Multilingual Fake News Detection with Satire

Gaël Guibon, Liana Ermakova, Hosni Seffih, Anton Firsov, Guillaume Le Noé-Bienvenu

Multilingual Fake News Detection with Satire

Gaël Guibon[1], Liana Ermakova[2], Hosni Seffih[3,4], Anton Firsov[5], and Guillaume Le Noé-Bienvenu[6]

[1] Aix-Marseille Université, LIS-CNRS UMR 7020, France
[2] HCTI – EA 4249, Université de Bretagne Occidentale, Brest, France
[3] IUT de Montreuil, Université Paris 8, LIASD-EA4383, France
[4] GeolSemantics
[5] Knoema Corporation Perm, Russia https://knoema.com/
[6] PluriTAL, France

Abstract. The information spread through the Web influences politics, stock markets, public health, people's reputation and brands. For these reasons, it is crucial to filter out false information. In this paper, we compare different automatic approaches for fake news detection based on statistical text analysis on the vaccination fake news dataset provided by the Storyzy company. Our CNN works better for discrimination of the larger classes (fake vs trusted) while the gradient boosting decision tree with feature stacking approach obtained better results for satire detection. We contribute by showing that efficient satire detection can be achieved using merged embeddings and a specific model, at the cost of larger classes. We also contribute by merging redundant information on purpose in order to better predict satire news from fake news and trusted news.

Keywords: fake news, deception, artificial intelligence, machine learning, natural language processing, satire

1 Introduction

According to the survey performed by the French journal *Le Monde*, on March 2017 the 14-15, over 12,000 Frenchmen 66% of the French population were going to vote, 41% among them had not made their choice yet [21]. According to the data of *Médiamétrie*, a French audience measurement company, more than 26 million of French people connected to social networks to read and share articles and posts in February 2017. Therefore Web information influences campaigns. *Fake news* and *post-truth* become more and more destabilizing and widespread (e.g., the 2016 United States presidential election [12], Brexit [11]). The information spread on the Web can influence not only politics, but also stock markets. In 2013, $130 billion in stock value was lost in a few minutes after an AP tweeted that Barack Obama had been injured by an "explosion" [24].

Fake news detection or rumor detection, is a concept that started in early 2010, as social media started to have a huge impact on people's views. Different approaches have been used throughout the years to detect fake news. They can be divided into two categories: manual and automatic ones. Facebook decided to manually analyze the content after a certain number of users have signalled the doubtful information [26]. Merrimack College published a blacklist of web sites providing fake information [16]. This list was integrated into a Google Chrome extension [13]. Numerous Web sites and blogs (e.g. Acrimed, HoaxBuster, Cross-Check, « démonte rumeur » of Rue89) are designed for fact verification. For example, the web site FactCheck.org proposes to a reader to verify sources, author, date and title of a publication [2]. Pariser started a crowdsourcing initiative "Design Solutions for Fake News" aimed at classifying mass media [23]. However, the methods based on manual analysis are often criticized for insufficient control, expertise requirements, cost in terms of time and money [27]. This system needs human involvement; the source is flagged unreliable by the community, as in BS detector, or by specialists, as in Politi-Fact[4].

On the other side, automatic methods are not widely used. The preference of manual approaches over the automatic methods of the large innovation companies like Facebook is indirect evidence of the lower quality of the existing automatic approaches. According to [9], automatic methods of fake news detection are based on linguistic analysis (lexical, syntactical, semantical, discourse analysis by the means of the Rhetorical Structure Theory, opinion mining) or network study. Various Natural Language Processing (NLP) and classification techniques help achieve maximum accuracy [14]. Criteria-Based Content Analysis, Reality Monitoring, Scientific Content Analysis and Interpersonal Deception Theory provide some keys for the detection of textual fake information [30]. In [30], the authors treated textual features of deception in a dialogue. Despite their work being not applicable for fake news detection since they are monologues, it gives some interesting perspectives. In [5], authors analyzed the non-verbal visual features. Castillo[8] used a feature-based method to define tweets' credibility. Ma[20] extracted useful features to detect fake news. The last two methods provided satisfying results. Ma[19] used Long-Short Term Memory networks (LSTM) to predict if a stream of tweets modeled as sequential data were rumors. This approach was more successful than the feature-based one. Rashkin[25] took a linguistic approach to detect fake news, examining lexicon distribution and discovering the difference between the language of fake news and trusted news, but the result showed only 22% accuracy. Volkova[28] tried to classify tweets into suspicious news, satire, hoaxes, clickbait and propaganda using linguistic neutral networks with linguistic features. It turned out that syntax and grammar features have little effect. Pomerleau and Rao created Fake News Challenge aimed at the detection of the incoherence between the title and the content of an article [3], while the task proposed by DiscoverText is targeted at the identification of fake tweets [1].

In this paper, we compare several classification methods based on text analysis on the dataset provided by the start-up specializing in fake news detection Storyzy[7]. This dataset was used a Fake News detection task during the french hckathon on Natural Language Processing[8]. We compare our work to other teams' work, especially on the satire detection. During the hackathon, different teams tried to obtain good results in distinguishing fake news from trusted news, in which we ranked second. However, we obtained by far the best scores in satire detection which is the main contribution of this paper. The second contribution is the usage of merged embeddings and lexical features containing redundancy for improved satire detection.

2 Experimental framework and results

We performed our evaluation on the dataset provided by the company Storyzy for a hackathon in *.tsv* format. Storyzy specializes in brand image protection by detection of the suspicious content on websites that can potentially show the advertisements of the brands. The corpus contains texts from various websites in English and French, as well as automatic transcripts of YouTube French videos about vaccination which is a widely disseminated topic in false news. The details are given in Table 1.

The task is to classify the textual content into 3 possible classes: *Fake News*, *Trusted*, or *Satire*. We compared our results with those of the participants of the hackathon on fake news detection. The main hackathon's task was to obtain the best score on the first two classes. However, we consider the satire class as the most interesting and challenging. This is why we have tried to effectively classify all texts including satire. To do so, we used an experimental approach, focusing on the impact of data representation to find the best way to classify text in English, transcribed French, and French.

Table 1: Corpus statistics

Language	Train	Test	Train format	Test format
English	3828	1277	id, domain, type, uri, author, language, title, text, date, external_uris	id, title, text
French	705	236	id, domain, type, uri, author, language, title, text, date, external_uris	id, title, text
YouTube	234	78	video-id, channel-id, video-title, video-view-count, lang, type, channel-title, text, id	id, video-title, text

[7] https://storyzy.com/?lang=en

[8] HackaTAL : https://hackatal.github.io/2018/

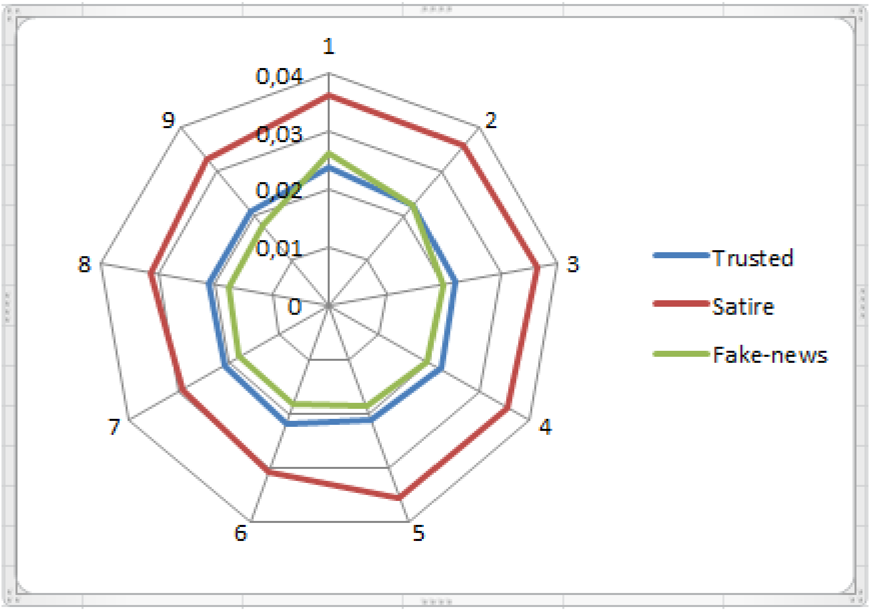

Fig. 1: Text resemblance scores. The closer they are, the more difficult they are to discriminate.

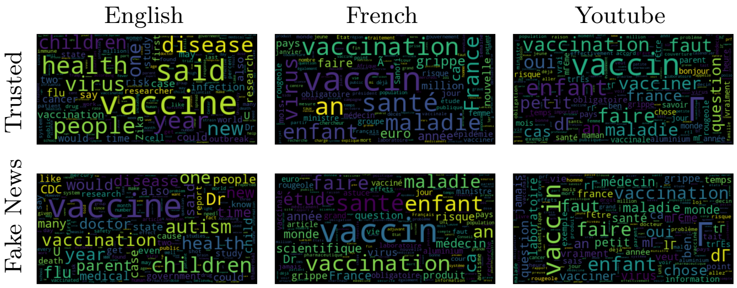

Fig. 2: Word clouds for each dataset.

2.1 Text resemblance

We searched in the Web snapshot using chatNoir API[6] for news titles and we compared the first ten results with the news texts after the tokenization and lemmatization using NLTK[9] for the English texts, French texts were preprocessed by using a tokenizer that splits texts by white spaces and apostrophes, deletes stop words. After that, texts are lemmatized by the NLTK FrenchStemmer. Then for each pair of query/article, we got a “resemblance ratio” using

[9] https://www.nltk.org/

difflib SequenceMatcher[10]. The difflib module contains tools for computing the differences between sequences and working with them. It is especially useful for comparing texts, and includes functions that produce reports using several common different formats. That way we created a ratio going from 0 to 1 describing how similar the resulting text is to the original, 1 being the exact same text. By applying this method on the train corpus, then calculating an average for all the trusted news, all the fake news and all the satire news results, we were able to obtain Figure 2 which also presents word clouds for different classes from the train set. Figure 2 testifies that trusted news have approximately the same ratio with each of the query texts and the news text (0.021–0.024). On the other hand, even if the fake news is relatively similar to the news text we get irregular ratios (0.018–0.026). The ratios of the satire query go from 0.029 to 0.037. Between query sites and news texts, there are small resemblances but more irregularities. We can also observe that some subjects are more frequent in satire (e.g. *Facebook*) or fake news (e.g. *autism, aluminum,* or *mercury* within the vaccination topic) than in trusted sources.

2.2 Domain type detection

Using chatNoir API to search news titles in the web snapshot, we obtained the first ten domain names and tested whether they belong to satire, fake, or trusted websites, using a predefined list in which we had a list of famous websites tagged as trusted fake or satire. To establish the list we used the domains on the train corpus and added lists we found on the Internet.

Our feature is composed of three parameters (Fake Site, Satire Site, Trusted Site). Each parameter is initialized to 0; it turns to 1 if the domain is found on the list. The result of the train corpus is on Table 2. The first observation is that 74% of the fake news title searches lead to fake news websites, 4% to satire websites, and 22% to trusted websites. This 22% rate is due to well-written titles, resembling trusted news titles, but once we go into the text we notice the "fakeness" of the news. Satire news lead toward fake news websites 83% of the times. This is probably because satire is written in the same way as trusted news. Therefore, we need to detect sarcasm to understand they are fake, hence satire gets reported by fake news websites as trusted news.

Table 2: Returned domain type for each news

	Fake Site		Satire Site		Trusted Site	
	Elements	Percentage	Elements	Percentage	Elements	Percentage
Fake News	484	74%	25	4%	146	22%
Satire	20	83%	0	0%	4	17%
Trusted	651	50%	116	9%	528	41%

[10] https://docs.python.org/2/library/difflib.html

The biggest problem is that trusted news are present with 50% frequency in fake sites, and 9% in satire, since fake news websites usually begin with trusted news and then change some facts and use a sort of “p-hacking” to make it match their fake message.

2.3 Classification Results

In order to obtain a classifier capable of better generalizing its predictions, we tried different data representations:

TF-IDF (tf). First we used a common vectorization of term frequency and inverse term frequency in the documents (TF-IDF). We first limited the dimensions arbitrarily to 300 and 12,000 in order to keep the most frequent terms from the vocabulary dictionary.

FastText (ft). We trained the FastText [7] model on the training set in order to avoid using external data such as already trained embeddings. Using up to bi-grams we obtained vectors of dimension 100. The training was configured with 5-iteration process, and ignored words with less than 5 occurrences in the corpus.

Word2Vec (wv). We also trained a skipgram model of Word2Vec [22] with a batch of 32 words, 20 iterations, and 300 dimension vectors. Words with less than 5 occurrences were also ignored.

Hashing Trick (hv). Additionally, we used the hashing trick [29] to obtain another data representation, normalized with a L2 regularization [15]. The vectors dimension was limited to 300.

All these data representations were applied in several classifiers in order to discriminate fake news from the trusted and satire ones: a Support Vector Machine classifier [10] (SVM) and Light Gradient Boosting Machine[11] [17] (LGBM) a state-of-the-art version of Gradient Boosting Decision Tree.

SVM were chosen as it was this method which obtained the best micro F1-score during the hackathon. Tree Boosting was decided in order to obtain better overall results with more insight on their explanations.

We also tried several neural network architectures (LSTM, CNN) applied for (1) characters represented by a number; (2) character embeddings; (3) word embeddings. The first representation prevents the neural network from training since characters with similar numbers are interpreted as almost the same by the network. French+YouTube datasets are not enough to train LSTM on char embeddings. We obtained both very high bias and variance (almost random). Among the deep learning approaches the best results we obtained were with the following architecture: *Embedding(64)* →*Conv1D(512,2,ReLu,dropout=0.4)* → *GlobalMaxPooling1D* →*Dense(3,SoftMax)* (**CNN**). As previously said, this architecture was selected after trying deeper ones. We think the size of the dataset played an important role in the number of layers to be applied in the network.

[11] https://github.com/Microsoft/LightGBM

Table 3: Classification schemes summary. FastText (ft). Word2Vec (wv). HackingTrick (hv). Domain Type (dt). Text Resemblance (tr).

Data Representation	**F1 micro**	**F1 macro**	**Cross Validation (f1 micro)**
Decision Tree (J48)			
Text Resemblance (tr)	59.79	61.34	58.33
Domain Type (dt)	58.63	60.15	57.20
Hackathon Winner (SVM)			
tf12K	94.28	82.25	/
Our LBGM during hackathon			
tf300+ft+wv	88.56	88.56	84.76
Optimized LGBM			
tf12K+ft+wv+hv	91.39	**91.87**	87.02
tf12K+ft+wv+dt+tr	92.01	91.36	87.52
Optimized Linear SVM			
tf12K+ft+wv+hv	93.02	91.19	**88.34**
tf12K+ft+wv+dt+tr	93.09	90.13	88.37
CNN			
/	**94.59**	86.02	/

Table 3 shows the different approaches we used to better detect fake news along with the one from the winner team during the hackathon organized by Storyzy. All these scores are based on micro and macro f1-score and cross validation (CV). In this table, the first two sections show the performance of fake news detection by applying the text resemblance and domain type detection methods described in Section 2. The other sections present the classification scores obtained with SVM and gradient boosting tree. Our systems were first tuned based on the cross validation micro f1-score from the training set. A grid search strategy was then used in order to find the best parameters for the optimized version of our LGBM and SVM classifiers. Grid search parameters were not exhaustive. For the LGBM classifier we tested boosting types such as GOSS and DART, number of leaves, different learning rates and number of estimators along with Lasso and Bridge regularizations values. On the other hand, SVM were tested with different kernels. The cross validation scores are micro f1 scores from a 5-fold stratified cross validation in order to compare directly the results of each method. The lower values of CV can be explained by this stratification strategy when classes are presented in the same proportion. Thus, the errors in satire detection influence a lot the macro scores. Table 3 also presents different data representation. In order to enhance the quality of the overall prediction of the model (macro f1-score) but without losing too much of its more logical prediction (micro f1-score), we used a stacking approach for data representation. Each acronym in the table refers to a representation strategy and the size of its vectors if multiple ones were used. Thus, the final data representation used named 'tf12K+ft+wv+hv+dt+tr' refers to a tf-idf vectorization (12,000) concatenated on the horizontal axis with FastText vectors (100), Word2Vec skipgrams vectors (300), vectors from the hashing trick (300), domain type (3), and text resemblance (9). At the end, every document was represented by a global vector of dimension 12,712 in which some information was obviously duplicated.

Merging embeddings. Embeddings do not always capture the same kind of information, for instance FastText uses hierarchical softmax and hashing trick for bi-grams allowing the model to capture local order into the words representations while a skip-gram embedding model captures a surrounding context into the words representations. This explains why combining them can prove itself worthy, especially for information difficult to obtain, such as satire.

2.4 Result Analysis

The best micro f1-score was achieved by CNN. However, optimized LGBM system performs better in macro f1-score. Detailed results on each class can be seen in Table 4. It shows the performance differences across each class and how the LGBM classifier better predicts the sparser cases of satire, hence leading to a better overall macro f1-score. Hence, CNN seems to be more suitable for discrimination between larger classes (fake news vs trusted ones), while boosted decision trees can better handle a fine-grained classes, such as satire.

Table 4: **Per class f1-scores**

	Trusted	**Fake News**	**Satire**
Hackathon Winner	**95.72**	**92.71**	58.33
Optimized LGBM	**92.96**	**88.89**	**93.75**
Optimized SVM	94.46	90.86	88.24
CNN	95.78	93.33	68.96

Feature Selection VS Macro F1-score. Our stacking approach for data representation was made in order to select features before training the classifier. To do so, we applied different kinds of feature selection to find the k best features, such as a chi-squared selection, mutual information [18], and ANOVA F-value. However, we found that applying feature selection significantly decreased the macro f1-score, even if it allowed a faster training process and almost the same micro f1-score. Indeed, all the stacked up minor information was really important to identify satire documents. This is why, we did not apply feature selection at all for our optimized classifiers. We believe that considering the decision tree nature by association with importance scores to each features, an inherent feature selection is made during the boosting process. This could explain why using a brute force approach of features stacking works better than a feature selection made before hand.

Scaling VS Stacking. Usually, vectors have to be scaled around 0 for SVM or to be set to a minimum and maximum range between 0 or 1 for decision trees to gain performance. However, the same conclusion as for feature selection was observed. Even more, scaling stacked up vectors could be quite inappropriate as they represent different information types obtained by different methods. Thus, the data was not scaled at all, to better preserve information gain.

Bi-lingual classification. The corpus is composed of texts in English (YouTube included) and French, the sources also varies from websites to transcribed text from videos. This particularity encouraged participants to create a system easily adaptable to any language given a training set, or even obtain a language agnostic system. We chose the first option in order to take into account lexical features and see their impact on the final classification.

Comparison with related work. As stated in the second part of the introduction (Section 1, several works on fake news automatic detection were done. Our work described in this paper is different from credibility analysis [8] because of the satire detection, which is more subtle than a credibility score. Indeed, satire is credible on its own way, only the nature of the information differs as the purpose of the writer is not to be taken seriously. Rashkin [25] used LSTM to detect the intents and degrees of truth from a range of 6 classes. Although their corpus contained some satire news, they did not try to precisely detect them, this is why we cannot fully compare to their work. Moreover, we did not use a scaled scores but only news types classification on 3 classes: fake, trusted, satire.

With our models, we show that LGBM can be really efficient to detect satire if they are combined with a good representation of the data, and can surpass CNN models on this particular task, but not on the classical Trusted/Fake classification.

3 Conclusion

The false information traveling through the Web impacts politics, stock markets, reputation of people and brands, and can even damage public health making filtering it out a burning issue of the day. In this paper, we compared different automatic approaches for fake news detection based on statistical text analysis on the dataset about vaccination provided by the startup Storyzy.

Text resemblance and domain type detection alone are not able to handle all the ambiguity. To have a much better vision between Fake/Trust and the Satire, we needed to combine some classification methods, such as text mining ones among others. The feature-based methods are open to improvement by adding more sites to the fake, satire, and trusted list, and by changing the resemblance ratio method for a much more powerful one. LGBM classifier better predicts the sparser cases of satire while neural networks outperform on bigger corpora, but only for the fake and trusted classification. Mapping a text into a vector of hash of characters or words is not appropriate for deep learning methods. One-hot representation and embeddings provide better results. On the small datasets, LSTM is not effective. Smaller CNN provide higher results than more complicated networks on this corpus.

Finally, we showed that good satire detection can be achieved automatically when combining Gradient Boosting Decision Trees with merged embeddings of different types. We also showed that the redundancy implied by these different embeddings help the classifier to better detect satire news from the fake and trusted news.

References

1. Fake News Student Twitter Data Challenge | DiscoverText (Dec 2016), http://discovertext.com/2016/12/28/fake-news-detection-a-twitter-data-challenge-for-students/
2. How to Spot Fake News (Nov 2016), http://www.factcheck.org/2016/11/how-to-spot-fake-news/
3. Fake News Challenge (2017), http://www.fakenewschallenge.org/
4. Adair, B.: Principles of politifact and the truth-o-meter. PolitiFact. com. February **21**, 2011 (2011)
5. Atanasova, M., Comita, P., Melina, S., Stoyanova, M.: Automatic Detection of Deception. Non-verbal communication (2014), https://nvc.uvt.nl/pdf/7.pdf
6. Bevendorff, J., Stein, B., Hagen, M., Potthast, M.: Elastic chatnoir: Search engine for the clueweb and the common crawl. In: Pasi, G., Piwowarski, B., Azzopardi, L., Hanbury, A. (eds.) Advances in Information Retrieval. pp. 820–824. Springer International Publishing, Cham (2018)
7. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics **5**, 135–146 (2017)
8. Castillo, C., Mendoza, M., Poblete, B.: Information credibility on twitter. In: Proceedings of the 20th international conference on World wide web. pp. 675–684. ACM (2011)
9. Conroy, N., Rubin, V., Chen, Y.: Automatic Deception Detection: Methods for Finding Fake News (2015)
10. Cortes, C., Vapnik, V.: Support-vector networks. Machine learning **20**(3), 273–297 (1995)
11. Deacon, M.: In a world of post-truth politics, Andrea Leadsom will make the perfect PM (Sep 2016), http://www.telegraph.co.uk/news/2016/07/09/in-a-world-of-post-truth-politics-andrea-leadsom-will-make-the-p/
12. Egan, T.: The post-truth presidency (Apr 2016), http://www.nytimes.com/2016/11/04/opinion/campaign-stops/the-post-truth-presidency.html
13. Feldman, B.: Here's a Chrome Extension That Will Flag Fake-News Sites for You (2016), http://nymag.com/selectall/2016/11/heres-a-browser-extension-that-will-flag-fake-news-sites.html
14. Gahirwal, M., Moghe, S., Kulkarni, T., Khakhar, D., Bhatia, J.: Fake news detection. International Journal of Advance Research, Ideas and Innovations in Technology **4**(1), 817–819 (2018)
15. Hoerl, A., Kennard, R.: Ridge regression: Biased estimation for nonorthogonal problems. Technometrics **12**(1), 55–67 (1970)

16. Hunt, E.: What is fake news? How to spot it and what you can do to stop it. The Guardian (Dec 2016), https://www.theguardian.com/media/2016/dec/18/what-is-fake-news-pizzagate
17. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.Y.: Lightgbm: A highly efficient gradient boosting decision tree. In: Advances in Neural Information Processing Systems. pp. 3149–3157 (2017)
18. Kraskov, A., Stögbauer, H., Grassberger, P.: Estimating mutual information. Physical review E **69**(6), 066138 (2004)
19. Ma, J., Gao, W., Mitra, P., Kwon, S., Jansen, B., Wong, K.F., Cha, M.: Detecting rumors from microblogs with recurrent neural networks. In: IJCAI. pp. 3818–3824 (2016)
20. Ma, J., Gao, W., Wei, Z., Lu, Y., Wong, K.F.: Detect rumors using time series of social context information on microblogging websites. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. pp. 1751–1754. ACM (2015)
21. Mandonnet, E., Paquette, E.: Les candidats face aux intox de Web. L'Express (3429), 28–31 (2017)
22. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
23. Morris, D.: Eli Pariser's Crowdsourced Brain Trust Is Tackling Fake News | Fortune.com (2016), http://fortune.com/2016/11/27/eli-pariser-fake-news-brain-trust/
24. Rapoza, K.: Can 'Fake News' Impact The Stock Market? (Feb 2017), https://www.forbes.com/sites/kenrapoza/2017/02/26/can-fake-news-impact-the-stock-market/
25. Rashkin, H., Choi, E., Jang, J., Volkova, S., Choi, Y.: 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. pp. 2931–2937 (2017)
26. Solon, O., Wong, J.: Facebook's plan to tackle fake news raises questions over limitations. The Guardian (Dec 2016), https://www.theguardian.com/technology/2016/dec/16/facebook-fake-news-system-problems-fact-checking
27. Twyman, N., Proudfoot, J., Schuetzler, R., Elkins, A., Derrick, D.: Robustness of Multiple Indicators in Automated Screening Systems for Deception Detection. J. of Management Information Systems **32**(4), 215–245 (2015), http://www.jmis-web.org/articles/1273
28. Volkova, S., Shaffer, K., Jang, J., Hodas, N.: Separating facts from fiction: Linguistic models to classify suspicious and trusted news posts on twitter. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). vol. 2, pp. 647–653 (2017)
29. Weinberger, K., Dasgupta, A., Langford, J., Smola, A., Attenberg, J.: Feature hashing for large scale multitask learning. In: Proceedings of the 26th Annual International Conference on Machine Learning. pp. 1113–1120. ACM (2009)
30. Zhou, L., Burgoon, J., Nunamaker, J., Twitchell, D.: Automating Linguistics-Based Cues for Detecting Deception in Text-based Asynchronous Computer-Mediated Communication. Group Decision and Negotiation (13), 81–106 (2004)