Received: October 26, 2019

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Credibility Detection in Twitter Using Word N-gram Analysis and Supervised Machine Learning Techniques

Noha Y. Hassan[1]* **Wael H. Gomaa[2]** **Ghada A. Khoriba[3]** **Mohammed H. Haggag[3]**

[1]Computer Science Department, Beni-Suef University, Beni-Suef, Egypt
[2]Information Systems Department, Beni-Suef University, Beni-Suef, Egypt
[3]Computer Science Department, Helwan University, Helwan, Egypt
* Email: noha.yehia@fcis.bsu.edu.eg

Abstract: With the evolution of social media platforms, the Internet is used as a source for obtaining news about current events. Recently, Twitter has become one of the most popular social media platforms that allows public users to share the news. The platform is growing rapidly especially among young people who may be influenced by the information from anonymous sources. Therefore, predicting the credibility of news in Twitter becomes a necessity especially in the case of emergencies. This paper introduces a classification model based on supervised machine learning techniques and word-based N-gram analysis to classify Twitter messages automatically into credible and not credible. Five different supervised classification techniques are applied and compared namely: Linear Support Vector Machines (LSVM), Logistic Regression (LR), Random Forests (RF), Naïve Bayes (NB) and K-Nearest Neighbors (KNN). The research investigates two feature representations (TF and TF-IDF) and different word N-gram ranges. For model training and testing, 10-fold cross validation is performed on two datasets in different languages (English and Arabic). The best performance is achieved using a combination of both unigrams and bigrams, LSVM as a classifier and TF-IDF as a feature extraction technique. The proposed model achieves 84.9% Accuracy, 86.6% Precision, 91.9% Recall, and 89% F-Measure on the English dataset. Regarding the Arabic dataset, the model achieves 73.2% Accuracy, 76.4% Precision, 80.7% Recall, and 78.5% F-Measure. The obtained results indicate that word N-gram features are more relevant for the credibility prediction compared with content and source-based features, also compared with character N-gram features. Experiments also show that the proposed model achieved an improvement when compared to two models existing in the literature.

Keywords: Credibility detection, Twitter, Fake news, Machine learning, N-gram analysis, TF-IDF representation.

1. Introduction

Social media is used for sharing news, opinions and experiences. It is now being used as a source of news rather than traditional media [1]. Recently, organizations especially the political are highly interested in analyzing the content on social media to measure the public opinion and people satisfaction towards different issues. Twitter is one of the most widely used social media platforms that has 330 million monthly active users [2]. Twitter enables users to send short messages and disseminate them easily through “re-tweet”. During emergencies, Twitter has proved to be very useful because of its ability to propagate news much faster than traditional media. News on Twitter can come from authorized news organizations, but most of them come from public users. Unlike traditional media sources, the absence of supervision and the ease of spreading make Twitter an environment conducive to rumors and fake news [3]. This issue becomes a problem as more people rely on social media for news especially during emergencies [4, 5]. A recent study [6] stated that fake news published on Twitter during the last American presidential elections in 2016, had a significant effect on voters. Several studies have shown that much of the content on Twitter is not credible [7-9]. Research by ElBallouli et al. [8] found that approximately 40% of the tweets posted per day are not credible tweets. Moreover, Gupta et al. [9]

DOI: 10.22266/ijies2020.0229.27

presented a study of fake news spreading during Hurricane Sandy. The study revealed that 86% of the rumors were "re-tweets". They found out that during the crisis people share news even if it is from an unknown source. Nowadays, determining the credibility of the content on Twitter is highlighted especially in the case of emergencies.

In fact, it is difficult to manually identify credible tweets. Several approaches have been presented for automatically predicting the credibility of tweets. These approaches are categorized into classification-based and propagation-based approaches. Propagation based approaches focus on the propagation concept to detect the credibility and rely on the network structure and social graph analysis [10, 11]. Social networks can be represented as a graph composed of nodes (Twitter users) and relationships connecting them (such as: follows, replies, mentions and tweets) called edges. These inter-entity relationships on Twitter can provide rich information and many studies incorporated graph analysis to measure information credibility.

Classification based approaches classify tweets into credible and not credible based on features extracted from them using machine learning techniques especially supervised techniques [8, 12-15, 17-19]. Supervised machine learning techniques require a ground truth that contains a dataset of annotated tweets with the features related to them. The relevance of the extracted features is an important factor affecting the efficiency of the prediction. There are several types of features introduced by previous research in this area. Most of these studies rely on content-based and source-based features. Content-based features focus on the content of the tweet itself such as the length of the message, the number of unique characters or emoticons and if the message contains a hashtag (#) or URLs. Source-based features consider characteristics of the user such as the number of followers and if the user is verified. Some studies used a combination of content-based and source-based features [8, 17, 18]. After the feature dataset is built, the next step is to determine the optimal classification algorithm to train them. Decision trees[8, 12, 19] and support vector machines (SVM) [14, 15] are the most popular supervised learning techniques used for classification.

To predict the credibility of a tweet, we should consider the content of the tweet as an important factor. This paper focuses on the credibility problem and introduces a supervised learning model based on word N-gram analysis and machine learning techniques to automatically classify tweets into credible and not credible.

Main contributions:

1) We show that word N-grams are more powerful than content and source-based features in predicting the credibility of Twitter posts.
2) We apply different machine learning techniques and compare their performance on two different datasets. Also, we measure and compare the performance of different feature representations (TF, TF-IDF) and the effect of the size of N on the performance as well as the number of extracted features (top features).
3) The experimental results showed that the proposed model outperforms Zubiaga et al. [17] and Ajao et al. [22] which use different models over the same dataset by 28% and 48% respectively in terms of F-measure .
4) We developed an online mobile application to extract real-time tweets and classify the resulted tweets according to their credibility.

This paper is organized as follows: In section 2, related work in credibility prediction is presented. The proposed model is presented in section 3. Next, in section 4 we describe how the model is evaluated and present the results of our experiments. Section 5, we evaluate the proposed model in comparison with two other models. Finally, section 6 includes the conclusions and future work.

2. Related work

The literature includes many studies on automated classification approaches based on supervised machine learning. In this section, we review some of the published work in this area.

Castillo et al. were the first group to work on the problem of credibility and proposed a model that automatically classify tweets based on features extracted from them [12, 13]. The research identified different types of features. Some features are related to the content or the author of the tweet while others are aggregated from the related topic. The extracted features were used to train a set of classifiers like SVM, Bayesian networks and decision trees. They achieved credibility classification with an accuracy of nearly 86% using the J48 decision tree. The research provided a feature analysis to perform the best feature selection process. The study indicated that the best features are related to the users such as the duration they spent as Twitter users, the number of followers that they have, and the number of tweets that they have written. Gupta et al. [14] proved that predicting the credibility of Twitter messages could be automated accurately. The research identified some relevant features such as the number of followers, number of unique characters and swear words.

Results showed that approximately 30% of tweets posted in an event include information about the event, 14% of the tweets related to an event were spam while only 17% are credible tweets. Another research by Lorek et al. [15] focused on the external link features and check if the link's content matches the tweet content or it leads to an interactive ad instead. O'Donovan et al. [16] identified the most useful indicators of credibility as the existence of URLs, mentions, retweet count, and tweet lengths.

Another research by Zubiaga et al. [17] developed a credibility detection system that warns users of unverified posts. Twitter streaming API was used to collect 5802 tweets related to five breaking news stories. The research evaluated and compared the performance of the system using two different feature sets: content-based features and social features. Conditional Random Fields (CRF) was used as a sequential classifier and its performance was compared with three more classifiers. The experimental results showed that the features related to the text of the tweet itself (such as word vectors, word count and the existence of question marks) are good indicators for credibility. Another set of content and source-based features were applied in [18] and examined over the same dataset. The research showed that the source-based features are more discriminant than other features as they indicate the author's experience and reputation. The model recorded an improvement of 18% in terms of F-measure over CRF [17] when using content-based features while the improvement was 49% when using source-based features. In fact, it is not easy to generate these handcrafted features and some of them can be misleading. The number of followers of a user or the number of retweets should not indicate the credibility of the tweet because malicious users can easily forge followers or re-tweets. Moreover, Twitter users often re-tweet without verifying the content [20].

Credibility assessment has been studied from another point of view based on similarity. Al-Khalifa et al. [21] developed a model to measure credibility of Twitter messages and assign the credibility level (high, low and moderate) to each tweet. The proposed model is based on the similarity between Twitter messages and authorized news sources like Aljazeera.net. The proposed model achieved acceptable results but requires the existence of credible external sources.

Another recent research tried to solve the credibility problem by using deep learning approach [22, 23]. Ajao et al [22] focused on RNN and long-short term memory model (LSTM) as it is the most widely used deep learning model for text classification. The research introduced a framework that predicts the credibility and detects the fake Twitter posts with accuracy 82%. The experiments were done using the same dataset that was used in [17, 18] and in this work as well. Deep learning models enable automatic feature extraction, but it requires large amount of labeled data for the perfect training of their models.

Text-based features are considered one of the most discriminative features that were used to represent documents in many applications [24-26]. These features can be N-grams, text similarity and POS tags. N-gram model is a statistical technique used in document classification to capture the relationships between words and use these relationships to predict the category to which a document belongs. In a recent study, Nieuwenhuis and Wilkens [26] presented a text and image gender classification using the N-gram model. They used word and character N-grams as textual features in addition to some image base features to predict the gender of a Twitter user. The outcome of the research is that the best results were achieved by using only the text features. N-grams are non-handcrafted features and easy to generate. No additional features are needed, only the tweets' text. Also, there is no dependence on pre-trained word embeddings or large corpora for training. They can capture the discriminative power of words as phrases and can be surprisingly powerful, especially for real-time detection.

3. Proposed model

We aim to develop a model for automatically classifying tweets into credible and not credible. In this section, we will discuss our model which is based on text analysis using word N-grams. Fig. 1 shows the proposed model architecture which consists of two modules, offline and online modules. The first module is used to train and build the classification model. The annotated dataset described in section 3.4 is input to the preprocessing step then N-gram features are extracted in order to create feature vectors. The feature vectors with their credible/not-credible labels are passed to the classifier training process to learn different tweet patterns and minimize the classification error. The N-gram model, preprocessing and feature extraction processes are described in the following sections.

The output model is then used as input to the online prediction module. We developed an online mobile application to extract real-time tweets based on certain text query. The result of the search query is used as input to the trained model after performing

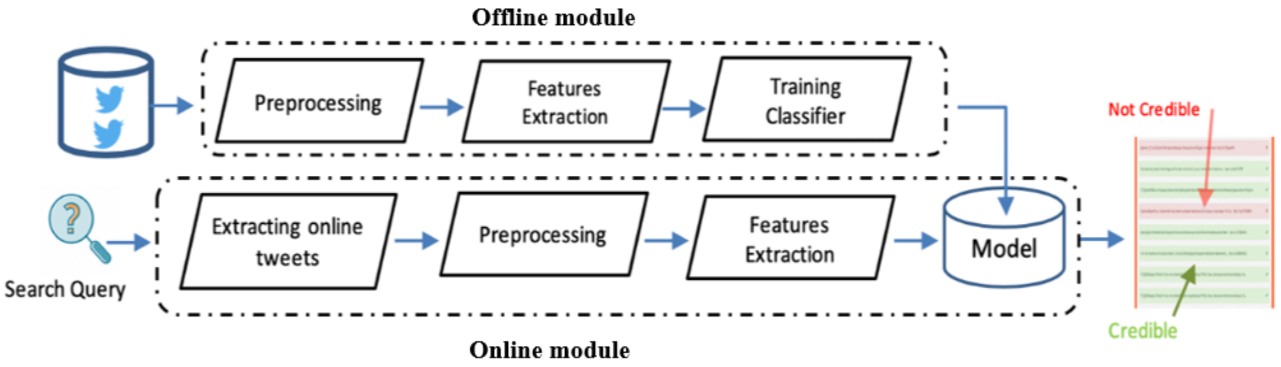


Figure. 1 The proposed model architecture

preprocessing and feature extraction tasks on them to predict their credibility.

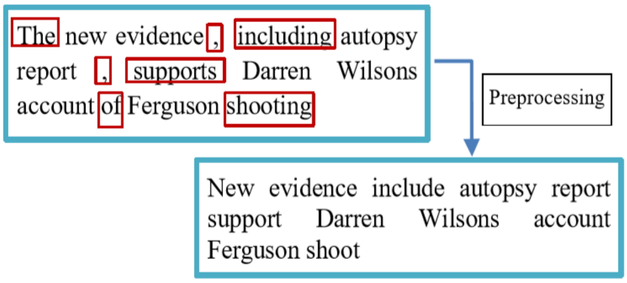
As shown in Fig. 1, the output of the search query is a list of tweets where each single tweet is distinguished by green color for the positive tweet (credible) and red color for negative ones (not credible).

3.1 Preprocessing

Data cleaning and preprocessing functions are required before extracting N-grams to reduce the text feature size. The dataset was cleaned by removing tweet ID, tweet time, hyperlinks, emoticons, punctuations and non-letter characters. Then preprocessing functions like stop word removal, stemming and tokenizing were done to remove trivial data and reduce the size of the actual data. Stop words are the words which occur commonly across all the tweets but actually they are insignificant like; a, an, the, will, was, were, is, are, to, of, that, these, what, when etc. These words must be removed because they are not discriminant when used as features in the classification task. Stemming is the process of removing suffixes and reduce words to their word stem. For example, words like (connects, connected, connecting, connection) all have the same meaning. Removing the suffixes (-ed, -s, -ion, -ing) and leaving the single word (connect) will reduce the number of unique words and make classification more efficient. Fig. 2 shows an example of a tweet before and after preprocessing.

3.2 Features extraction

Features extraction is the process of obtaining the most relevant information from the original data forming the feature vectors. Most of the machine learning algorithms cannot accept the raw text data as input because they expect numerical feature vectors with a fixed size. Vectorization is the process of turning a collection of text documents into numerical feature vectors to represent them into a lower dimensionality space. This process consists of two stages: First tokenizing strings and giving an integer id for each possible token, then weighting the tokens or terms to represent the importance of each token. In this study, we used two different weighting methods, namely, Term Frequency (TF) and Term Frequency-Inverted Document Frequency (TF-IDF). Term frequency is simply assigning the weight to be equal to the number of occurrences of word w in tweet t. In this scheme, tweets are described by word occurrences where the word that occurs frequently is rewarded with completely ignoring the relative position of the words in the tweet. TF-IDF is used in machine learning and text mining as a weighting factor for features. The weight increases as the word frequency in a document increases but that is offset by the number of times that word appears in the dataset. This mechanism helps to remove the importance from really common words that appear frequently in all documents and rewards words that are rare overall in a dataset. High TF-IDF weight is reached when a word has high TF in any given tweet and low DF of the word in the entire dataset. TF-IDF of a word in any given tweet is the product of the TF of this word and the inverse document frequency IDF of the word where,


The new evidence , including autopsy report , supports Darren Wilsons account of Ferguson shooting

New evidence include autopsy report support Darren Wilsons account Ferguson shoot

Figure. 2 Preprocessing example

$$IDF_w = log\ \frac{1+N}{1+df_w} + 1 \qquad (1)$$

where document frequency df_w is the number of tweets containing a word w in the entire dataset and N is the number of tweets.

N-gram model is commonly used in natural language processing applications. N-grams are sequences of words or characters as they appear in texts one after another where "N" corresponds to the number of elements in a sequence. The most used N-gram models in text analysis are word-based and character-based N-grams. In this work, we used word-based N-grams with n=1 (unigrams) which is known as the bag of words BOW, n=2 (bigrams) and n=3 (trigrams). For example, given this tweet ("Great Pyramids of Egypt"), the word-based unigrams (n=1) are (Great, Pyramids, Egypt) while bigrams are: (Great Pyramids, Pyramids of, of Egypt) and so on. The idea is to generate various sets of N-gram frequencies from the training data to represent the collected tweets. When using word-based N-gram analysis, determining the perfect value of N is an area of research. We used different values of N to generate N-gram features and examine the effect of the N-gram length on the accuracy of the different classification algorithms.

Our task is to predict whether a tweet is credible or not depending on the presence or absence of the most discriminative words related to it. For this task, we used the unigram model to compute the frequency for each word on the training set. Unigram model ignores the complete context of the word, so we extend our model by using bigrams, trigrams, and quad-grams. We observe that the unigram model achieved good results but using a combination of unigrams and bigrams leads to better performance.

3.3 Machine learning classifiers

We compared the performance of five different supervised classifiers namely: Linear Support Vector Machines (LSVM), Logistic Regression (LR), Random Forests (RF), Naïve Bayes (NB) and K-Nearest Neighbor (KNN) in order to choose the best classifier. we evaluated the five classifiers with the two feature representations which we have extracted (TF/TF-IDF) in two separated experiments. The implementation of the classifiers in *scikitlearn* python library [27] is used setting all the parameters to the default values.

3.4 Datasets

Machine learning techniques require building a dataset contains a collection of tweet messages. The messages in this dataset are then labeled by human annotators which is an important step affecting the accuracy of the model. Publicly datasets are not available, so we used the PHEME [28] dataset collected and labeled by Zubiaga [17] in our experiments. Twitter streaming API was used to collect the dataset during some famous events and accidents. The events were highly commented and retweeted at the time of occurrence namely: " Charlie Hebdo, Sydney Siege, Ottawa Shooting, Germanwings-Crash, and Ferguson Shooting". The authors selected the tweets that have the highest number of retweets to use them as samples. The annotation process was conducted and reviewed with the assistance of a team of journalists. The dataset contains 5802 tweets and was annotated as 3830 (66%) credible and 1972 (34%) non-credible tweets.

Another dataset was used in this work is the Arabic dataset collected and labeled by El Ballouli et al. [8]. Around 17 million Arabic tweets were collected using Twitter Streaming API. Data cleaning was performed by removing all retweets, tweets that contain hashtags or emoticons only and all tweets that are ads. The process ended with 9000 tweets addressing different topics. The final dataset contains tweet text and metadata about the tweet and the author as well. The annotation process was carried out by a group of seven annotators, each was provided by the tweet's URL and the author's profile to guide them in determining the label of the tweet. The 9000 Arabic tweets were finally annotated as 5400 (60%) credible and 3600 (40%) non-credible.

4. Experimental results

Our experiments included training the proposed model using the two datasets described in Section 3.4. We applied 10-fold cross validation on the entire dataset and use different performance measurements to evaluate the results. Accuracy, Precision, Recall, and F-measure as follows:

$$\textit{Accuracy} = \frac{(TP+TN)}{(TP+FP+TN+FN)} \tag{2}$$

$$\textit{Precision} = \frac{TP}{(TP+FP)} \tag{3}$$

$$\textit{Recall} = \frac{TP}{(TP+FN)} \tag{4}$$

$$\textit{F-measure} = \frac{2\ (Precision \times Recall)}{(Precision + Recall)} \tag{5}$$

Where *TP* is the number of tweets correctly identified as credible, *FP* is the number of tweets incorrectly identified as credible, *TN* is the number of tweets correctly identified as non-credible and *FN* is the number of tweets incorrectly identified as non-credible. The following sections present the results of our experiments on the two datasets.

4.1 PHEME dataset

We conducted two experiments over the PHEME dataset [28]: in the first experiment, we used the annotated tweets and the TF extracted features to train and test our model. We run the five classifiers on the dataset using different sizes of N-grams. We start the experiment with unigram (n=1) then increase (n) until reaching trigrams (n=3). To gather more context, we combined unigrams and bigrams which achieves the highest performance. Table 1 below shows the Accuracy, Precision, Recall, and F-measure results of the proposed model using a combination of both unigrams and bigrams with TF feature representations. Linear SVM achieved the best accuracy and F-measure, NB achieved the best Precision, and KNN achieved the best Recall.

We repeated the previous experiment using TF-IDF features and recorded the results. Table 2 below shows the Accuracy, Precision, Recall, and F-measure results of the proposed model using TF-IDF feature representations with Unigrams and Bigrams. Linear SVM achieved the best accuracy, precision, and F-measure while the best Recall was achieved by NB classifier.

The results indicate that the accuracy rate is affected noticeably by increasing the size of N-gram with different classifiers. All the classifiers achieved the highest performance when using a combination of unigrams and bigrams. Adding trigrams to the combination doesn’t affect the accuracy rate significantly. Fig. 3 shows the effect of changing the size of N on the accuracy of all the classifiers.

Table 1. Results of word N-gram using TF feature representation with Unigrams and Bigrams.

Classifier	Accuracy	Precision	Recall	F-Measure
LSVM	**0.844**	0.849	0.929	**0.887**
LR	0.841	0.846	0.928	0.885
RF	0.824	0.828	0.927	0.874
NB	0.840	**0.897**	0.865	0.881
KNN	0.685	0.678	**0.998**	0.807

Table 2. Results of word N-gram using TF-IDF feature representation with Unigrams and Bigrams.

Classifier	Accuracy	Precision	Recall	F-Measure
LSVM	**0.849**	**0.866**	0.919	**0.890**
LR	0.846	0.862	0.907	0.886
RF	0.820	0.841	0.898	0.869
NB	0.844	0.836	**0.952**	0.889
KNN	0.825	0.834	0.918	0.874

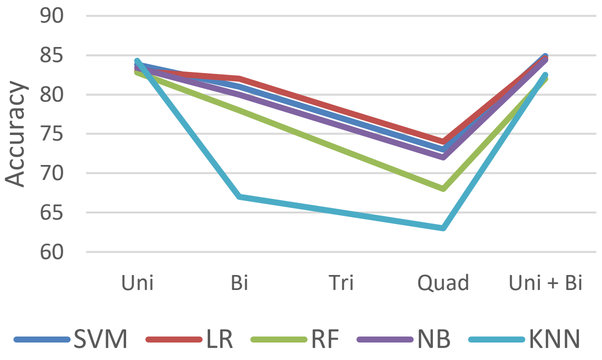


Figure. 3 The Accuracy of different classifiers using TF-IDF features with different values of N

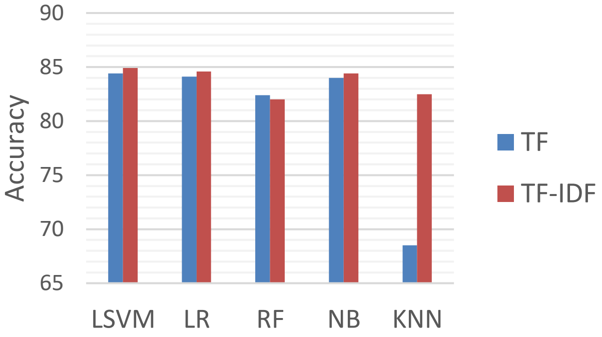


Figure. 4 Accuracy comparison between TF and TF-IDF feature extraction techniques

Furthermore, using TF-IDF features enhance the performance of the model more than using TF features with four of the selected classifiers. As described earlier, TF measures how important a word is to a tweet while TF-IDF measures how important a word is to a tweet in a collection (dataset) of tweets. In other words, TF-IDF figures out which words are important representative words for this tweet.

Fig. 4 illustrates a comparison between the two techniques in terms of accuracy. As shown in the figure, TF-IDF outperformed TF using LSVM, LR, NB, and KNN classifiers. RF is the only classifier that performs better with TF features while KNN achieved a noticeable increase in performance with TF-IDF features.

We also changed the number of top selected features ranging from 500 to 50,000 and record the results. We observed a direct relationship between the performance and the number of selected features. The

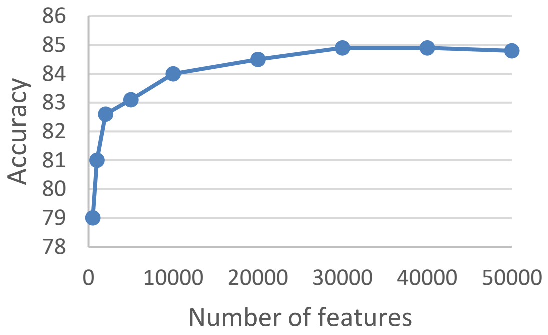


Figure. 5 Effect of increasing number of top selected features on accuracy

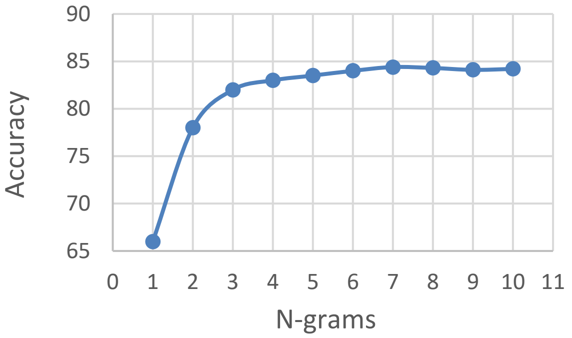


Figure. 6 Effect of increasing character N-grams on accuracy of LSVM classifier

best results were achieved when we select 30,000 features then the performance stabilized at this point since tweets are limited by 140 characters per tweet. Fig. 5 shows the relation between the number of selected features and the accuracy rate of the proposed model using LSVM as a classifier and TF-IDF feature representation.

To complete the experiment, we investigated the use of character N-gram instead of the word N-gram. We recorded the accuracy of LSVM classifier with a range of 1:10 character grams. We started the experiment with a series of one-character N-gram which achieved an accuracy rate of 66%. By increasing the size of N, the accuracy increases till it reaches 84.4% with a series of 1:7 characters. Fig. 6 shows the effect of increasing character N-grams on the accuracy of the LSVM classifier.

We noticed that choosing a large character N-gram range achieved approximately the same accuracy rate as word N-grams but significantly increased the detection time. Table 3 shows a comparison between the accuracy of the LSVM and the computation time in seconds for 5802 tweets using word and character N-gram with TF-IDF feature representation.

Table 3. Comparison between word and character N-grams

N-grams	Accuracy	Computation time in seconds
Unigrams + Bigrams	0.849	0.453
Characters 1:7	0.844	6.835

Table 4. Results of the PHEME dataset using feature-based [18] and the proposed N-gram based models

Model	Accuracy	Precision	Recall	F-Measure
Feature based	0.784	0.796	0.916	0.852
N-gram based	**0.849**	**0.866**	**0.919**	**0.890**

Our previous research in Twitter credibility [18] relies on a set of features extracted from the tweet itself and from the author of the tweet to evaluate the credibility of any given tweet. The feature set that was used contains 17 content-based features and 15 source-based features. Some of the features were computed like followers to friends ratio while others were extracted from the author's history such as the mean of URLs and the average number of retweets. The best results were achieved using a combination of the content and source features and Random Forests as a classifier. The research applied 10-fold cross validation on the PHEME dataset which is used in this work as well. Table 4 illustrates a comparison between the two models in terms of Accuracy, Precision, Recall, and F-measure. The comparison shows that the proposed N-gram model outperformed the feature-based model when applied to the PHEME dataset.

4.2 CAT dataset

In this section, we review the results of applying our model on the Arabic dataset (CAT) [8]. First, we remove hashtags #, URLs, emoticons and all not needed text. Then, TF and TF-IDF features were extracted to prepare the dataset for training. We run the five classifiers on the dataset using different sizes of N-grams. As resulted from applying our model on PHEME dataset, using a combination of both Unigrams and Bigrams achieved the best performance with CAT dataset.

Tables 5 and 6 show the Accuracy, Precision, Recall, and F-measure results of the proposed model using both Unigrams and Bigrams with TF and TF-IDF feature representations. As shown in Table 5, NB classifier achieves the best Accuracy, Precision and

Table 5. Results of word N-gram using TF feature representation with Unigrams and Bigrams

Classifier	Accuracy	Precision	Recall	F-Measure
LSVM	0.707	0.748	0.779	0.763
LR	0.699	0.744	0.768	0.755
RF	0.709	0.737	**0.808**	**0.771**
NB	**0.722**	**0.769**	0.774	**0.771**
KNN	0.633	0.725	0.640	0.677

Table 6. Results of word N-gram using TF-IDF feature representation with Unigrams + Bigrams

Classifier	Accuracy	Precision	Recall	F-Measure
LSVM	**0.732**	**0.764**	0.807	**0.785**
LR	0.704	0.757	0.755	0.756
RF	0.702	0.739	0.786	0.762
NB	0.711	0.681	**0.908**	0.778
KNN	0.698	0.708	0.855	0.774

Table 7. Results of the CAT dataset using feature-based [18] and the proposed N-gram based models

Model	Accuracy	Precision	Recall	F-Measure
Feature based	0.700	0.748	0.769	0.747
N-gram based	**0.732**	**0.763**	**0.809**	**0.785**

F-Measure with TF features while RF achieves the best Recall. LSVM recorded the best results with TF-IDF features in terms of Accuracy, Precision, and F-Measure. The best Recall was achieved by NB classifier as shown in Table 6.

In order to compare the feature-based model introduced in [18] and the proposed N-gram model, we applied the two models on the CAT dataset. Table 7 illustrates a comparison between the two models in terms of accuracy, precision, recall, and F-measure. The comparison shows that the proposed N-gram model outperformed the feature-based model when applied to the CAT dataset as well.

5. Evaluation

In this section, we compare the performance of the proposed model with two models existing in the literature. The first model was introduced by Zubiaga et al. [17] and relies on content and source-based features. The research depends on the textual features such as word vectors and Part of speech tags but ignored the relationships between words. The research applied CRF Conditional Random Fields as a classifier. We applied five-fold cross validation on the PHEME dataset which is used by Zubiaga et al. in order to achieve fair comparison. We used a combination of both unigrams and bigrams with TF-IDF features and LSVM as a classifier in this comparison. The results in Table 8 show that the proposed model outperformed CRF by 19%, 36%, 28% in precision, recall, and F-measure respectively. It is well-known that the relations between words are very important for language modeling. Our intuition is that the proposed model outperforms CRF because the inclusion of bigrams with unigrams which can capture the discriminative power of words as phrases.

Table 8. Comparison between the proposed model and Zubiaga et al [17]

Classifier	Precision	Recall	F-Measure
CRF [17]	0.667	0.556	0.607
The Proposed model	**0.861**	**0.919**	**0.889**

Moreover, we aim to compare our work with another work proposed by Ajao et al [22]. The research applied deep learning approach using long short-term memory (LSTM) to predict the tweets' credibility and detect fake Twitter posts. The authors applied 10-fold cross validation on the PHEME (5802 tweets) dataset and achieved 82% accuracy rate. Table 9 depicts the accuracy, precision, recall, and F-measure of both LSTM proposed by [25] and the N-gram based model presented in this paper. As shown in Table 9, N-gram based model outperforms LSTM when classifying tweets by 2%, 42%, 51% and 48% in terms of accuracy, precision, recall, and F-measure respectively.

In fact, deep learning approaches require large amounts of labeled data for the perfect training. As Chen Su et al. [29] observed that the performance of deep learning models increase logarithmically as the training dataset increases, we think that 5802 tweets are not enough to learn features directly from the data without the need for manual feature extraction. This observation is proven to be true when compared with the results of the proposed model which proved that N-gram models can perform better for small datasets.

Table 9. Comparison between the proposed model and Ajao et al. [22]

Classifier	Accuracy	Precision	Recall	F-Measure
Ajao et al. [22]	0.822	0.443	0.405	0.405
The Proposed model	**0.849**	**0.866**	**0.919**	**0.890**

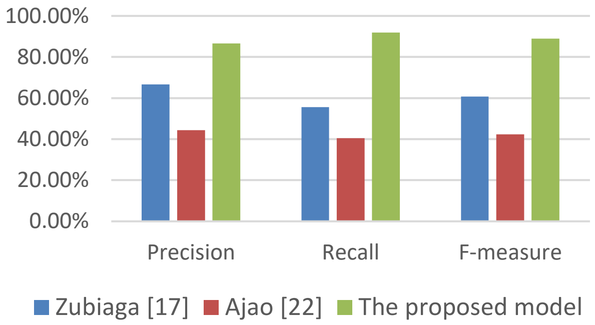


Figure. 7 Results Compared with Zubiaga et al.[17] and Ajao et al. [22]

Fig. 7 shows a comparison between the models presented in [17, 22] and the proposed model. The figure indicates that the proposed N-gram model outperforms the other models in terms of precision, recall, and F-measure.

6. Conclusion and future work

In this work, we focused on the problem of detecting the credibility of Twitter messages using text-based features. We have presented a credibility detection model based on N-gram analysis and investigated two different feature extraction techniques. The obtained results indicated that word N-gram features are more relevant compared with content and source-based features, also compared with character N-gram features. Linear classifiers as LSVM and logistic regression are more suitable for this problem. Best results were achieved using a combination of unigrams and bigrams, 30000 TF-IDF extracted features and LSVM as a classifier. The proposed model achieved 84.9% accuracy, 86.6% precision, 91.9% recall, and 89% F-measure over the PHEME dataset. For the CAT dataset, the model achieved 73.2% Accuracy, 76.4% Precision, 80.7% Recall, and 78.5% F-Measure. The evaluation shows higher performance of the proposed model in comparison with three different models existing in the literature using the same dataset.

As a future work, we plan to train the N-gram model on a larger dataset to increase the robustness of the model. We think that applying dimensionality reduction techniques to keep only the most relevant features from 30000 features set will improve the model performance. Regarding the online mobile application, we think it will be better if the output is presented to the user as a continuous score instead of the binary classification.

References

[1] A. Stocker, A. Richter, and K. Riemer. “A Review of Microblogging in the Enterprise”, *it-Information Technology Methoden und innovative Anwendungen der Informatik und Informationstechnik* Vol. 54, No. 5, pp. 205-211, 2012.

[2] “Twitter by the numbers: Stats, Demographic & Fun Facts.” [Online]. Available: *https://www.omnicoreagency.com/twitter-statistics/* [Accessed: 17-10-2019]

[3] X. Lu, and C. Brelsford. "Network structure and community evolution on twitter: human behavior change in response to the 2011 Japanese earthquake and tsunami.", *Scientific reports* 4, p. 6773, 2014.

[4] Á. Cuesta, D. Barrero, and M. R-Moreno. "A descriptive analysis of twitter activity in spanish around boston terror attacks." *In International Conference on Computational Collective Intelligence*, pp. 631-640. Springer, Berlin, Heidelberg, 2013.

[5] T. Sakaki, M. Okazaki, and Y. Matsuo. "Earthquake shakes Twitter users: real-time event detection by social sensors." In *Proceedings of the 19th international conference on World wide web*, pp. 851-860. ACM, 2010.

[6] H. Allcott, and M. Gentzkow. "Social media and fake news in the 2016 election." *Journal of economic perspectives* Vol. 31, No. 2, pp. 211-36, 2017.

[7] Z. Ashktorab, C. Brown, M. Nandi, and A. Culotta. "Tweedr: Mining twitter to inform disaster response." In *ISCRAM*, 2014.

[8] R. El Ballouli, W. El-Hajj, A. Ghandour, S. Elbassuoni, H. Hajj, and K. Shaban. "CAT: Credibility Analysis of Arabic Content on Twitter." In *Proceedings of the Third Arabic Natural Language Processing Workshop*, pp. 62-71, 2017.

[9] A. Gupta, H. Lamba, P. Kumaraguru, and A. Joshi. "Faking sandy: characterizing and identifying fake images on twitter during

hurricane sandy." In *Proceedings of the 22nd international conference on World Wide Web*, pp. 729-736. ACM, 2013.

[10] M. Mendoza, B. Poblete, and C. Castillo. "Twitter under crisis: Can we trust what we RT?" In *Proceedings of the first workshop on social media analytics*, pp. 71-79. ACM, 2010.

[11] Z. Jin, J. Cao, Y. Jiang, and Y. Zhang. "News credibility evaluation on microblog with a hierarchical propagation model." In *2014 IEEE International Conference on Data Mining*, pp. 230-239. IEEE, 2014.

[12] C. Castillo, M. Mendoza, and B. Poblete. "Information credibility on twitter." In *Proceedings of the 20th international conference on World wide web*, pp. 675-684. ACM, 2011.

[13] C. Castillo, M. Mendoza, and B. Poblete. "Predicting information credibility in time-sensitive social media." *Internet Research* Vol. 23, No. 5, pp. 560-588, 2013.

[14] A. Gupta, and P. Kumaraguru. "Credibility ranking of tweets during high impact events." In *Proceedings of the 1st workshop on privacy and security in online social media*, p. 2, ACM, 2012.

[15] K. Lorek, J. Suehiro-Wiciński, M. Jankowski-Lorek, and A. Gupta. "Automated credibility assessment on Twitter." *Computer Science* Vol. 16, No. 2, pp. 157-168, 2015.

[16] J. O'Donovan, B. Kang, G. Meyer, T. Höllerer, and S. Adalii. "Credibility in context: An analysis of feature distributions in twitter." In *2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing*, pp. 293-301. IEEE, 2012.

[17] A. Zubiaga, M. Liakata, and R. Procter. "Exploiting context for rumour detection in social media." In *International Conference on Social Informatics*, pp. 109-123. Springer, Cham, 2017.

[18] N. Hassan, W. Gomaa, G. Khoriba, and M. Haggag. "Supervised Learning Approach for Twitter Credibility Detection." In *2018 13th International Conference on Computer Engineering and Systems (ICCES)*, pp. 196-201. IEEE, 2018.

[19] S. Sabbeh, and S. Baatwah. "Arabic news credibility on twitter: an enhanced model using hybrid features.", *journal of theoretical & applied information technology* Vol. 96, No. 8, 2018.

[20] S. Ravikumar, R. Balakrishnan, and S. Kambhampati. "Ranking tweets considering trust and relevance." In *Proceedings of the Ninth International Workshop on Information Integration on the Web*, p. 4. ACM, 2012.

[21] H. Al-Khalifa, and R. Al-Eidan. "An experimental system for measuring the credibility of news content in Twitter.". *International Journal of Web Information Systems* Vol. 7, No. 2, pp.130-151, 2011.

[22] O. Ajao, D. Bhowmik, and S. Zargari. "Fake news identification on twitter with hybrid cnn and rnn models." In *Proceedings of the 9th International Conference on Social Media and Society*, pp. 226-230. ACM, 2018.

[23] T. Wu, S. Liu, J. Zhang, and Y. Xiang. "Twitter spam detection based on deep learning." In *Proceedings of the australasian computer science week multiconference*, p. 3. ACM, 2017.

[24] X. Zhang, J. Zhao, and Y. LeCun. "Character-level convolutional networks for text classification." In *Advances in neural information processing systems*, pp. 649-657, 2015.

[25] H. Ahmed, I. Traore, and S. Saad. "Detection of online fake news using N-gram analysis and machine learning techniques." In *International Conference on Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments*, pp. 127-138. Springer, Cham, 2017.

[26] M. Nieuwenhuis, and J. Wilkens. "Twitter text and image gender classification with a logistic regression n-gram model." In *Proceedings of the Ninth International Conference of the CLEF Association (CLEF 2018)*, 2018.

[27] *SCIKIT-LEARN* Machine Learning in Python http://scikit-learn.org/stable/

[28] https://figshare.com/articles/PHEME_dataset_of_rumours_and_non-rumours/4010619.

[29] C. Sun, A. Shrivastava, S. Singh, and A. Gupta. "Revisiting unreasonable effectiveness of data in deep learning era." In *Proceedings of the IEEE international conference on computer vision*, pp. 843-852, 2017.