# Gossip Detection in Microblogging Platforms

**S.Rahila Ammara[1], Dr.J.Chockalingam[2]**
[1]Dept of Computer Science
[2]Associate Professor, Dept of Computer Science
[1, 2] Khadir Mohideen College, Adirampattinam

***Abstract-*** *Microblogging platforms facilitate fast and frequent communication among very large numbers of users. Gossips, especially in times of crisis, tend to spread quickly, causing confusion, and impair people's ability to make decisions. Hence, it is of utmost importance to automatically detect a Gossip as soon as possible. Using the PHEME dataset that contains Gossips and non- Gossips pertaining to five major events, we have developed a Gossip detection system that classifies posts from Twitter, a popular microblogging website. We have first analyzed and ranked a number of content-based and user-based features. Some content -based features are derived using natural language processing techniques. We then trained multiple machine learning models (Naive Bayes, Random Forests and Support Vector Machines) using different combinations of the features. Finally, we compared the performance of these models. The performance of the models on one such event resulted in 78% accuracy.*

***Keywords***- automatic Gossip detection, microblog, twitter, machine learning, feature engineering, social media

## I. INTRODUCTION

A microblog is a blogging platform where the length of the information broadcast is restricted. Twitter is one of the most popular public microblogging platforms. A microblog is a powerful platform to broadcast messages, send requests for emergency help, send notifications, or quickly inform a large number of users about a certain event. Although microblogs such as Twitter, are used mostly for benign purposes, they also lend themselves to being misused, by the spread of false information. The Merriam-Webster dictionary defines a Gossip as "talk or opinion widely disseminated with no discernible source". Due to the popularity of online microblogs, it is easy to start and quickly spread misinformation in the form of false Gossips. If large numbers of Gossips are frequently circulated, the veracity and authenticity of the information on the microblogging platform becomes questionable. Genuine cases in which people require help can be mistakenly ignored. Hence it is important to automatically detect Gossips on microblogs such as Twitter. Once a Gossip is detected, it is possible to stop the spread of the Gossip by several countermeasures, including corrections, or taking actions against users who start and disseminate such Gossips.

This paper is organized as follows. In Section II, we investigate related work that has been carried out in Gossip detection. In Section III, we provide an overview of the Twitter platform. In Section IV, we explain the proposed approach. In Section V, we present the experimental results. Finally, we conclude this paper in Section VI.

## II. RELATED WORK

In this section, we briefly summarize the results of existing Gossip identification research.

Bursty term identification and multi-dimensional sentence modelling are used to automatically detect emerging hot topics for Gossip identification. The burst-weight of a term is a function of 3 parameters i.e. skewness, timeliness score and periodicity score. The term- weighting scheme considers both frequency and topicality of terms to identify 'bursty' terms. Sentences are clustered based on hot terms and sentences with the largest number of 'bursty' terms are selected as clustering centroids. A 'bursty term' vector and 'named entity' vector are combined to calculate similarity between sentences. Gossips are identified based on 3 types of features i.e. content-based features, twitter-based features and network-based features. Several classifier models such as Naive Bayes, Decision Trees, Logistic Regression and Random Forest are run, with Random Forests giving the best results.

The social network is modelled as a directed graph and a small subset of nodes called monitors are selected and designated 'positive' if they received a Gossip and 'negative' otherwise. The Gossip source is expected to be close to positive monitors, but far from negative monitors. Nodes are ordered using four different metrics. Further, six methods for choosing monitor nodes, and their accuracy in identifying sources of Gossips are calculated. To identify Gossips, the 'Greedy Sources Set Size' and 'Maximal Distance of Greedy Information Propagation' algorithms are used. Logistic Regression is used for classification. The authors conclude that the 'Betweenness Centrality' method of selecting

monitors produces the best accuracy, and, that having a large number of positive monitors is not necessarily helpful. If Gossips and non-Gossips have very large difference in the number of sources, Gossip classification can be done with very high accuracy.

In Aker et al., a piece of text is classified based on the author's stance (attitude of the author as evident from the text). The Pheme dataset consists of tweets from 5 different events annotated with a stance. The second is the GossipEval dataset which is derived from the Pheme dataset. Decision Trees, Random Forest, and K-Nearest Neighbours classifiers were used. The authors use a wide range of features including Bag of Words, Part of Speech Tagging and Brown Clustering. They also propose a set of additional problem-specific features, known as the 'AF-Feature Set' which improves the performance. A cumulative vector of all the features is created. The Random Forest classifier performs best on the GossipEval dataset, with 79% accuracy, thus out -performing Turing. The J48 Decision Tree performs best on Pheme dataset. Also, the removal of the 'AF-Feature Set' decreases the accuracy by approximately 2.5%.

Gang et al. extracts features of a post based on the behaviour of the post's author, as well as the behaviours of those people who reply to this post. Examples of features are: 'verified user', 'number of followers' and 'number of followees'. Other features, like number of microblog sources for the information, and number of questioned comments on the post are also considered. Various graphs such as boxplots are constructed, in order to simplify the task of choosing the best subset of features. Each post (Gossip or non-Gossip) is then represented by a vector of these features. Classification is performed using K-Nearest Neighbours, Neural Networks, Naïve-Bayes, Decision Trees etc. Naïve-Bayes was found to be the best classifier.

Three categories of features (content based, network based and microblog specific memes) are used to (a) retrieve online microblogs that are Gossip related and to (b) identify tweets that contain support for the Gossip. Tweets are retrieved using Twitter API and manually annotated. The Kappa coefficient is used to measure inter-judge agreement. Bayes classifiers are used as high-level features and a linear function of these features is learned to solve both retrieval and classification tasks. Features calculated using the content language models achieved the best precision and recall for Gossip retrieval. For belief classification, good results are obtained when all features are used together,

Zubaiga et al provide a survey of current research in Gossip detection and resolution. They discuss two types of Gossips based on the length of circulation: long-standing Gossips and newly emerging Gossips that occur during fast-paced events. They provide an overview of the research in Gossip detection and classification. Current research trends in these four components viz. Gossip detection, Gossip tracking, Gossip stance classification and Gossip veracity classification, is presented.

Zubaiga et al. propose a sequential classifier that distinguishes between verified and unverified posts on social media. The authors hypothesize that aggregating Gossips and non-Gossips preceding the tweet will improve Gossip classification. In order to validate this hypothesis, the performance of sequential (Conditional Random Fields) and non-sequential (Maximum Entropy) classifiers are compared against each other. The authors have also compared the results with 3 non-sequential classifiers – Support Vector Machines, Naïve Bayes and Random Forests. The features used involved a combination of 'content -based' and 'social' features. From the results, it is concluded that when a combination of content-based and social features is used, the Conditional Random Fields classifier outperforms all the others in terms of precision and F1-score. Naïve Bayes was the best in terms of recall.

In research on Persian tweets,a graph is created representing the relationship between the follower-followee for all users. Content-based and user-features are evaluated using information gain and multiple models are used for classification. Some Persian words clearly show racism, negotiations and are used as features for Gossip detection. Since no natural language processing tool currently supports these, this experiment hasn't been totally exploited in the Persian environment.

## III. THE TWITTER PLATFORM

Twitter is an online social networking service, in which the primary mode of communication is in the form of short messages known as 'tweets'. The character limit for a tweet, enforced by Twitter, is now 280 characters. This restriction on tweet size, promotes clever use of language in order to effectively convey information. In the beginning of 2018, Twitter had 330 million active users on a monthly basis

Creating a Twitter account is free of cost. Once registered, users function as either broadcasters or receivers. Users can create tweets, and may also subscribe to other users' tweets, by 'following' other users. Personage tweets can be forwarded by other account owners to their own feed, a progression recognized as a 'retweet'. Users can also express their appreciation for a given tweet's content, by 'liking' the

tweet. Users can group posts together by topic or type by using 'hashtags' i.e. words prefixed with a '#' symbol. An account that has been verified by Twitter, is a 'Verified Account', and this ensures that the person or entity being represented by the account is legitimate.

People create tweets for a variety of reasons, some of which include vanity, self-promotion and attention and some to publish very useful content. Hence it becomes very difficult, if not impossible for an average

Twitter user, to discriminate between factual information, and baseless Gossips. This is precisely the problem that we wish to address through this paper.

## IV. PROPOSED METHOD

Our system is divided into three main phases, the analysis phase, the build phase and the operational phase. The analysis phase involves comparing features across all five events and finding the common traits of Gossips. During the build phase, these features are extracted and the machine learning models are trained on these feature vectors. The models are tested and tweets belonging to both testing data and unseen events are classified into Gossips or non-Gossips in the operational phase. The modules in the system design Fig. 2 are explained in detail.

**TABLE I.DATASET**

| Event Name | Gossips | Non-Gossips |
|---|---|---|
| charliehebdo | 2290 | 8105 |
| ferguson | 1420 | 4295 |
| germanwings-crash | 1190 | 1155 |
| ottawashooting | 2350 | 2100 |

A. Data extraction – Training data is obtained as described in Section IV (dataset). The features extracted from the dataset are a combination of direct features (from the tweet json) and derived features. The feature vector is obtained using a combination of these.

B. Pickle Generator – The entire dataset, containing all the tweets along with their attributes, must be stored in a form that is convenient to work with programatically.

Since the language of choice for this project was Python, the method chosen to accomplish this task, was pickling the dataset. Pickling is a process by which a hierarchy of Python objects, is converted into a byte stream, and stored on disk, so that it can be retrieved later, when required.

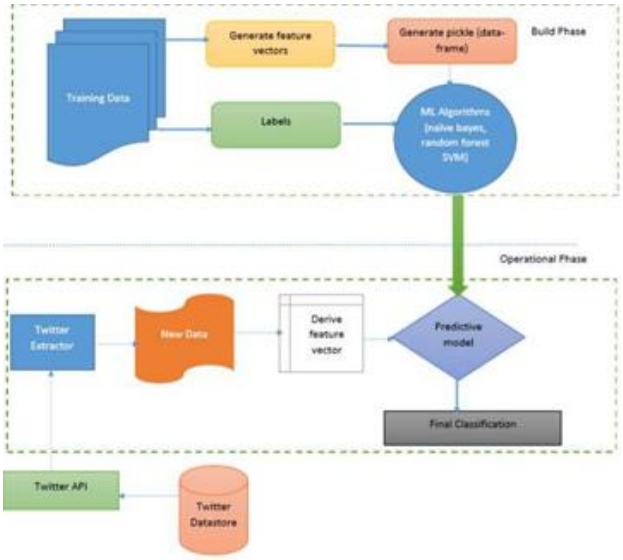


Fig. 1. SYSTEM DESIGN

C. Feature Analysis and Extraction –

The direct and derived (extracted) features can be divided into two categories viz. user features, content features.

1. User based features, are those that describe the characteristics of the user who posted the tweet. The rationale is that, users who post Gossips, have similar characteristics. We can build features out of these special characteristics and train the model to classify posts as Gossips or non-Gossips. Some direct user-based features include number of tweets, retweets, followers, friends of the user.
2. Content Based features, are those that abstract the tweet's content into a form that describes its text. The challenge is to extract meaningful Content Based features from the data, using techniques made available by Natural Language Processing, such that, the content of every tweet is well represented in the feature vector. These are the features extracted from the text posted by the user that captures the special characteristics of a Gossip. These special characteristics can be used as features to train machine learning models. For e.g. whether the tweet contains acronyms, bad-words, emoticons, question marks, exclamation mark, punctuation marks, dot-dot-dot are some features.

D. Classifiers – Three classifiers are used to build the Gossip detection engine. They are :

a) SVM: Support vector machines are a set of supervised learning algorithms used for classification, regression and outlier detection. SVMs are effective in high dimensional spaces and where the number of dimensions is greater than the number of samples. It is versatile, since different kernel functions can be specified for the decision function. The Radial Basis Function Kernel is used to perform Gossip classification.

b) Naïve Bayes: Naive Bayes (NB) methods are a set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of independence between every pair of features. In spite of their apparently over-simplified assumptions, NB classifiers have worked quite well in many real-world situations.

They require a small amount of training data and can be extremely fast compared to more sophisticated methods. The algorithm used for Gossip classification, is the Gaussian Naïve Bayes algorithm, as the data contains continuous variables.

TABLE II.FEATURE SCORES

| **Feature** | **Significant score (rank)** |
|---|---|
| User_followers_count | 8.159e+08 |
| User_friends_count | 1.466e+05 |
| Retweet_count | 1.15E+05 |
| Age_of_tweet | 4.47E+04 |

c) Random Forest Classifier: Random forests or random decision forests is an ensemble learning method for classification, regression and other tasks, that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes or mean prediction (regression) of the individual trees. This classifier uses multiple decision trees' vote to classify posts as Gossips or not. In this way, they correct the decision trees' habit of over-fitting data. The result can be a majority selection or based on some confidence percentage. We use the random forest classifier model provided by the 'scikit' library and initialise with multiple features throughout various runs.

E. Training/Validation - Each of the 3 classifiers is trained with 75% of the dataset. Initially, only user based features are used. The model is trained and tested for each subset of those user based features. For eg. the first model will have only the top 4 ranked user features. Next, the model is trained and tested on several combinations of content based features. Finally, the model is executed on a combination of user and content based feature.

F. Classification on a new dataset – Classification of posts pertaining to new events into Gossips and non-Gossips is done by first procuring the dataset and then running it through SVM, naïve bayes and random forest classifiers. The dataset is obtained by using the twitter API to get all posts related to that evemt and then by parsing it into JSON files.

## V. EXPERIMENTAL RESULTS

Each of the classifiers were run on different user based, content based features and their combinations. The PHEME dataset (of the 5 events) was split into 75% training and the rest 25% testing data. Table III summarizes the highest accuracy obtained for the various classifiers after the experiments on the 5 Events.

The combination of features which provided the best results for all models include – user_follower_ count, user_friends_count, favourite_count, retweet_count, age, average_number_posts and breaking_count

After training the models, the classifiers were tested on a new event, Bollywood actress Sridevi's untimely death in 2017. The accuracy obtained improved to 78%.

TABLE III. RESULTS (ACCURACY)

| Classifier | User Features | Content Features | Hybrid |
|---|---|---|---|
| Naïve Bayes | 67.7% | 54.45% | 57.2% |
| SVM | 72.5% | 71.32% | 67.5% |
| Random Forest | 70.31% | 72.8% | 74.6% |

## VI. CONCLUSION

Automatic detection of Gossips in microblogging sites, can assist to contain and even eliminate the spread of Gossips. However, the challenge is to extract relevant features from a large number of possible features.

In addition to 'User Based' features, we have extracted meaningful 'Content Based' features from the data, using Natural Language Processing techniques. We have come up with a mechanism to rank the different features and then do the feature selection based on the ranking. We have compared 3 different machine learning models, each model using only user, only content and also a hybrid of these features. The Random Forest showed the best results on PHEME dataset and accuracy improved when both user and content -based features are used. We also tested this model on a new event,

(recent untimely death of Sridevi in 2017) and accuracy was found to be 78%. These results compare favorably with other techniques used in a number of research projects. However, further improvements are needed both in accuracy and in processing before a fully automated Gossip detection solution can be integrated into a microblogging site.

Since there do not currently exist natural language processing libraries for Indian Languages, developing these and then using the same to derive content-based features such as sentiment analysis may help us classify tweets in Indian languages with higher accuracy.

Also, finding out the source of a Gossip by building a social network graph can be considered as a worthy addition to the existing Gossip detection system.

## VII. APPENDIX

Future work can be continued in the method of Using Different Algorithm for Gossip Detection in twitter making method more and more exact ad also more dependable.

## VIII. ACKNOWLEDGMENT

We are thankful to all portion hands in achievement of this project. We would like to communicate our honest recognition to all those who have provided us with valuable leadership towards achievement of thesis.

## REFERENCES

[1] Dr.S.Vijayarani, Ms.J.Ilamathi, Ms.Nithya,Preprocessing techniques for text mining-An overview, International Journal of Computer Sci-ence and CommunicationNetworks,Vol5(1),Pages7-16.

[2] Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J.Jansen, Kam-Fai Wong, Meeyoung Cha, ”Detecting Rumors from Microblogs with Recurrent Neural Networks” in IJCAI’16 Proceedings, Twenty-Fifth International Joint Conference on Artificial Intelligence, Pages 3818-3824.

[3] Zhe Zhao, Paul Resnick, Qiaozhu Mei,”Enquiring Minds: Early De-tection of Rumors in Social Media from Enquiry Posts” in Proceeding WWW’15 Proceedings of the 24th International Conference on World Wide Web, Pages1395-1405.

[4] Samantha Finn, Panagiotis Takis Metaxas, Eni Mustafaraj, ”Investi-gating Rumor Propagation with TwitterTrails”, partially supported by NSF grant CNS-1117693 and by the Wellesley Science Trustees Fund.

[5] Liang, Gang, et al. "Gossip identification in microblogging systems based on users’ behavior." IEEE Transactions on Computational Social Systems 2.3 (2015): 99-108.

[6] Qazvinian, Vahed, et al. "Gossip has it: Identifying misinformation in microblogs." Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2011.

[7] Zubiaga, Arkaitz, et al. "Detection and resolution of Gossips in social media: A survey." arXiv preprint arXiv:1704.00656(2017).

[8] Zubiaga, Arkaitz, Maria Liakata, and Rob Procter. "Exploiting context for Gossip detection in social media." International Conference on Social Informatics. Springer, Cham, 2017..