v}_i - \mathbf{v}_j\| \tag{3}$$

Based on this choice, the Word mover's distance is defined to be the solution of the following linear program,

$$
\begin{aligned}
WMD(A,B) = \quad &\min_{\mathbf{T} \geq 0} \sum_{i=1}^{V} \sum_{j=1}^{V} \mathbf{T}_{i,j}\delta(i,j) \\
\text{such that} \quad &\sum_{i=1}^{V} \mathbf{T}_{i,j} = d_{A,j} \\
\text{and} \quad &\sum_{j=1}^{V} \mathbf{T}_{i,j} = d_{A,i}
\end{aligned}
\tag{4}
$$

Here, $\mathbf{T} \in \mathbb{R}^{V \times V}$ is a non-negative matrix, where $\mathbf{T}_{i,j}$ denotes how much of word *i* in document *A* is assigned to tokens of word *j* in document *B*. Empirically, WMD has reported improved performance on many real world classification tasks as demonstrated in [40]. The WMD has intriguing properties. The distance between two documents can be broken down and represented as the sparse distances between few individual words. The distance metric is also hyper-parameter free. The most important feature is that it incorporates the semantic information encoded in the word embedding space and is agnostic to arbitrary word embedding models.

## 4. A Retrieval-Based Approach Supporting Fake News Identification Methods

We constructed a pipelined system which helps in extracting semantically similar articles from reliable news sources. Its core is the analysis of the credibility of news articles based on the
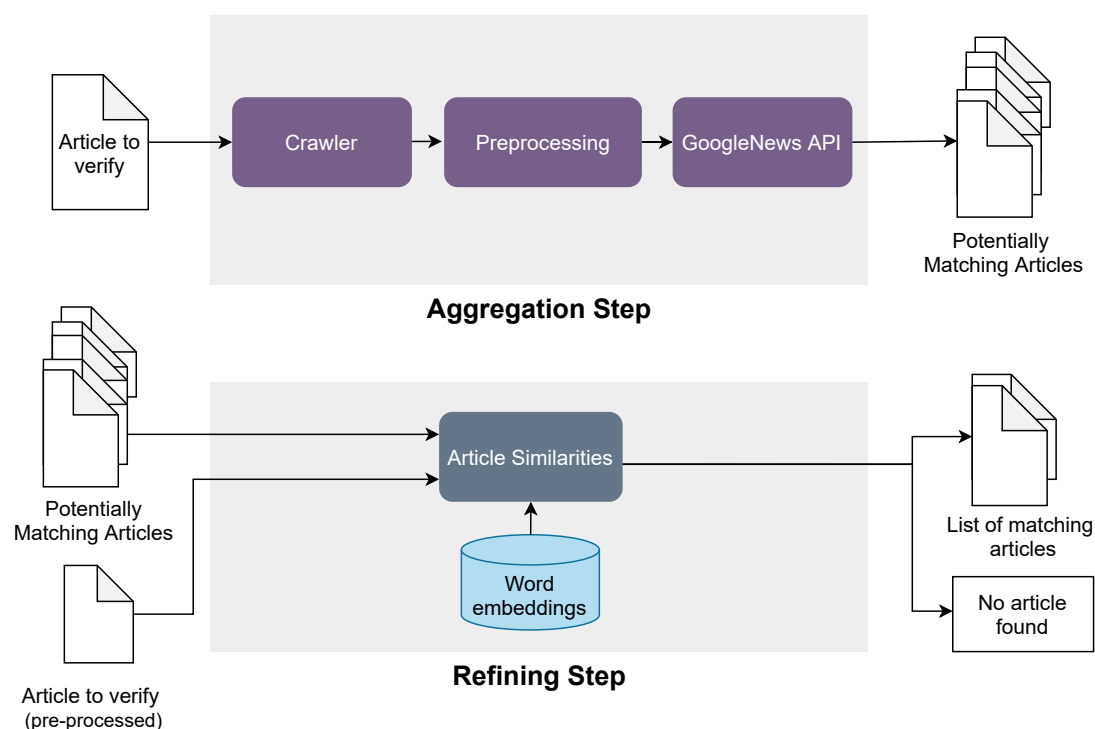
**Figure 1:** Overview of the system's workflow with aggregation and refining step.

overall evidence collected from a set of automatically retrieved articles published by reliable news sources. Figure 1 gives an overview of the components and workflow. The system consists of three components: a news content extractor, a search engine query and a content analyzer. All of these components can easily be exchanged and extended depending on the language and the list of reliable news sources.

## 4.1. Article Extractor

Given a link to an article that should be verified the article extractor component extracts information from the link such as the publication date, the article title, its authors and the textual content. For most of the news sources, the python library Newspaper3k[1] is suitable and manages to extract all of the above information. However, at least for some sources, we built our own article extractors by parsing the HTML page of the article and extracting certain tags. We extract relevant keywords and entities from the title and body of the article using the Newspaper API keyword extractor. The goal of this step is to get as much relevant information characterizing the article as possible. This extracted information is used in building an automated query.

---

[1]https://newspaper.readthedocs.io/en/latest/

## 4.2. Querying the Google News API

To obtain news articles from reliable news sources we use a query component which queries the Google News API. Using the keywords and entities we obtained in the step before, we construct a query. We structure the query so that we can filter the articles based on date, number of requested news articles from the sources, location and language. The API returns ten article links based on the search criterion from every source selected. The number of articles returned by the API can be changed based on individual requirements and computation power. The article extractor component extracts the content of the articles obtained from the search API. The system offers six news sources and we can easily add new sources or remove existing ones from the list. Automatic querying used here is different from manual news search using search engines as we aggregate news based on dates, keywords extracted from the article, and selected reliable sources.

## 4.3. Content Analysis: Semantic Distance Analysis

The content analysis component computes the semantic distance between the query article and the articles returned by the query component. Before computing the distance score, we clean the article titles and bodies by removing stop words and special symbols and computing their bag of n-grams representations. The semantic distance of articles is calculated using word embeddings and the WMD. For the word embeddings, we experimented with different word embeddings such as fastText and the pre-trained Google news embeddings. The quality of the word embeddings depends on the size of training data, thus, we use pre-trained word embeddings.

Since the original WMD is computationally expensive, we approximate the distance by using the Regularized Wasserstein distance proposed by [41] and only keep the five closest articles. The five articles with the least distance are then selected for computation with the original WMD. The WMD returns a distance score for each remaining article from the individual sources. The smaller the distance, the more related the articles are. Only articles that are below a predefined threshold are considered as similar to the given article. We set the distance threshold by empirically checking the distances of a couple of articles. Similar news articles, i.e. articles that fall below the distance threshold, are then displayed with a message that closely related articles were found. If the system does not return similar articles the reader is informed that the given article is potentially fake.

Our prototype was exemplary tested on a small set of articles. A systematic evaluation with a self-curated dataset and the FakeNewsNet [42] dataset is planned. The semantic distance analysis in our approach is based on unsupervised models which in turn make the system highly adaptable to different languages. We just need to replace the word embeddings and adapt the threshold. Furthermore, the unsupervised nature renders the approach agnostic to concept drifts which means that the machine learning task is independent of the hypotheses in a text.
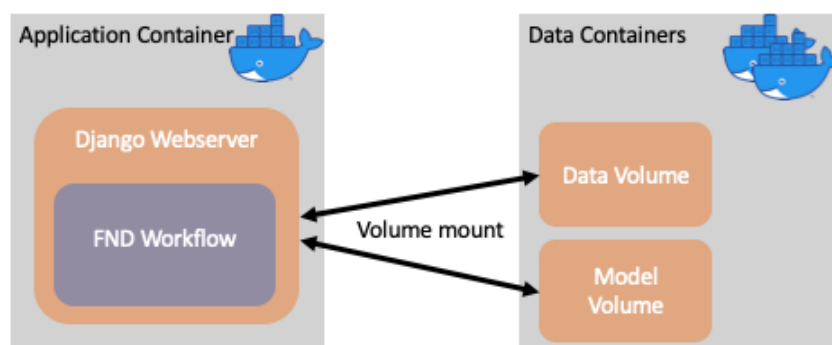
**Figure 2:** The technical architecture of the Fake News Detector with three docker components.

## 5. Architecture and Deployment

To showcase our approach we build a "Fake News Detector" system. The Fake News Detector system consists of a few technical components. Its technical architecture is based on a set of docker components (see Figure 2). In detail, there are the following three docker components:

1. A container with simple django running the python code of our application and serving the frontend.
2. A container serving the data for the backend - the model container.
3. A container that includes data pre-processed by several NLTK functions.

The Fake News Detector application can be accessed via web UI (see Figure 3). In the UI, we can insert a link of a news article to be verified, in this example, we want to verify an article titled *Gatorade banned and fined $300k for bad-mouthing water*. Next, we select the news sources to check and match against. By clicking on the verification button the analysis process is started at the backend running all the components. After the analysis, the results are shown as a list of potentially matching articles. If no matching articles are found after the analysis, a message is displayed that the article might be potentially fake. In the example shown in Figure 3, we selected all six sources. The system queried against all sources and analysed the potentially matching articles using semantic similarity and found that CNN has published a similar news article during the same time frame.

## 6. Discussion and Future Work

In the previous sections, we have addressed various benefits of following such an unsupervised approach. These benefits are the plasticity of the system, its modularity and user-driven decision making. In this section, we discuss several challenges and insights for the future work.

One challenge for the demonstrator is to deal with very recent news. The system will not be able to collect semantically similar articles from other reliable news sources that might not have published yet on the subject. Here, date as well as publishing time become important. For

**Figure 3:** The User Interface of the Fake News Detector. In this example, the query article is titled *Gatorade banned and fined $300k for bad-mouthing water*. The user pastes the URL into the text box, selects sources and receives a matching article from the source CNN.

future work, we propose to highlight this fact to the user to avoid perceiving recent news as potentially fake.

Another challenge in analyzing news articles is novel words which are out-of-vocabulary. For these words no word vectors exist which in turn affects the WMD score. The system can handle this scenario but the performance may be impaired. To overcome this we are crawling news from major news sources every hour to build a text dataset. We propose to update word embedding models frequently to avoid an increased number of out-of-vocabulary words in recent news articles.

Recently, Yokoi et al. [43], proposed an improvement on Word Mover's Distance, namely Word Rotator's Distance (WRD) which measures the degree of semantic overlap between two texts using word alignment. This approach is designed such that the norm (a proxy for the importance of word) and angle of word vectors (a proxy for dissimilarity between words) correspond to the probability mass and transportation cost in EMD, respectively. This approach outperforms the WMD in several semantic textual similarity tasks. This alternative might help

in improving the document distance threshold and is subject for our planned evaluation.

We also propose to use the approach of document similarity for related use cases where we see potential in two directions. The first direction is helping fact-checkers by providing adequate evidence to verify hypotheses. By providing them similar content, e.g. evidence in the form of news but also scientific articles, the system can support their task. Second, reviewers in several domains, e.g. medical health news review, need to determine how comprehensible a text document is. By comparing a text to scientific or more simple language it is possible to provide a comprehensibility score. Since the system alone does not guarantee news verification or falsification we recommend considering combining it with fact-checking methods.

## 7. Conclusion

We presented a system to find semantically similar articles to a given news article from selected reliable sources. For the system, we propose an evidence retrieval approach to handle fake news instead of treating it as a classification task. This way, we aid the users in finding supporting evidence and, thus, manual search work can be reduced. The benefits of our system are that (i) it is unsupervised and therefore agnostic to concept drifts, (ii) it gives the user decision power and (iii) it is modular, i.e. the system can be easily adapted to other languages, extended and improved with further components.

## Acknowledgments

## References

[1] The long and brutal history of fake news, https://www.politico.com/magazine/story/2016/12/fake-news-history-long-violent-214535, 2016.

[2] C. Wardle, H. Derakhshan, Information disorder: Toward an interdisciplinary framework for research and policy making, Council of Europe report 27 (2017).

[3] A. R. Activities, Intentions in recent us elections, Intelligence Community Assessment, Office of the Director of National Intelligence 6 (2017).

[4] L. Howell, et al., Digital wildfires in a hyperconnected world, WEF report 3 (2013) 15–94.

[5] D. Barstow, Behind tv analysts, pentagon's hidden hand, New York Times 20 (2008) A1.

[6] H. Allcott, M. Gentzkow, Social media and fake news in the 2016 election, Journal of economic perspectives 31 (2017) 211–36.

[7] T. Quandt, L. Frischlich, S. Boberg, T. Schatto-Eckrodt, Fake news, The international encyclopedia of Journalism Studies (2019) 1–6.

[8] M. M. Waldrop, News feature: The genuine problem of fake news, Proceedings of the National Academy of Sciences 114 (2017) 12631–12634.

[9] N. Kshetri, J. Voas, The economics of "fake news", IT Professional 19 (2017) 8–12.

[10] J. A. Braun, J. L. Eklund, Fake news, real money: Ad tech platforms, profit-driven hoaxes, and the business of journalism, Digital Journalism 7 (2019) 1–21.

[11] N. Wingfield, M. Isaac, K. Benner, Google and facebook take aim at fake news sites, The New York Times 11 (2016) 12.

[12] M. Koohikamali, A. Sidorova, Information re-sharing on social network sites in the age of fake news., Informing Science 20 (2017).

[13] A. Bovet, H. A. Makse, Influence of fake news in twitter during the 2016 us presidential election, Nature communications 10 (2019) 1–14.

[14] K. Shu, H. R. Bernard, H. Liu, Studying fake news via network analysis: detection and mitigation, in: Emerging Research Challenges and Opportunities in Computational Social Network Analysis and Mining, Springer, 2019, pp. 43–65.

[15] D. M. Lazer, M. A. Baum, Y. Benkler, A. J. Berinsky, K. M. Greenhill, F. Menczer, M. J. Metzger, B. Nyhan, G. Pennycook, D. Rothschild, et al., The science of fake news, Science 359 (2018) 1094–1096.

[16] X. Zhou, R. Zafarani, A survey of fake news: Fundamental theories, detection methods, and opportunities, ACM Computing Surveys (CSUR) 53 (2020) 1–40.

[17] C. Castillo, M. Mendoza, B. Poblete, Information credibility on twitter, in: Proceedings of the 20th international conference on World wide web, 2011, pp. 675–684.

[18] Z. Jin, J. Cao, Y. Zhang, J. Luo, News verification by exploiting conflicting social viewpoints in microblogs, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 30, 2016.

[19] V. Pérez-Rosas, B. Kleinberg, A. Lefevre, R. Mihalcea, Automatic detection of fake news, in: Proceedings of the 27th International Conference on Computational Linguistics, Association for Computational Linguistics, 2018.

[20] Newsguard: The internet trust tool, https://www.newsguardtech.com, Accessed: 20.12.2020.

[21] R. Baly, G. Karadzhov, D. Alexandrov, J. Glass, P. Nakov, Predicting factuality of reporting and bias of news media sources, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 3528–3539. URL: https://www.aclweb.org/anthology/D18-1389. doi:10.18653/v1/D18-1389.

[22] B. Botnevik, E. Sakariassen, V. Setty, Brenda: Browser extension for fake news detection, in: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '20, Association for Computing Machinery, New York, NY, USA, 2020, p. 2117–2120.

[23] L. Page, S. Brin, R. Motwani, T. Winograd, The PageRank citation ranking: Bringing order to the web., Technical Report, Stanford InfoLab, 1999.

[24] Mediabiasfactcheck, https://mediabiasfactcheck.com/, Accessed: 26.02.2021.

[25] D. Esteves, A. J. Reddy, P. Chawla, J. Lehmann, Belittling the source: Trustworthiness indicators to obfuscate fake news on the web, in: Proceedings of the First Workshop on Fact Extraction and VERification (FEVER), Association for Computational Linguistics, 2018.

[26] H. Rashkin, E. Choi, J. Y. Jang, S. Volkova, Y. 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, Association for Computational Linguistics, 2017.

[27] W. Y. Wang, "liar, liar pants on fire": A new benchmark dataset for fake news detection, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, 2017.

[28] Correctiv: Investigations in the public interest, https://correctiv.org, Accessed: 20.12.2020.

[29] T. P. Institute, Politifact, https://www.politifact.com, Accessed: 20.12.2020.

[30] Snopes, https://www.snopes.com, Accessed: 20.12.2020.

[31] J. Thorne, A. Vlachos, O. Cocarascu, C. Christodoulopoulos, A. Mittal, The fact extraction and VERification (FEVER) shared task, in: Proceedings of the First Workshop on Fact Extraction and VERification (FEVER), Association for Computational Linguistics, 2018, pp. 1–9.

[32] T. Hossain, R. L. Logan IV, A. Ugarte, Y. Matsubara, S. Young, S. Singh, COVIDLies: Detecting COVID-19 misinformation on social media, in: Proceedings of the 1st Workshop on NLP for COVID-19 (Part 2) at EMNLP 2020, Association for Computational Linguistics, Online, 2020.

[33] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, in: Y. Bengio, Y. LeCun (Eds.), 1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings, 2013.

[34] Z. Harris, Distributional structure, Word 10 (1954) 146–162. URL: https://link.springer.com/chapter/10.1007/978-94-009-8467-7_1. doi:10.1007/978-94-009-8467-7_1.

[35] J. R. Firth, Papers in Linguistics, 1934-1951, Oxford University Press, London, 1957.

[36] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information, Transactions of the Association for Computational Linguistics (2017) 135–146.

[37] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in neural information processing systems, 2017, pp. 5998–6008.

[38] Y. Li, T. Yang, Word embedding for understanding natural language: a survey, in: Guide to Big Data Applications, Springer, 2018, pp. 83–104.

[39] F. Almeida, G. Xexéo, Word embeddings: A survey, arXiv preprint arXiv:1901.09069 (2019).

[40] M. Kusner, Y. Sun, N. Kolkin, K. Weinberger, From word embeddings to document distances, in: International conference on machine learning, 2015, pp. 957–966.

[41] G. Balikas, C. Laclau, I. Redko, M.-R. Amini, Cross-lingual document retrieval using regularized wasserstein distance, in: Proceedings of the 40th European Conference ECIR conference on Information Retrieval, ECIR 2018, Grenoble, France, March 26-29, 2018, 2018.

[42] K. Shu, D. Mahudeswaran, S. Wang, D. Lee, H. Liu, Fakenewsnet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media, Big Data (2020).

[43] S. Yokoi, R. Takahashi, R. Akama, J. Suzuki, K. Inui, Word rotator's distance, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020, pp. 2944–2960.