Classifying Arabic Tweets Based on Credibility Using Content and User Features

Ghaith Jardaneh
Information & Computer Science Dept.
An-Najah National University
Nablus, Palestine
ghaithjardane@gmail.com

Hamed Abdelhaq*
Information & Computer Science Dept.
An-Najah National University
Nablus, Palestine
hamed@najah.edu

Momen Buzz
Information & Computer Science Dept.
An-Najah National University
Nablus, Palestine
momenbuzzsd@gmail.com

Douglas Johnson
Department of Physics
University of Colorado, Boulder
Colorado, USA
douglas.johnson@colorado.edu

***Abstract*—**

Social Media services, such as Facebook and Twitter, have recently become a huge and continuous source of daily news. People all around the world rely heavily on news published via social media to know more about current events and activities. As a result, many users have started to exploit social media by broadcasting misleading news for financial and political purposes, which has an adverse impact on society. In this paper, we utilize machine learning to identify fake news from Arabic tweets based on a supervised classification model.

Twitter content published in Arabic is very noisy with a high level of uncertainty, where little work has been accomplished to process and extract important features for classification purposes. In this paper, we utilize content- and user-related features, and employ sentiment analysis to generate new features for the detection of fake Arabic news. Sentiment analysis led to improving the accuracy of the prediction process. Among a number of machine learning algorithms used to train the classification models, four algorithms are chosen, namely Random Forest, Decision Tree, AdaBoost, and Logistic Regression. The experimental evaluation shows that our system can filter out fake news with an accuracy of 76%.

***Index Terms*—Credibility Analysis; Fake News Detection; Natural Language Processing; Arabic Sentiment Analysis**

I. INTRODUCTION

Social media networks have emerged as platforms for communication and information sharing. These platforms serve hundreds of millions of users and provide many services such as content creation and publishing. Social media provides a major source of news read each day by many people. For instance, over two thirds of Americans consume their daily news from social media [8]. Twitter has become a popular platform for news sharing. It is one of the primal social media sites with over 71% of the news-focused users [8].

Not all published news on Twitter is reliable and accurate. Many people try to publish fake and erroneous news in order to influence public opinion. In addition, Twitter is known for its less-restrictive policy towards bots [2]. A bot is an account that is controlled via software using the Twitter API and can perform any task real accounts can, ranging from following and liking to direct messaging and replying. This results in opening up the opportunity for the spread of fake news[1].

Usually, fake news is generated for financial and political purposes, e.g., manipulating the stock market or influencing presidential elections. This can bring significant effects to society. For example, fake news played a major role in helping Donald Trump win the 2016 U.S. presidential election [6]. This argument is supported with verifiable evidence that viral fake election stories outperformed genuine stories [14].

Detecting fake news on social media can be a tedious task for many reasons: (1) it is extremely time-consuming to validate suspicious news and to look for verified evidences, (2) the lack of reliable sources since much of the fake news is based on real-time events, and (3) the content of the posts from social media are normally very noisy without enough information to easily verify their credibility.

Although a number of studies have been conducted to identify tweets with English language as fake news, assessing the credibility of tweets written in Arabic is still in its infancy. In this paper, we build a machine learning model aimed at quantifying the credibility of Arabic tweets with the goal of detecting and eliminating fake news.

In Figure 1, we illustrate the components comprising our tweet classification system. These components generate the model and learn its parameters from previously annotated tweets in order to predict the class ("credible" or "non-credible) of a new incoming tweet. Both the model generation and the prediction operations require the extraction a set of features to be associated with tweets. Using the Twitter API, the system extracts two types of features: (1) content-related features: extracted from the content of a tweet (see Table I) and (2) user-related features: extracted from the

* Corresponding Author

[1]In this paper, the term fake news refers to the information published by non-credible tweets.

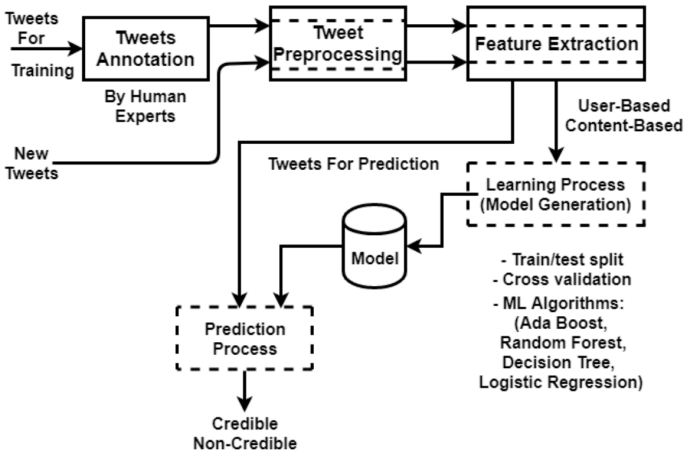

Fig. 1: System Overview

profiles of users such as status counts and registration date (see Table II). However, not all features can be extracted directly via the Twitter API. Some features can only be derived by using a systematic feature engineering process as detailed in Section VII.

This paper is organized as follows. In Section II, we review some related research efforts, and then present the definition of credible news along with the problem statement in Section III. In Section IV, we describe the generated features along with the different tools used for extraction. In Section V, we outline the various Machine Learning algorithms used to build the classification model. In Section VI, we introduce the various experiments used for model optimization, and dataset processing. In Section VII we present the effects of sentiment-based features on our models and predictions. Finally, we conclude our paper with Section VIII.

II. Related Work

Assessing the credibility of tweets with the aim of filtering out those holding fake and unreliable information has attracted the attention of the research community. Several studies on detecting bots and fake accounts in social media have been undertaken with the goal of detecting fake news. For instance, Gurajala et al. [5] analyzed the characteristics of fake Twitter accounts by establishing a strategy to automatically identify generated fake profiles. One of the core characteristics adopted was the followers-to-friends ratio because fake users usually focus more on gathering friends.

Another research effort by Dickerson et al. [2] proved that sentiment-related factors are core to identifying bots on social media. They developed a collection of network and linguistic variables that could be used as features for distinguishing between humans and bots. For this reason, we also included sentiment features in our work as they are likely to enhance the performance of our model.

Only a few research studies have attempted to detect non-credible Arabic tweets. Sabbeh et al. [12] utilized a set of topic and user-related features to quantify news credibility using a machine learning model. Additionally, polarity analysis of users' replies was employed to gain a more accurate assessment. Adding polarity of users' comments is beneficial for tweets credibility assessment. Al-Khalifa et al [7] proposed a system to measure the credibility of news content published in Twitter. The system assigns three levels of credibility to each tweet, namely low, medium, and high. Two approaches are utilized to assign the judgment value. The first approach utilizes authentic news sources to find a connection between a Twitter post and these sources. In the second approach, they use the results of the first approach in addition to a set of proposed features. However, their findings show that assigning credibility levels using the first approach resulted in higher precision and recall.

In this work, we improve the accuracy of our model and the prediction results by utilizing sentiment analysis for Arabic tweets in addition to the use of a combination of user and content features. Moreover, the accuracy obtained from our model is boosted by conducting extensive hyper-parameter tuning, and feature normalization.

III. Background and Preliminary Concepts

In this section, we describe the structure of a "tweet" as the main textual entity under study. Then, we discuss the definition of "tweet credibility" that is considered throughout the paper, Finally, the problem statement is presented.

A. Tweets

Tweets are status messages that can contain a maximum of 280 characters [15] and are published by people all around the world to share information, interact with others, publish news, jokes or even pointless babbles. Moreover, the content of each tweet might contain a number of markup symbols which can be summarized as follows: (1) *hashtag* "#": used to attach some key phrases to a tweet, and thus, to group and categorize tweet messages according to the theme of the tweet. Hashtags make it possible for users to track specific trends or events in real-time; and (2) *mention* "@" followed by a username which indicates that a tweet message is a reply to a user and the sender wants to draw the attention of the recipient to the topic of the tweet. There are other markup symbols such as "reply to" and "RT" indicating that a user is replying to another user's tweet and the tweet is just a forward of another tweet, respectively. All these markups add a thematic dimension to tweets, which attracts many research efforts to make use of them in extracting useful insights.

B. Credibility of Tweets

Twitter has spent a lot of efforts to ensure trust on their web service due to the widespread propagation of erroneous and misleading information[1] such as rumors and fake news. This could result in a lot of damage [4] and thus justifies the actions being conducted to mitigate the adverse consequences of publishing non-credible tweets.

[1]https://blog.twitter.com/en_us/topics/company/2018/how-twitter-is-fighting-spam-and-malicious-automation.html

Fig. 2: Example of Arabic fake news

To build an effective model that can screen out "non-credible" tweets, there is a need to clearly define the *credibility* of a tweet and distinguish between *credible* and *non-credible* tweets. Although the term "credible tweet" has several definitions in the literature, there is no general definition that had been agreed upon by the research community. For instance, a non-credible tweet is defined as a tweet containing news that is posted with the intention to mislead people [3]. In [11], non-credible tweets are those conveying news that includes serious fabrications and misinformation. In this paper, we consider the second definition of credibility, and thus a "credible" tweet is a tweet containing genuine, high quality news and information. On the other hand, non-credible tweets are those conveying fake and false information regardless of the user's intention.

Figure 2 shows an example of a non-credible Arabic tweet that was made by an activist to report a mysterious explosion in Saudi Arabia. The reason was found later to be due to an oil truck explosion.

C. Problem Statement

To alleviate the consequences of publishing fake Arabic news, we need to automatically identify non-credible tweets using a machine learning technique based on features extracted from the content of tweets and information about the publishing user. In other words, the goal is to build a binary classifier that can categorize a certain tweet into "non-credible" or "credible" along with a probability estimate.

Many research efforts have been accomplished to detect fake news in social media. However, developing a machine learning model for detecting non-credible Arabic tweets proved to be a challenging task for several reasons [7]: (1) Most NLP tools do not support the Arabic language. (2) The noisy nature of Twitter content.

IV. Features Extraction

In this section, we detail the different features used in this study to train our tweets classifier. Using Tweepy API[1], we extracted a total of 45 features for each tweet. These extracted features can be categorized into the following types: (1) content-based features, and (2) user-based features.

[1]http://docs.tweepy.org/en/v3.5.0/api.html

Content-related Features

Analyzing the content of a tweets is crucial for detecting a non-credible tweet. For this, a set of 26 content-related features (listed in Table I) are constructed, where some of these features are extracted directly from the Twitter API, such as `has_mention` and `is_reply`, others have to be derived from other features (e.g: `count_unique_chars`). With respect to "mentions", a feature indicating whether a tweet contains a mention or not, namely, `has_mention`, and another feature referring to the number of mentions in the tweet are extracted. Similar two features are calculated for the contained URLs and retweets. Another `is_reply` feature is included to indicate whether the tweet is a reply to another tweet. The feature `retweeted` indicates whether the tweet has been retweeted by the user or not. In addition, `day_of_week` corresponds to the day on which the tweet was published. To quantify the total number of words in a tweet, we use `#_words`, while `#_unique_words` refers to the number of distinct words in a tweet. Likewise, we created a feature for `#_unique_chars`. Finally, The feature `count_symbols` represents the number of special characters, i.e., non-alphabet or numeric characters.

Since Twitter is considered a rich source of emotions people show while expressing their feelings towards a certain aspect (see [9]), we included 4 sentiment features inferred from the text of tweets, namely, (1) `has_positive_sentiment`, (2) `has_negative_sentiment`, (3) `positive_score`, and (4) `negative_score`. To estimate `positive_score` and `negative_score` features, a set of Arabic sentiment words [13] stored in a lexicon with about 5212 words are used. Each word in the lexicon has an associated positive or negative score. We used rule-based approach to compute the sentiment features, by searching for positive words from the tweet and summing up their respective scores retrieved from the lexicon to produce `positive_score`. The same procedure is performed to calculate `negative_score`. The stem of each word is extracted using NLTK[2] and utilized in finding a match for the word in the lexicon. Sentiment-based features accounts for **9%** of the feature set.

TABLE I: Content-Based Features

has_mention	mentions_count	URL_shortner
retweets_num	is_retweet	is_reply
retweeted	day_of_week	#_words
length_chars	has_URL	URLs_count
has_hashtag	hashtags_count	#_unique_words
#_unique_chars	has_?	#_?
has_!	#_!	#_ellipses
#_symbols	has_pos_sent	has_neg_sent
pos_score	neg_score	

User-related Features

User-related features are directly engineered from the user profile associated with each tweet. Some of these features pertain to the description attached to user `has_description`,

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

others relate to usernames `username_len`. Some features are derived from already existing user-specific fields, e.g., `registration_diff` which is the difference between the current time and the creation date in days. A complete list of user-related features are listed in Table II. While others utilize timeline related properties, e.g., average favorite count, average tweet length, listed count.

Moreover, we extracted features relating to the users' most recent 20 tweets (i.e users timeline's). These features consists of `re-tweet_fraction` which is the ratio the of re-tweets of the timeline tweets. `tweet_time_ spacing` shows the average time difference between each two consecutive tweets. `focused_topic` is the ratio of the most tweeted hashtag in the user timeline tweets over the total number of tweets in the timeline.

TABLE II: User-Based Features

avg_hashtags	avg_URLs	followers_to_friends
avg_retweet	followers_count	tweet_time_spacing
focused_topic[1]	default_image	has_desc
desc_len	username_len	avg_tweet_len
listed_count	status_count	retweet_fraction
friends_to_follower	is_verified	registration_diff

These features described above account for **49%** of the entire set of employed features. In addition, a number of new features are derived to help boost the performance of the built classifier using a systematic feature engineering process.

Derived Features

These new programmatically-generated features account for **51%** of the either set of features and are extracted from fields related to tweet content and to the associated user profile. We can divide them into two categories: (1) element-based features. 2) sentiment-based features.

Element-based Features: Element-based features are computed from the attributes (elements) of a tweet or the publishing user, such as `len_tweet_words`, `len_tweet_char`, and `#_unique words`. Examples of features derived from the user profile are: `avg_favorite_count`, `avg_tweet_len`, and `avg_hashtag_count`.

Sentiment-based Features: Sentiment Analysis is the attempt to quantify the feelings and attitude of a user toward a topic from a piece of text. We use rule-based approach to extract our sentiment-based features, where a tweet is classified into either positive or negative. The process of extracting sentiment analysis features and their effects on our results is comprehensively discussed in Section VII. Sentiment-based features are inferred from the text of the tweet. Based on the quantified sentiment, a number of features are extracted: `has_positive_words`, `has_negative_words`, and `sentiment_score`.

V. Machine Learning Algorithm

Using Scikit-learn [10], we examined a number of machine learning algorithms, such as k-nearest neighbors, random forest, adaboost, SVM, neural network, and logistic regression. After initial experimentation, we retain only four algorithms for further experimentation based on the following criteria: (1) Accuracy: using the default parameters, the algorithms that lead to few false-positives, few false-negatives, larger number of true-positives and larger number of true-negatives are chosen. (2) Speed of convergence with minimal parameter tuning. Namely, the following ML algorithms are used for our fake news prediction task: **Decision Tree Classifier**, **Random Forest**, **AdaBoost**, and **Logistic Regression** for further parameter tuning and experimentation.

The results obtained from a ML model may vary based on the randomly-chosen initial values of the model parameters. Each time the model training process is triggered on a randomly-chosen seed, the model parameters are assigned different values. Therefore, we set the seed to a constant value across the applied models. This is used to generate each model's initial hyper-parameters to the same values across our various experiments, in order to reproduce the results and to properly verify improvements while tuning the model hyper-parameters.

VI. Experimental Evaluation

In this section we describe a number of experiments conducted to verify the effectiveness of our tweet classification model. All of the experiments were accomplished using *Google Colaboratory*[1] environment, *Colab* for convenience. Google Colab offers a Jupyter notebook environment that runs entirely in the cloud. Each experiment is done using four machine learning models, namely Random Forest, Decision Tree, Logistic Regression, and AdaBoost.

A. Dataset

We utilized a dataset used in [1], containing a total of 1862 tweets published on topics covering the Syrian crisis, which is very suitable for credibility assessment as many independent activists were using Twitter to publish crisis-related tweets that might hold some false information. The dataset is relatively balanced around the two classes, with 1051 credible and 810 non-credible tweets. The original dataset contained 2708 tweets. However, a number of tweets are excluded because they become unavailable and could not be reached via the public Twitter API, and thus the feature extraction process cannot be achieved for these tweets.

B. Models Optimization

Initially, the hyperparameters of the used models are optimized using scikit-learn's GridSearchCV module. The optimization performed using GridSearchCV that exhaustively searches within a predefined range of chosen hyperparameter values over the training set. For each model, and due to the sheer amount of possible hyperparemeter combinations, we first optimized the model hyperparameters using relatively sparse values in order to identify the most influential hyperparameters. After that, we chose the top two hyperparameters

[1]https://colab.research.google.com

TABLE III: Hyperparameters optimization for each classifier

Model	Parameter	Value
Random Forest	bootstrap	False
	n_estimators	300
Decision Tree	max_depth	10
	max_features	47
Logistic Regression	C	10
	solver	newton-g
Ada Boost	algorithm	SAMME
	n_estimators	500

TABLE IV: Model performance without standardized features

Model	Accuracy	Precision	Recall	F1-score
Random Forest	74%	80%	79%	79%
Decision Tree	69%	75%	74%	74%
Logistic Regression	74%	78%	80%	79%
Ada Boost	74%	78%	83%	80%

that have the highest accuracy for extensive optimization over more dense range of values.

Moreover, to accurately validate the yielded results, we performed cross validation during the learning process, which provides an insight on how effective a model will perform on unseen tweets. The outcome of this step is illustrated in Figure 3. The tuned parameters for the used ML algorithms are provided in Table III, where any other hyperparameter not listed in the table is set to its default value.

C. Dataset Processing

In this step, various techniques were applied on the dataset, ranging from cleaning and balancing to normalization and standardization.

1) Standardization: Standardization transforms each feature so that it has a mean of 0 and standard deviation of 1. Standardization is achieved by applying (1), where $\acute{x}$ is the resulted standardized feature vector. x is the original feature vector, $\bar{x}$ is the mean of the feature vector, and σ is the standard deviation. Standardization was achieved by utilizing scikit-learn's [10] StandardScaler package. The results of applying standardization are shown in Table V. The results from the same experiment on the original dataset are illustrated in Table IV.

$$\acute{x} = \frac{x - \bar{x}}{\sigma} \tag{1}$$

2) Min-max Rescaling: Rescaling is the process of transforming the extracted features to obtain values in the range [0,1]. This can be done by applying Eq. (2) on the feature vectors, where $\acute{x}$ is the resulted feature vector after rescaling, x is the original feature vector, and $\min(x)$ and $\max(x)$ are the minimal and maximal value of vector x, respectively. The resulted scores are reported in Table VI.

$$\acute{x} = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{2}$$

This sort of feature rescaling is important because some features have continuous values with wide range, e.g., *tweet_text_length*, while others have a value of either 0 or 1, such as *has_default_profile_image*. Without rescaling, some features will have more influence on the target variables than others do.

TABLE V: Results when standardized features are used

Model	Accuracy	Precision	Recall	F1-score
Random Forest	76%	78%	82%	80%
Decision Tree	70%	75%	74%	74%
Logistic Regression	77%	78%	85%	81%
Ada Boost	76%	79%	81%	80%

TABLE VI: Results after applying min-max rescaling

Model	Accuracy	Precision	Recall	F1-score
Random Forest	76%	79%	82%	80%
Decision Tree	70%	75%	74%	74%
Logistic Regression	76%	79%	83%	81%
Ada Boost	77%	80%	82%	81%

VII. Discussion

Sentiment analysis is exploited in hope obtain a better accuracy across our models. Table VII presents a comparison between the performance of the system without sentiment features (first row) and using 4 sentiment-based features as detailed in Section IV (second row). We notice the positive impact of sentiment features on the accuracy of the system, in particular when ensemble-based ML algorithms are used.

TABLE VII: Results of the optimized models

Model	Accuracy	Precision	Recall	F1-score
Random Forest	68%	74%	72%	73%
	76%	79%	82%	80%
Decision Tree	70%	76%	74%	75%
	69%	75%	74%	74%
Logistic Regression	76%	78%	83%	81%
	75%	78%	83%	80%
Ada Boost	74%	78%	79%	78%
	74%	78%	80%	79%

Quantifying the sentiment of a tweet might have a considerable impact on identifying a fake tweet when applied on the replies of the tweet under investigation. In other words, utilizing the wisdom of the crowd detecting the credibility of a tweet based on their feelings uncovered from their replies. If the majority of relies have negative polarity, then this might be a good evidence that a tweet is fake. We leave examining the polarity of replies as new features for future work.

In Figure 4, we illustrate the impact of selecting a subset of features on the overall accuracy. Features having a variance lower than a predefined threshold is eliminated. Although the value of 0.1 achieved the highest accuracy in AdaBoost, we set the threshold to 0.2 since it leads to high accuracy in most of the tested models.

VIII. Conclusion

As Twitter has become a leading social media platform for news consumption, many users tend to post tweets with false information for different financial and political purposes. Identifying tweets with fake news is a non-trivial task due to many variables that can be purposely altered. As a result, we introduced a machine learning model for quatifying the crediility of tweets in Arabic language. For this, we engineered different features and categorized them into two categories: content-based features and user-based features. During the learning process, different machine Learning algorithms are

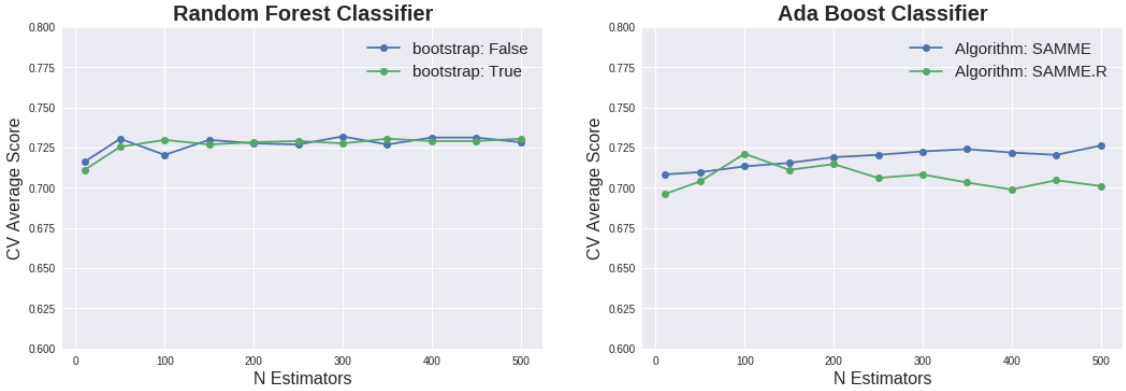

Fig. 3: Hyperparameters optimization for Random forest and AdaBoost.

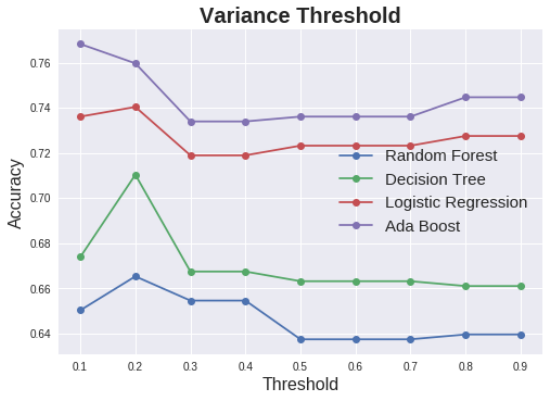

Fig. 4: Variance threshold vs. Accuracy.

applied and the ones with the best results are retained. Finally, we compared between the output of each experiment and presented the results of each one. The experimental evaluation reveals that our system can filter out non-credible tweets with an accuracy of 76%. As ongoing work, we plan to conduct further feature engineering and to utilize Twitter replies to improve the overall system accuracy.

References

[1] Ayman Al Zaatari et al. "Arabic Corpora for Credibility Analysis." In: *LREC*. 2016.

[2] John Dickerson, Vadim Kagan, and VS Subrahmanian. "Using Sentiment to Detect Bots on Twitter: Are Humans more Opinionated Than Bots?" In: *Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis*. 2014, pp. 620–627.

[3] BJ Fogg et al. "What Makes Websites Credible? A Report on a Large Quantitative Study". In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM. 2001, pp. 61–68.

[4] Manish Gupta, Peixiang Zhao, and Jiawei Han. "Evaluating Event Credibility on Twitter". In: *Proceedings of the 12th SIAM SDM* (Apr. 2012), pp. 153–164.

[5] Supraja Gurajala et al. "Profile characteristics of fake Twitter accounts". In: *Big Data & Society* 3.2 (2016).

[6] Mike Isaac and Sydney Ember. *For Election Day Influence, Twitter Ruled Social Media*. https://www.nytimes.com/2016/11/09/technology/for-election-day-chatter-twitter-ruled-social-media.html. [Online: accessed 02-Mar-2019]. 2016.

[7] Hend S Al-Khalifa and Rasha M Al-Eidan. "An experimental system for measuring the credibility of news content in Twitter". In: *International Journal of Web Information Systems* 7.2 (2011), pp. 130–151.

[8] Katerina Masta and Elisa Shearer. *News Use Across Social Media Platforms 2018*. http://www.journalism.org/2018/09/10/news-use-across-social-media-platforms-2018/. [Online: accessed 02-Mar-2019]. 2018.

[9] Brendan O'Connor et al. "From Tweets to Polls: Linking Text Sentiment to Public Opinion Time Series." In: *Icwsm* 11.122-129 (2010), pp. 1–2.

[10] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[11] Victoria L Rubin, Yimin Chen, and Niall J Conroy. "Deception Detection for News: Three Types of Fakes". In: *Proceedings of the 78th ASIS&T*. 2015, p. 83.

[12] Sahar Sabbeh and Sumaia Baatwah. "Arabic News Credibility on Twitter: An Enhanced Model Using Hybrid Features." In: *Journal of Theoretical & Applied Information Technology* 96.8 (2018).

[13] Mohd Saif, Salameh Mohd, and Kiritchenko Svetlana. "Sentiment Lexicons for Arabic Social Media". In: *Proceedings of 10th edition of LREC*. 2016.

[14] Craig Silverman. *This Analysis Shows How Viral Fake Election News Stories Outperformed Real News On Facebook*. https://www.buzzfeednews.com/article/craigsilverman/viral-fake-election-news-outperformed-real-news-on-facebook\#.JP. 2016.

[15] *Twitter officially expands its character count to 280.* https://techcrunch.com/2017/11/07/twitter-officially-expands-its-character-count-to-280-starting-today/. 2016.