2018 IEEE International Conference on Information Reuse and Integration for Data Science

Identifying Tweets with Fake News

Saranya Krishnan, Min Chen
Division of Computing and Software Systems, School of STEM
University of Washington Bothell
Bothell, USA
e-mail: {sarank, minchen2}@uw.edu

***Abstract*—The extensive use of social media has tremendous impact on culture, business, and politics on the world at large with potentially positive and negative effects. For example, social media coverage of crisis events may be used by authorities for effective disaster management or by malicious entities to spread rumors and fake news for financial or political benefit. Considering the harmful consequences of fake news in social media, there is a profound need to detect false information, control and/or prevent it from spreading. In this paper, we propose an advanced framework to identify tweets with fake news contents using techniques including statistical analysis of Twitter user account, reverse image searching, cross verification of fake news sources, and data mining. Experimental results on a large miscellaneous events dataset demonstrate the effectiveness of our proposed approach in identifying fake tweets.**

Keywords-fake news identification; tweet processing, data mining, reverse image search

I. INTRODUCTION

Nowadays, online social media plays a vital role in real world applications with potentially positive and negative effects. For example, many companies are using social media to advertise their products and build customer loyalty. Social media websites have also played an important role in many elections around the world.

Specifically, Twitter has been widely used during emergencies, such as wildfires [5] [9] [20] and earthquakes [6]. Journalists have hailed the immediacy of the service that allowed "to report breaking news quickly – in many cases, more rapidly than most mainstream media outlets" [12]. However, it was also reported in [4] that rumors propagated via tweets in a different way than the actual news. During hurricane Sandy, 86% of tweets spreading fake images were retweets and there were very few original tweets [7]. As a result, a user study in [16] showed users tend to take news less credible when presented on Twitter compared to traditional media websites or blogs.

There have been research efforts to detect fake news contents in Twitter and other social media. Tweet text characteristics are used in [2] to identify fake tweets. However, while tweets are basically short texts and text analysis is traditionally performed with Natural Language Processing (NLP) techniques, NLP tools alone may not deal with tweets properly because tweets often do not follow even the simplest and most basic syntactic rules [8]. In [14], a dataset of fake news source is summarized which is a useful resource but the list is not exhaustive [14]. In [15], a web application is presented to check the credibility of a tweet and detect malicious users by using reverse image search, user analysis and crowd sourcing. It shows a new and feasible direction in validating tweets but it does not incorporate tweet's text characteristics. Different from the existing work, we take into consideration of not only tweets' user account statistics, text characteristics, but also the media contents using data mining techniques. In addition, according to the impact of source and sentiment over source credibility and information credibility studied in [3], we conduct tweet sentiment analysis and cross verify tweets over fake news sources. Furthermore, many essential components are deployed as RESTful web services that can be called from any user applications, allowing the users to extend functionality as required. Meanwhile, web-UI is developed for general public to validate tweet credibility and view associated details via web browsers.

The rest of this paper is organized as follows. In Section II, the overall framework architecture is presented. Section III introduces more research details of the proposed framework. In Section IV, experimental results and analysis are presented. Finally, the last section summarizes this paper.

II. FRAMEWORK ARCHITECTURE

As shown in Fig. 1, the proposed framework consists of two major components: core and website. Core is responsible for fetching the tweet content, constructing the feature set using the tweet dataset provided by the developer, building the classifier and generating the evaluation report. Website is to provide tweet credibility predictions, tweet details and crowd sourcing results to the end user.

978-1-5386-2659-7/18/$31.00 ©2018 IEEE
DOI 10.1109/IRI.2018.00073

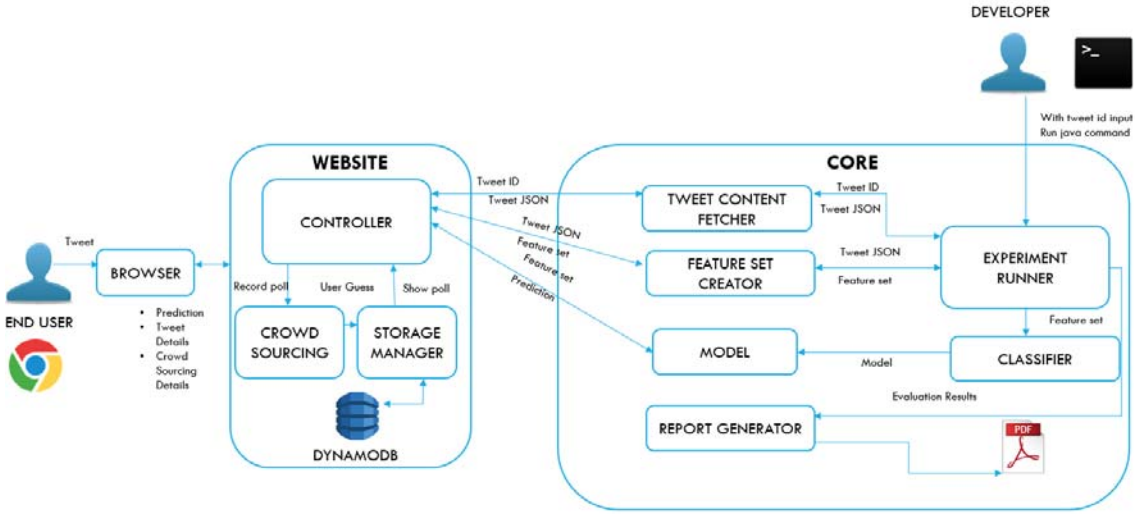

Figure 1. Overall framework architecture

A. Core

To fulfill the required functions, the core component consists of four major modules, namely experiment runner, tweet content fetcher, feature set creator, classifier and report generator.

- Experiment runner: it is the entry point to the core component which receives tweet ID's and orchestrates the whole core process by interacting with other components to create the feature set, run the classifier and finally generate the report.
- Tweet content fetcher: it is responsible for fetching the tweet content from Twitter using Twitter API and outputting the tweet content as JSON.
- Feature set creator: it generates the feature set from the JSON tweet content for classification.
- Classifier and report generator: it performs classification on the feature set and generating evaluation report.

We will discuss more research details about feature extraction and classification in Section III.

B. Website

The website is for the end user to supply the tweet URL via browser, and to receive the associated tweet details and system's classification on tweets (as fake or real). It consists of three major components: controller, crowd sourcing and storage manager.

- Controller: it is responsible for the interaction between website and core components. It passes the tweet URL submitted by users to the tweet content fetcher in core component and receives predictions and tweet details from the core for display.
- Crowd sourcing: it records and shows users' guesses (polls) for particular tweets, and passing such polls along with associated tweet ID to storage. Currently crowd source data are collected but have yet been incorporated in refining the framework performance, which is one of our future work items.
- Storage Manager: handling interactions between crowd sourcing, controller and database component (AWS DynamoDB is used in this study) to store and update crowd-sourcing data.

III. Research Details

For a given tweet set $T = \{T_1, T_2, \ldots, T_n\}$, the goal is to assign one of the labels in the label set $L = \{Real, Fake\}$ to each tweet T_i. To fulfill this task, various tweet features are extracted for T_i and then the feature vector is passed to data mining component for classification.

A. Feature extraction

As discussed earlier, our framework analyzes not only tweets' user account statistics, text characteristics, but also media contents. Correspondingly, for any given tweet *Ti*, its user features and content features are extracted. Here user features refer to the property of the Twitter user *Ui* who issued the tweet *Ti* while content features capture the content representations of *Ti*.

As shown in Table I, seven user features are extracted in our study. Here,

- *noOfFriends* refers to the number of friends for the Twitter user U_i;
- *noOfFollowers* is U_i's number of followers;

- *friendFollowerRatio* is the ratio between U_i's *noOfFriends* and *noOfFollowers*;
- *noOfTimesListed* is the number of times U_i has listed;
- *isUserHasURL* is whether U_i has URL;
- *isVerifiedUser* shows whether U_i is a verified user; and
- *noOfTweets* lists number of tweets U_i has posted.

TABLE I. LIST OF USER FEATURES

User Features	
noOfFriends	noOfFollowers
friendFollowerRatio	noOfTimesListed
isUserHasURL	isVerifiedUser
noOfTweets	

All these user features can be directly obtained from the Twitter's metadata that is retrieved by the tweet content fetcher module discussed in Section II.A.

TABLE II. LIST OF CONTENT FEATURES

Content Features	
isUrlCredible	isImageCredible
sentimentScore	tweetLength
wordCount	noOfQuestionMark
noOfExclamationMark	containsQuestionMark
containsExcalmationMark	noOfUpperCaseLetter
noOfhasgTags	noOfUrls
noOfRetweets	

Similarly, as showing in Table II, many of the content features that show the tweet's text characters are also available in the metadata. These include tweetLength (number of characters in the tweet), wordCount (number of word), noOfQuestionMark (number of question marks), noOfExclamationMark (number of exclamation marks), containsQuestionMark (whether question mark is presented), containsExclamationMark (whether exclamation mark is presented), noOfUpperCaseLetter (number of upper case letters), noOfhasgTags (number of hash tags), noOfUrls (number of URLs), noOfRetweets (number of retweets). In contrast, more processing steps are required to achieve the features of isUrlCredible (whether the URLs in the tweet are valid), isImageCredible (whether the images in the tweet are credible), and sentimentScore (tweet's sentiment score).

1) isUrlCredible. To find whether the URL(s) presented in the tweet content is credible, cross verification is conducted against the fake news sources provided by [14]. This dataset contains text and metadata scrapped from 244 websites tagged as fake by the BS Detector Chrome extension [17]. The data is pulled using webhose.io API and used to verify the URL domain presented in the tweet content. isUrlCredible feature is set to "No" if a match is identified in the fake news sources and set to "Yes" otherwise.

2) isImageCredible. The algorithm in Table III is developed to set the isImageCredible feature values. As shown in Table III, isImageCredible is initially set to "Yes" (step 1). Google reverse image search is then performed. It has a feature called "Pages that include matching images" that takes each image in tweet Ti as query and return webpages P containing such image (steps 2 and 3). isImageCredible is changed to "No" if any of the webpages P is in the fake news sources [14] or if the page creation date and the tweet creation date are far apart (steps 4-7). Currently the threshold Th is empirically set to be 6 months but it may be adjusted in the future through data-driven analysis. If none of the condition becomes true, the "Best guess for this image" feature in Google Image is invoked to return textual query matching with the images (step 8), which is then checked against the tweet text to see whether they are related using Python genism library (step 9) [13]. isImageCredible is changed to "No" if they are not related (step 10).

TABLE III. PSEUDO CODE TO SET ISIMAGECREDIBLE

INPUT: Image set I and word set W in tweet T_i, T_i creation time t, fake news source F
OUTPUT: T_i's *isImageCredible* feature is set to Yes or No

```
1.  set isImageCredible = Yes
2.  for each image i in I
3.     perform Google reverse image search to retrieve pages
       P containing matching image
4.     if P ∩ F ≠ ℰ
5.        set isImageCredible = No; break;
6.     else if |P's creation time – t| ≥ Th
7.        set isImageCredible = No; break;
8.     else perform Google best guess to retrieve texture
       descriptions T for i
9.        if T and W are not related
10.           set isImageCredible = No; break
11.       end if;
12.    end if;
13. end for;
```

3) sentimentScore. Sentiment analysis is the process of computationally determining whether a piece of writing is positive, negative or neutral by assigning a sentiment score between -1.0 (most negative) to 1.0 (most positive). It is analyzed and computed by using TextBlob and NLTK corpora. Natural Language Toolkit (NLTK) [1] is a leading platform to work with human language data. NLTK provides library to compute sentiment score but is very primitive and hard to use. TextBlob [10] is a wrapper on top of NLTK library.

To compute sentimentScore feature values, firstly the tweet is cleaned to remove all hyperlinks, special characters, etc. using simple regex. Secondly, a TextBlob object from the tweet text is created using the TextBlob library as follows:

- The text in the tweet is tokenized into tokens (words).

- The stopwords are removed from the list of tokens. Here the stopwords refer to the commonly used words that are irrelevant in text analysis like I, am, you, are, etc.
- Significant features/tokens like adjectives, adverbs, etc. are tagged and selected to pass into sentiment classifier (Naïve Bayes Classifier in TextBlob) to be classified as positive, negative or neutral by assigning a polarity between -1.0 to 1.0. A tweet is considered positive in nature if its polarity is greater than 0, negative if it is less than 0, and neutral if it is 0.

B. *Classification*

Classifier module discussed in Section II.A is responsible for classifying tweets as fake or real based on the feature set extracted by the feature set creator. In this study, J48 decision tree classifier and support vector machines (SVM) are used to train the models and perform classification. The reason that J48 is chosen is that in [2], the study found J48 outperforms other classifiers on identifying fake tweets in hurricane Sandy and Boston marathon datasets. However, no experiment has been conducted on datasets with miscellaneous events, which will be explored in our study. SVM is widely accepted as the baseline classifier, especially for binary classification problems [18][19] that fits our needs well, i.e., to classify tweets as either fake or real. In particular, the sequential minimal optimization (SMO) algorithm proposed by [11] is used, which has been implemented in WEKA machine learning library.

IV. EXPERIMENTS

To demonstrate the effectiveness of our proposed framework for multimedia data classification, it is tested on the dataset provided by [2]. The dataset comprises of tweets on hurricane Sandy event and miscellaneous events including MH370, Boston marathon, Paris attack and Russia air strike on Syria, etc. It has 5,349 English tweets with images or URLs that are accessible while others are either deleted or marked as private by their owners. Among them, 3,812 are hurricane Sandy tweets and 1,537 miscellaneous events tweets. As discussed earlier, the content and user features listed in Tables I and II are first extracted, which are then passed into classifier module for classification.

A. *Evaluation metrics*

Three performance metrics are used in this project to evaluate the classifiers: precision, recall, and F-measure. As defined in the equation below, precision is defined as the fraction of relevant instances among all retrieved instances. Recall is the fraction of the retrieved relevant instances over the total number of relevant instances in the dataset. F-measure is defined as the weighted harmonic mean of the precision and recall of the test. It captures the trade-offs between precision and recall and is considered an objective and ultimate quality metric of a classifier.

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{F-measure} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

Here, *TP* stands for true positive, i.e., the fake tweets are correctly predicted as fake in our study, *FP* is false positive where real tweets are misclassified as fake, and *FN* is false negative where fake tweets are mislabeled as real.

B. *Experiments using k-fold cross-valuation*

In the experiments, *k*-fold cross-validation scheme [7] is used, where the dataset is divided into *k* subsets, and the holdout method is repeated *k* times. Each time, one of the *k* subsets is used as the test set and the other *k*-1 subsets are grouped to form a training set. Then the average error across all *k* trials is computed. The advantage of this method is that it matters less how the data are divided. Every data point gets to be in a test set exactly once, and in a training set *k*-1 times. The variance of the resulting estimate is reduced as *k* is increased. In our study, *k* is set to 10 that is the most widely used cross-validation setup.

TABLE IV. 10-FOLD CROSS-VALIDATION RESULTS

Hurricane Sandy

Classifier	TP Rate	FP Rate	Precision	Recall	F-Measure
J48	0.923	0.368	0.842	0.923	0.881
SVM	0.999	0.468	0.819	0.999	0.900

Miscellaneous Event

Classifier	TP Rate	FP Rate	Precision	Recall	F-Measure
J48	0.825	0.320	0.787	0.825	0.805
SVM	0.801	0.480	0.706	0.801	0.750

Hurricane sandy + Miscellaneous Event

Classifier	TP Rate	FP Rate	Precision	Recall	F-Measure
J48	0.871	0.410	0.801	0.871	0.834
SVM	0.965	0.593	0.755	0.965	0.847

The experiments are conducted on 1) the dataset containing only hurricane Sandy, 2) dataset containing only miscellaneous events, and 3) dataset combining both hurricane Sandy and miscellaneous events. The results are listed in Table IV. Compared to the accuracy results (80.68% for hurricane Sandy and 81.25% for Boston Marathon using J48 tree) reported in [2], our performance is better in predicting fake tweets with high recall without sacrificing too much on precision value. In fake tweet identification, recall is often considered more important than precision. It is because we want to control/prevent all (if possible) fake tweets from spreading so it is important to locate fake tweets as extensively as possible even at a cost of misidentifying a few credible tweets as fake. After the classifier retrieves all possible results, users can then double check them to confirm. In addition, the F-measure

values are also high especially when tested on a single event like hurricane Sandy.

C. *Experiments using independent training/testing set*

We also perform experiments using independent training/testing set with the goal to test our model's extensibility on tweets from different events/domains. Specifically, we train the model using hurricane Sandy data and test it on miscellaneous events data, and vice versa. The results are shown in Table V.

To our best knowledge, no existing work has used this experimental setup on the same dataset for us to compare the performance. However, as we can see, the recall values in most cases remain higher than precision and F-measure is in line with what reported in [4] where training and testing are performed on the same dataset.

TABLE V. INDEPENDENT TRAINING/TESTING RESULTS

Hurricane Sandy for training and Miscellaneous event for testing

Classifier	TP Rate	FP Rate	Precision	Recall	F-Measure
J48	0.907	0.796	0.621	0.907	0.737
SVM	0.937	0.897	0.600	0.937	0.732

Miscellaneous event for training and Hurricane Sandy for testing

Classifier	TP Rate	FP Rate	Precision	Recall	F-Measure
J48	0.533	0.373	0.752	0.533	0.624
SVM	0.729	0.756	0.672	0.729	0.699

V. CONCLUSIONS AND FUTURE WORK

In this paper, we present a generalized framework to predict tweet credibility. First, the essential content features and user features of tweets are extracted through Twitter API. If a tweet has an image or image URL, the reverse image search is performed to check whether the same image has been tagged with different information in the past. In addition, if any URL is presented in the tweet, it will be cross-checked against the fake news sources to see whether it is part of the fake websites dataset. All these features are then used by the data mining algorithm to classify tweets as fake or real. The experiments demonstrate the effectiveness of this proposed framework. In the future, the crowd sourcing data currently collected and stored in DynamoDB may be used to feed into the algorithm design and classifier re-training to further improve the framework performance.

REFERENCES

[1] S. Bird and E. Loper, "Nltk: the natural language toolkit," Proc. ACL 2004 on Interactive Poster and Demonstration Sessions, Association for Computational Linguistics, 2004, p. 31.

[2] C. Boididou, S. Papadopoulos, Y. Kompatsiaris, S. Schifferes, and N. Newman, "Challenges of computational verification in social multimedia," Proc. 23rd International Conference on World Wide Web, 2014, pp. 743–748.

[3] K. Byrum, "A comparison of the source, media format, and sentiment in generating source credibility, information credibility, corporate brand reputation, purchase intention, and social media engagement in a corporate social responsibility campaign presented via social media," All Dissertations, 2014, 1312.

[4] C. Castillo, M. Mendoza, and B. Poblete, "Information credibility on twitter," Proc. 20th International Conference on World Wide Web, 2011, pp. 675–684.

[5] B. De Longueville, R. S. Smith, and G. Luraschi, "Omg, from here, I can see the flames!: a use case of mining location based social networks to acquire spatio-temporal data on forest fires," Proc. 2009 International Workshop on Location Based Social Networks, 2009, pp. 73–80.

[6] P. Earle, M. Guy, R. Buckmaster, C. Ostrum, S. Horvath, and A. Vaughan, "Omg earthquake! can twitter improve earthquake response?" Seismological Research Letters, vol. 81, no. 2, pp. 246–251, 2010.

[7] A. Gupta, H. Lamba, P. Kumaraguru, and A. Joshi, "Faking sandy: characterizing and identifying fake images on twitter during hurricane Sandy," Proc. 22nd International Conference on World Wide Web, 2013, pp. 729–736.

[8] H. Hamdan, P. Bellot, and F. Bechet, "Lsislif: Feature extraction and label weighting for sentiment analysis in twitter," Proc. 9th International Workshop on Semantic Evaluation, Association for Computational Linguistics, 2015, pp. 568–573.

[9] A. L. Hughes, and L. Palen, "Twitter adoption and use in mass convergence and emergency events," Inter. J. Emergency Management, vol. 6, no. 3-4, pp. 248–260, 2009.

[10] S. Loria, P. Keen, M. Honnibal, R. Yankovsky, D. Karesh, E. Dempsey, et al., "Textblob: simplified text processing," Secondary TextBlob: Simplified Text Processing, 2014.

[11] J. Platt, "Fast training of support vector machines using sequential minimal optimization," Adv. Kernel Methods - Support Vector Learning, B. Schoelkopf, C. Burges, and A. Smola, Eds. MIT Press, 1998.

[12] K. Poulsen, "Firsthand reports from california wildfires pour through twitter," Wired, 2007.

[13] Python genism library. https://radimrehurek.com/gensim/

[14] M. Risdal, Getting Real about Fake News. https://www.kaggle.com/mrisdal/fake-news.

[15] D. Saez-Trumper, "Fake tweet buster: a webtool to identify users promoting fake news on twitter," Proc. 25th ACM conference on Hypertext and Social Media, 2014, pp. 316–317.

[16] M. Schmierbach and A. Oeldorf-Hirsch, "A little bird told me, so i didn't believe it: Twitter, credibility, and issue perceptions," Communication Quarterly, vol. 60, no. 3, pp. 317–337, 2012.

[17] D. Sieradski, BS Detector. https://github.com/selfagency/bs-detector.

[18] P. Somwang and W. Lilakiatsakun, "Computer network security based on support vector machine approach," Proc. IEEE 2011 11th International Conference on Control, Automation and Systems (ICCAS), 2011, pp. 155–160.

[19] V. A. Sotiris, W. T. Peter, and M. G. Pecht, "Anomaly detection through a bayesian support vector machine," IEEE Trans. Reliability, vol. 59, no. 2, pp. 277–286, 2010.

[20] S. Vieweg, A. L. Hughes, K. Starbird, and L. Palen, "Microblogging during two natural hazards events: what twitter may contribute to situational awareness," Proc. ACM SIGCHI Conference on Human Factors in Computing Systems, 2010, pp. 1079–1088.