A Graph-Based Approach Leveraging Posts and Reactions for Detecting Rumors on Online Social Media

Asimul Haque
Department of Computer Science
South Asian University
New Delhi-21, India
asimulhaque@gmail.com

Muhammad Abulaish, SMIEEE
Department of Computer Science
South Asian University
New Delhi-21, India
abulaish@sau.ac.in

Abstract

In this paper, we present a novel graph-based contextual and semantic learning approach for detecting rumors on online social media. The underlying hypothesis is that social media entities are intertwined, and if an event unfolds then similar narratives or user reactions with common interests get circulated. The proposed approach uses tweets and their reactions to understand the underlying interaction patterns and exploits the textual and latent information. Textual data is modeled as a words co-occurrence graph, which produces two prevalent categories of words – *substantial words* and *bridge words*. These words serve as building blocks for constructing contextual patterns for rumor detection by computing node-level statistical measures. The contextual patterns are further enriched by identifying negative emotions and inquisitive aspects in the reactions. The patterns are finally ranked and only top-k check-worthy patterns are used for feature generation. In order to preserve the semantic relations, we use word-level GloVe embedding trained over a Twitter dataset. The proposed approach is evaluated over a publicly available PHEME dataset, and compared with various baselines and SOTA techniques. The experimental results are promising and the proposed approach seems useful for rumor detection on online social media.

1 Introduction

The increasing popularity of online social media has motivated various actors to use them for spreading misinformation to set their personal agendas. Misinformation is dangerous, and it can stymie our efforts to address global challenges as many issues are being fuelled and distorted by it. Misinformation acts like a virus in the sense that it exploits our weaknesses, biases, prejudice, and emotions. It has deadly consequences, including polarizing debates, creation or deepening of societal tensions, undermining truth, and manipulation of the electoral processes. What we share online can have consequences in the real-world. The *world economic forum* has ranked massive digital misinformation as one of the top global risks[1]. The United Nations urged social media users to *"pause- take care before you share"* on the *world social media day* to combat misinformation. The world health organization termed misinformation as an *"infodemic"* which is spreading faster than COVID-19, disrupting public health efforts and distorting the sound scientific guidance[2].

Manual fact-checking sites like Snopes, PolitiFact, and FactCheck are available with some traditional media fact-checkers, such as (i) Reality Check – a fact-checking arm of the BBC which is reported in an article entitled *Coronavirus: the human cost of fake news in India*[3], and found spam propagating life-threatening consequences based on the false claims related to coronavirus outbreak, Delhi riots, citizenship amendment act, and claims about the minority com-

[1] https://www.weforum.org/reports/the-global-risks-report-2020

[2] https://www.who.int/health-topics/infodemic

[3] https://www.bbc.com/news/world-asia-india-53165436

munity, (ii) Verified, which is an United Nation's initiative providing facts and life-saving information to the citizens of the world, (iii) Google News Initiative, which provides funding to fight misinformation during the COVID-19 pandemic, (iv) NewsGuard, which is an internet tool for tracking misinformation. It investigated and flagged several Facebook pages as rumor spreaders. Apart from these, social media firms are also concerned with the identification of misinformation on their sites and consequently employing experts and encouraging users to flag or report posts that are not credible.

In the recent era of social media, a type of misinformation that spreads across the network in a short time span is known as a *rumor*. Rumor consists of controversial news, including factually incorrect information about celebrities, politics, crises, and social affairs. Information keeps circulating across social media platforms with unchecked veracity. In due course, the veracity of any information may be determined as true or false, or the same may remain unchecked (Zubiaga et al., 2018). Any kind of rumor or false news disseminates faster than authentic and true news. However, in the case of rumors on political matters, the rapidity of spread and the negative consequences of rumors or fake news is exacerbated (Vosoughi et al., 2018).

The ruckus caused by the spread of misinformation with nefarious motives distorts the truth and tarnishes the spirit of social media platforms. Therefore it is essential to curtail the spread of false rumors and elevate trust in the whole ecosystem of social media. Accordingly, it is becoming a challenge for social media scientists and researchers to devise effective algorithms for determining the veracity of any information on the breeding ground of social media. However, the methods devised for identifying and determining the veracity of rumors on social media, especially on Twitter, are based on hand-crafted features that come from two main aspects – content and users' social context (Zubiaga et al., 2017). Researchers are generally using traditional machine learning approaches for learned-representation of deep learning techniques to classify rumorous messages. There are recent works on determining the veracity of rumors, but very few of them focus on the detection of rumors based on the contextual representation learning using a graph-based approach. When an event begins to unfold, a similar type of user responses and a comparable set of patterns are generated. The re-circulation of the original post and showing skeptical or negative responses create echo chambers, convincing us to incorporate reactions and retweets while addressing the truthfulness of a tweet.

In this paper, we present a graph-based contextual learning representation for detecting rumors in Twitter. The graph-based approach is exploited to capture the contextual information from the tweets. Word co-occurrence graphs are constructed using textual data, and node metrics like *eigenvalue centrality* and *clustering coefficient* are computed. The *clustering coefficient* sets out the topical words that are generally clustered together, while *eigenvalue centrality* lists the words that connect the topical words. These two families of words are collaborated to create patterns. The novelty of our approach is in incorporating reactions with the source tweets to capture their inherent semantic affinity. Our approach is also unique in its way of graph-based representation learning and identifying two prevalent categories of words to extract the rumorous patterns. Moreover, the emotional and inquisitive words are considered to make patterns more generic. The ranking of patterns is performed through *tf-idf* weight score and top-k check-worthy patterns are extracted. For preserving the semantic relations, word-level GloVe embeddings learned from a Twitter dataset is applied. The tweets are split into n-grams, and *Cosine* similarity is used to generate the feature vectors by calculating the similarity between the patterns and tweets. Different classification models are trained over the original and a balanced distribution of the publicly available PHEME dataset. In comparison to the word-based method, the pattern-based approach has the advantage of being more representative and retaining the syntactic sense.

2 Related Works

In recent years, rumor and fake news detection are becoming one of the most explored areas of research. Although the rumor is an old phrase, the implications of such mis/disinformation on online social networks become apparent when events unfold as breaking news, prompting individuals to rely on social media for news and information. It can be said that the term

fake news is also old, but got popularized after the 2016 presidential election in the USA. The very first work tackling the detection of emerging rumors was proposed in (Zhao et al., 2015), based on extracting correction and verification signals using regular expressions. Alternately, the authors in Zubiaga et al. (2017) leveraged the context using CRF with assumptions that context plays a crucial role in determining the rumors.

Gradually, research on the subject of rumor identification progressed, with mechanisms ranging from evaluating the rumorous nature of tweets to detecting stances and establishing the truth value of social media posts (Mohammad et al., 2016; Kochkina et al., 2017; Rosenfeld et al., 2020). The transformation of approaches varies from traditional models like SVM and Random Forest to probabilistic graphical models like Bayesian classifier and deep neural networks like LSTM and GCN (Castillo et al., 2011; Bai et al., 2021). Some researchers have explored behavioral, emotional, and sentimental aspects for the detection of rumors and fake news (Ajao et al., 2019). Abulaish et al. (2019) proposed a graph-based approach for rumor detection using `POS` tags to identify anxious and doubtful terms present in microblogging posts.

In recent years, various deep learning approaches are proposed for rumor detection. Some recent approaches are based on Graph Convolutional Neural Net (GCN) Bai et al. (2021) using a relationship between source and reply, whereas Dong et al. (2019) identified multiple rumor sources without knowing the underlying propagation structure. In Tu et al. (2021), the authors proposed a CNN-based model using text and propagation structure. Ma et al. (2019) applied a Generative Adversarial Network (GAN) based model to detect rumor-inductive patterns even for a low-frequent rumor. Nguyen et al. (2020) discussed a graph learning framework for fake news detection using the social structural engagements of the users. Compared with traditional textual content and social user features, the images and video-based rumors have been researched less. Deep learning has encouraged researchers to explore multimodal features of rumors (Zhou et al., 2020). Related research has looked into many elements of detecting rumors and fake news on social media, but they are less focused on inferring underlying information and extracting patterns from tweets and reactions. Moreover, of all the strategies produced, only a few are centered on graph-based representation learning mechanisms.

3 Proposed Approach

Following the existing state-of-the-art works, we consider rumor detection as a binary classification problem. Given a tweet $t_i \in T$ of a `Twitter` dataset T, rumor detection problem aims to estimate a function r which predicts the class level of t_i as a rumor or non-rumor. Mathematically, it can be defined as r: $t_i \rightarrow \{0, 1\}$ such that,

$$r(t_i) = \begin{cases} 1 & \text{if } t_i \text{ is a rumor} \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

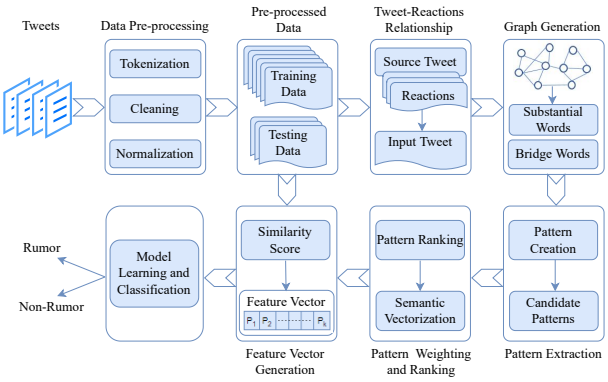

Figure 1: Work-flow of the proposed approach for rumor detection

The work-flow of the proposed approach for rumor detection is presented in figure 1. Different functionalities of the proposed approach are explained in the following sub-sections.

3.1 Data Pre-processing

In this step, several data pre-processing tasks such as tokenization, cleaning and normalization are performed on the `Twitter` dataset. All the tweets and reactions are tokenized with white space. To normalize the dataset and avoid any bias towards any events or twitter-name, we replace the twitter-specific tokens *hashtags, urls, retweet* and *mentions* with the tag $\prec$*hashtags*$\succ$, $\prec$*urls*$\succ$, $\prec$*retweet*$\succ$ and $\prec$*mentions*$\succ$, respectively. The punctuation and emoticons such as ?, !, and :(are not removed because they play a

significant role in examining the writing style of rumors. Stop-words and capital letters are significantly important to understand the context of tweets. The capital letters show emphasizing and stop-words intact syntactic sense. Therefore, stop-words are also retained to aid in forming rumor patterns.

3.2 Inferencing Tweet-Reactions Relationships

This section aims to identify closeness between the source tweets and reactions. The reactions describe the underlying latent information. Users show *skepticism*, *correction*, and *verification* in their reactions towards the truthfulness of a tweet. Some users give opinions or clues regarding the factuality of a tweet. Therefore, tweets and reactions are highly correlated. The reactions provide supplements to the source tweet that enhance and describe their quality and are also helpful for identifying the underlying patterns in rumorous tweets.

For making tweets explainable towards rumors, the reactions are incorporated with them to get additional knowledge. When combining of tweets and reactions, the repetition of the same tweets in reactions should be avoided to make them unique and descriptive. For this purpose, the symmetric difference approach is used at the sentence level on the tweets and their reactions. For any tweet t_i and its set of reactions $R_{t_i} = \{r_1^{t_i}, r_2^{t_i}, ..., r_n^{t_i}\}$, the repetitions are removed using equation 2 by calculating the symmetric difference of a tweet and all its reactions.

$$SD_i = t_i \Delta \{r_n^{t_i}\}_1^n \tag{2}$$

The symmetric differences between a tweet and reactions can be formally defined as:

$$t_i \oplus \{r_n^{t_i}\}_1^n \tag{3}$$

To maintain the inclusiveness of a tweet and its reactions, a single input tweet is created by combining a tweet and its deduplicated reactions by making a union of all the symmetric differences SD_i. We define this relation in equation 4, where $X = (t_i, R_{t_i})$ is the set of a tweet and the union of symmetric differences SD_i of a tweet $t_i \in T$. This equation 4 represents an input tweet X_i in which the first sentence is a tweet followed by all its deduplicated reactions.

$$X_i = t_i + \bigcup_1^n SD_i^{t_i} \tag{4}$$

3.3 Graph Generation

With minimum domain knowledge, the graph-based method can effectively capture linguistic variation and contextual information in textual data. The input tweets are represented as a word co-occurrence graph, $G(V, E)$, where V is a finite set of nodes representing words and E is the set of edges representing the relationship between the nodes. To preserve the underlying structure of the input tweets, edges are defined as a sequence of words in an input tweet with a window size of 2. The graph is used to model two prevalent categories of the words – substantial words and bridge words that are identified by computing node-level statistical measures. The key objective is to collect the words that are relevant to constructing rumorous patterns. Two operations are performed on the graph for computing node statistics; *clustering coefficient* and *eigenvector centrality*. *Clustering coefficient* is used to identify substantial and topical words, whereas *eigenvector centrality* is used to identify bridge words that provide connections to the substantial words.

3.3.1 Substantial Words

When any event starts unfolding, a similar type of tweets start posted on social media that contains a similar set of responsive or reactive words. These words can be psychological, sentimental, or skeptical that are normally clustered together to impose sentiments, feelings, or suspicions about the event. These types of words are also connected with each other through the same words that we recognize as bridge words. Therefore, in order to trace the clustering behavior of such topical words, the *clustering coefficient* of each node $v_i \in V$ is calculated using equation 5.

$$C_{c(v_i)} = \frac{2|e_{jk} : v_j, v_k \in N_i, e_{jk} \in E|}{k_i(k_i - 1)} \tag{5}$$

The average clustering coefficient of a node v_i is defined as $C_{c(v_i)}/|V|$ and the average clustering score of the graph G is defined as $\sum_{v \in V} C_{c_v}/|V|$. At the end, the list of substantial words is compiled with the words having a clustering coefficient value greater than the threshold θ_{C_c}, which is determined empirically as a value just greater than the average clustering score of the underlying graph.

3.3.2 Bridge Words

As the name suggests, this category of words makes a bridge between two substantial words and contributes to making rumorous patterns. It is used to deliberate the meaning of the substantial words and categorize the informal representation of the words because people use to write in casual ways on social media. To consider the frequent as well as focal nodes in the graph and based on the fact that a node is important if its neighbors are important, we choose the *eigenvector centrality* measure to identify bridge words. The benefit of choosing *eigenvector centrality* is to avoid frequent but irrelevant words in the graph. The *eigenvector centrality* of a node $v_i \in V$ is calculated using equation 6, where N_i is the neighbors of v_i and $A(i, j)$ is the adjacency matrix of the graph G. In other words, the centrality of a node v_i is proportional to the combined centrality values of its neighbors $v_j \in N_i$.

$$C_{e(v_i)} \propto \sum_{v_j \in N_i} A(i,j) C_e(v_j) \tag{6}$$

Now, it can be re-written in a matrix form as given in equation 8, where λ is a proportionality constant.

$$x \propto Ax \tag{7}$$

$$\lambda x = Ax \tag{8}$$

This equation looks exactly like the eigenvector equation. The centrality vector x is the eigenvector of the adjacency matrix $A(i, j)$ associated with the eigenvalue λ. By virtue of the *Perron-Frobenius* theorem, it takes the largest value of λ, and we find the corresponding eigenvector that is positive and unique, giving the *eigenvector centrality* of each node $v_i \in V$. The list of bridge words is compiled with the words having a centrality value greater than the empirically calculated threshold θ_{C_e} value.

3.4 Pattern Extraction

The objective of this step is the extraction of the rich rumorous patterns that are found more frequently in rumors. The order of words in any language is critical for delivering meaningful information. People communicate based on syntactic rules to convey the proper meaning. The same syntactic rule is followed in social media while writing posts. During the extraction of check-worthy patterns, we have considered the syntactic rule as well as the emotion and skeptical nature of social media users. We make the patterns as a combination of above-mentioned two categories of words. Keeping syntactic rules and extracting meaningful patterns; first, the nodes that satisfy the threshold condition are identified from both categories and located in the graph. Thereafter, we find the connections between the words of different categories. The connections are basically the edges between a node v_i and all its neighbor N_i. Since we are interested in considering the patterns that consists of maximum words to get more contextual and syntactic meanings, we have taken 3-word patterns that are created by combining a node with its two neighbors in a way that not all three nodes are from the same category. Finally, all possible n patterns are collected and stored in the list P such that $P = \{p_1, p_2, \ldots, p_n\}$.

3.4.1 Candidate Patterns Selection

The check-worthy patterns identified from list P are named as candidate patterns. The following steps explain the selection of the candidate patterns.

The first step is based on the assumption that the rumor contents are related to a specific event, i.e., when an event unfolds, individuals begin to circulate similar sets of responses and generate comparable types of information without verifying their veracity. Therefore, to incorporate event-specific patterns as well as making patterns close to being rumorous, the same set of procedures discussed above are applied to that portion of training data which is labeled as rumor only. Then this newly obtained list of patterns P_{new} is used to shorter the patterns list P. It is achieved through matching the list P with a newly designed list of patterns P_{new}. The matching is performed as described below:

N-gram matching: The *n*-gram matching works on the word level, providing the collection of patterns that match with rumor patterns. The matching is done at the trigrams level, i.e., the patterns that match all three words are collected in a list, P_R, named as rumor patterns list.

Similarity matching: In the above step, the exactly matched patterns are captured, but some similar patterns are left out that are extracted using a similarity measure. We have used Cosine similarity for this

purpose. The Cosine similarity between a pair of patterns is calculated using equation 9, where p_i is the i^{th} pattern of list P and p_{new_j} is the j^{th} pattern of list P_{new}. The obtained patterns from this step are appended to the rumor patterns list P_R.

$$Cosine(p_i, p_{new_j}) = \frac{p_i \cdot p_{new_j}}{||p_i|| \cdot ||p_{new_j}||} \quad (9)$$

The first step gives the rumor patterns that are event-specific and about the content of the in-hand dataset. To make patterns generic, we consider the responsive behavior of social media users since the writing style and user behavior are mostly the same for any sort of unverified or rumorous social media posts. Incorporating reactions with tweets are advantageous to examining the emotional, correcting, and verifying contents in the input tweets.

The second step incorporates the sentimental and emotional words in the patterns list P. As discussed in Vosoughi et al. (2018), the reactions to rumors contain fear, surprise, and disgust. People re-circulate the rumorous tweet with negative sentiments. To consider the negative emotional words, `NRC Word-Emotion Association Lexicon` is used, consisting of words along with associative emotions *(anger, fear, anticipation, trust, surprise, sadness, joy*, and *disgust)* and *(negative* and *positive)* sentiments (Mohammad and Turney, 2013). The patterns having negative emotional words are extracted from the pattern list P and produced as a part of a list of emotional patterns P_E.

The third step considers the skeptical nature of social media users. When a tweet gets posted, people show skepticism and start *questioning* and *verifying* them in the form of support and denial. The correction and verification inquiry in the replies are also observed in (Zhao et al., 2015). Inspired by their work, the regular expressions such as $(true|not|true)$, $(real?|really?)$, $(rumor|debunk)$, and $(false|fake)$ are constructed and passed through the pattern list P. The patterns containing the skeptical words are extracted and considered as a part of skeptical patterns list P_S.

The above three steps extract the generic and specific patterns from the list P. Thus, candidate patterns have consolidated three categories of pattern lists – rumor patterns, emotional patterns, and skeptical patterns.

3.5 Pattern Weighting and Ranking

In this step, the top-k check-worthy patterns are selected from each list of candidate patterns. Since we are interested in finding highly rumorous patterns, a weight value is assigned to each pattern to define its ranking. It is achieved through *tf-idf* weight scoring, where each pattern of all three categories of the candidate list is assigned a weight score through the rumor training corpus tc. Furthermore, semantic vectorization is performed using embeddings to preserve semantic richness.

We modified the standard formula of *tf-idf* for calculating the weight score to map the patterns close to being rumorous. For this purpose, we split the input tweets of the rumor training corpus into n-grams where n is equal to the number of words in a pattern p_i. The modified *tf-idf* is applied separately on rumor patterns list P_R, emotional patterns list P_E, and skeptical patterns list P_S to assign weight scores to every pattern of the respective categories. For any of the above lists, each pattern of that list is considered as a term, and its frequency value is calculated, named as pattern frequency PF. We pass a pattern through the n-grams rumor corpus, and word co-occurrence is counted. The pattern frequency of a pattern PF_{p_i} is defined in equation 10, where, tc is the rumor training corpus, $f(p_i, tc)$ is the frequency of a pattern p_i in tc, and $f(p, tc)$ is the total number of patterns present in tc.

$$PF_(p_i, tc) = log\left(1 + \frac{f(p_i, tc)}{f(p, tc)}\right) \quad (10)$$

The inverse document frequency IDF for a pattern p_i is computed to provide the relevance of pattern p_i with the training corpus using equation 11, where $n(X)$ is the total number of input tweets in the rumor training corpus tc and $f(x, p_i)$ represents the total number of input tweets in which pattern p_i occurs.

$$IDF_{p_i} = log\left(1 + \frac{n(X) \in tc}{f(x, p_i)}\right) \quad (11)$$

Thus, the pattern frequency-inverse document frequency $PF - IDF$ is calculated by scalar multiplication of equations 10 & 11, as shown in equation 12,

and it is considered as the final weight score for a pattern p_i.

$$PF - IDF_(p_i, tc) = PF_{(p_i,tc)} \times IDF_{p_i} \quad (12)$$

We arrange the patterns in each category of the list in descending order of their weight score for selecting top-k check-worthy patterns. Finally, the top-k patterns from all three lists are combined to make a hybrid patterns list that is useful in detecting the rumor.

3.5.1 Semantic Vectorization

We have applied embedding on the input tweets to preserve the semantic similarity with the patterns. For each input tweet X_i of the training corpus, we split it into n-grams and generate a semantic vector of each n-gram using word embedding. For example, if a tweet has m number of n-grams, then the semantic vector of the tweet is represented as a matrix of $\big[[|n| \times [d] \quad |n| \times [d] \quad \dots \quad |n| \times [d]\big]_{1\times m}$, where $|n|$ is number of words in n-grams and d is the dimension of the word embedding vector.

The vectorization step is also applied over the identified patterns. It enriches patterns and provides semantic information by using word embedding. For preserving the semantic relations, we make a semantic vector for a pattern using the same word embedding. The semantic vector of a pattern is a matrix of order $|v_i| \times [d]$, where $|v_i|$ represents the number of words in a pattern p_i.

3.6 Feature Vector Generation

In this step, tweets are converted into a feature vector using Cosine similarity. The semantic vector of both the patterns and n-grams of the input tweets are matched, and the mean similarity score is calculated. For example, a list consists of top-k patterns; for an i^{th} pattern p_i, the similarity score is obtained by matching it with each n-gram. Furthermore, we calculate the mean of the similarity score. This step repeats until all the patterns from the list are matched with n-grams of the input tweets. Since there are total k patterns, the size of the final feature vector is k. For a pattern p_i, the numeric value for the feature vector is denoted by χ_{p_i} and calculated using equation 13, where m is the total number of n-grams in input tweet X_i and $Cosine(p_i, X_{i_m})$ is a similarity score of a pattern p_i and a n-gram of input tweet X_i.

$$\chi_{p_i} = \frac{\sum_1^m Cosine(p_i, X_{i_m})}{m} \quad (13)$$

4 Experimental Setup and Results

4.1 Dataset

We conduct our experiment on a publicly available PHEME dataset described in (Zubiaga et al., 2017). The dataset contains a collection of tweets with their reactions (direct or nested) and metadata of five breaking news events. There is a total of 5802 source tweets in which 1972 are labeled as a rumor and 3830 as a non-rumor. This dataset contains less number of rumors in comparison to non-rumors. Therefore, another variant of a balanced dataset is created from the original dataset, where both rumor and non-rumor have an equal number of 1972 instances. We conduct our experiment on both variants of the datasets. The detailed statistics of the dataset is presented in Table 1.

Table 1: Statistics of the dataset

Events Name	Source Tweets	Reactions	Rumors	Non-Rumors	Total
Charlie Hebdo	2079	36189	458	1621	38268
Sydney Siege	1221	22775	522	699	23996
Ferguson	1143	23032	284	859	24175
Ottawa Shooting	890	11394	470	420	12284
Germanwings Crash	469	4020	238	231	4489
Total	5802	97410	1972	3830	103212

4.2 Evaluation Metrics

This section discusses the formal description of performance evaluation metrics for classification. The three standard data mining evaluation metrics- *Precision, Recall*, and *F1-Score* are defined in the equation 14, 15, and 16, respectively. For defining the evaluation metrics, we have used the concept of *True-Positive (TP)*, i.e., the number of rumorous tweets identified correctly, *False-Positive (FP)*, i.e., the number of non-rumor tweets identified as rumorous, and *False-Negative (FN)*, i.e., the number of rumorous tweets identified as non-rumor. *Precision* evaluates the correctness of the classifier, and *Recall* evaluates the completeness of coverage of the classifier, while *F1-Score* provides the way to combine the contribution of both *precision* and *recall* evenly by using

harmonic mean.

$$Precision = \frac{TP}{TP + FP} \tag{14}$$

$$Recall = \frac{TP}{TP + FN} \tag{15}$$

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{16}$$

4.3 Evaluation Results and Comparative Analysis

The experimental evaluation is performed using four machine learning classification algorithms *support vector machine (SVM), gradient boosting (GB), conditional random field (CRF)*, and *logistic regression (LR)* over both the variants of benchmark dataset. The proportion of train and test sets in the proposed approach is $8:2$. We have implemented the above classification models using `scikit-learn python` library.

Following pre-processing, the unigram frequency matrix of graph words is generated. Those unigrams with a frequency of less than ten are discarded. After that, we applied *clustering coefficient* and *eigenvector centrality* on the graph to extract the two prevalent categories of words. The graph generation and its operations are implemented using the `networkx` package for the `Python` programming language. We have considered different threshold values to extract top words from both categories. Words of these two categories are located in the graph and their immediate connections i.e., $e_{ij} \in E$ of a node v_i with neighbors N_i, where $v_j \in N_i$, are retrieved. We used the sliding window of size 2; therefore, patterns of three words are created by combining a node with its neighbor in a way that not all three words are from the same category.

To make patterns generic, we incorporated negative emotional lexicons and skeptical words. As a result, we curated three lists of patterns named rumor patterns, emotional patterns, and skeptical patterns. The word clouds in figure 2 present words of the top-100 patterns for each category in which the font size of a word is directly proportional to its relevance score. It can be observed from these word clouds that the words with larger font sizes are significantly related and explain the context of the underlying categories. We ranked the patterns through *tf-idf* and selected the top-k patterns empirically. We have used 100-dimensional word-level pre-trained `GloVe` embedding trained over the `Twitter` dataset with 27 billion tokens. Those words that do not exist in the pre-trained word vectors are ignored. We split each input tweet into trigrams since each pattern has three words. While splitting the input tweets, 2 rumors and 15 non-rumor are ignored since they have less than three words, and trigrams can not be created. The patterns and trigrams of tweets are represented in the 3×100 dimensions; we have taken their centroid value and reduced it to the 1×100 dimensions. At last, the mean similarity value of a pattern with each trigram of tweets is calculated. The top-k patterns produce a k-dimensional feature vector.

The propose approach is compared with the following baseline methods and state-of-the-art approaches.

Baseline 1: In this method, only the top-k rumor patterns are incorporated to generate the feature vector.

Baseline 2: In this method, the top-k patterns from rumor as well as skeptical patterns are incorporated to generate the feature vector.

Baseline 3: In this method, the top-k patterns from rumor as well as emotional patterns are incorporated to generate the feature vector.

Ajao et al. (2019): This approach considered the relationship of rumors with the sentiments of the social media post. It has used emotional words to detect the sentiment-aware misinformation.

Abulaish et al. (2019): This approach used the graph-based approach for rumor detection that incorporated the sentimental aspects, such as anxiety and doubtful terms from the social media post.

Zubiaga et al. (2017): This approach has learned through the sequential dynamics of the social media post. The content features and user social features have experimented with *conditional random fields (CRF)*.

We have compared the proposed approach with three baselines and three state-of-the-art approaches.

(a) Rumor patterns (b) Emotional patterns (c) Skeptical patterns

Figure 2: A word-cloud representing words of the top-100 patterns for each category

Table 2: Comparative performance evaluation results of our proposed approach with state-of-the-art approaches and baseline methods over the original dataset

Approach	GB			SVM			CRF			LR		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Baseline1	82.48	72.44	77.13	82.07	62.61	71.03	79.81	65.68	72.04	76.09	52.35	62.03
Baseline2	79.38	76.35	77.83	81.20	63.32	71.15	82.32	72.34	77.00	80.00	53.71	64.27
Baseline3	85.80	87.82	86.80	87.53	91.45	89.45	86.34	82.56	84.40	83.84	76.50	80.00
Ajao et al. (2019)	84.80	85.12	84.96	86.68	85.82	86.24	86.43	84.64	85.52	83.83	85.11	84.46
Abulaish et al. (2019)	56.26	55.40	55.82	41.30	45.56	43.32	64.62	60.10	62.28	40.80	41.93	41.11
Zubiaga et al. (2017)	52.43	54.19	53.29	36.60	44.78	40.27	69.19	54.59	61.02	33.10	40.72	36.51
Proposed Approach	92.83	94.02	93.42	93.48	91.88	92.67	91.06	88.97	90.00	90.97	86.11	88.47

Table 2 summarizes the comparative results in terms of *precision*, *recall*, and *f1-score* over the original dataset. It can be observed that the proposed approach outperforms all other approaches for all four classification algorithms. Our best result is obtained through the *gradient boosting*. It can also be observed that *gradient boosting* achieved highest *recall* and *f1-score*, whereas *SVM* achieved highest *precision*. To assess the effect of class imbalance, we repeated the same set of experiments with the variant of a balanced dataset. As shown in table 3, *gradient boosting* scored the highest value of *precision* and *f1-score*, whereas *CRF* achieved the highest *recall* value. It can also be observed that the *recall* and *f1-score* values are better over the balanced dataset for all four classification algorithms.

Figures 3 and 4 present a visualization of the comparative analysis results of the proposed approach with three state-of-the-art techniques over both original and balanced datasets, respectively. It can be observed that the proposed approach performs significantly better than all state-of-the-art approaches. The improvements of the proposed approach over the best state-of-the-art approach range from $4.01 - 8.46\%$ for the original distribution of the dataset and $3.93 - 6.33\%$ for the balanced dataset. The out-performance consistency of our proposed approach is maintained for both variants of the dataset with a difference of $1.03 - 3.85\%$ in the *f1-score* value. In contrast, the state-of-the-art approaches are inconsistent; some methods improved 12.07% in the *f1-score* value, whereas the performance of a few decreases. The reason for the consistent performance of the proposed approach is that the graph-based approach covers the topical words and the writing style of the social media post, whereas the word embed-

Table 3: Comparative performance evaluation results of our proposed approach with state-of-the-art approaches and baseline methods over the balanced dataset

Approach	GB			SVM			CRF			LR		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Baseline1	82.49	79.54	80.99	79.62	79.96	79.79	81.21	82.38	81.79	76.82	75.53	76.17
Baseline2	82.77	85.86	84.29	76.35	85.24	80.55	80.54	87.24	83.75	76.57	78.79	77.66
Baseline3	91.10	92.83	91.95	89.90	95.10	92.42	88.13	95.04	91.45	85.80	94.30	89.85
Ajao et al. (2019)	86.88	89.21	88.03	88.00	89.56	88.77	84.50	83.86	84.18	85.11	82.37	83.71
Abulaish et al. (2019)	61.22	59.46	60.32	52.41	50.86	51.62	82.04	67.98	74.35	63.56	60.02	61.74
Zubiaga et al. (2017)	58.63	60.40	59.50	51.13	50.75	50.94	80.00	62.53	70.19	52.43	45.66	48.81
Proposed Approach	95.36	95.36	95.36	91.72	95.78	93.70	91.18	95.88	93.47	90.82	93.88	92.32

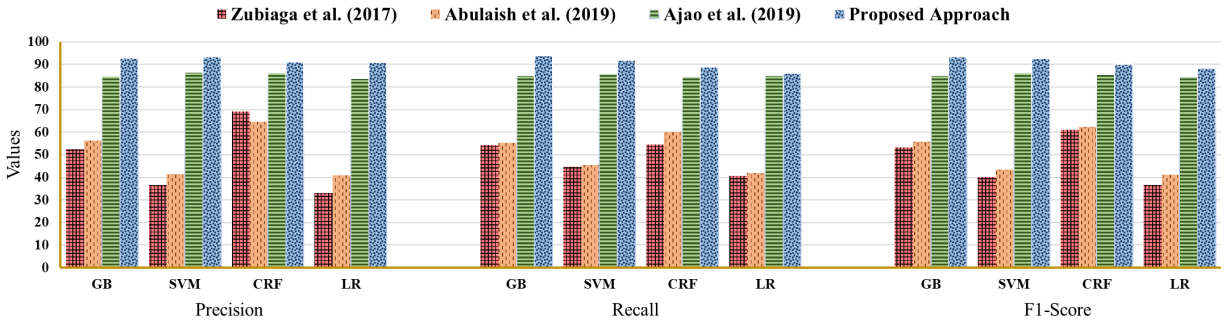

Figure 3: Comparative performance evaluation analysis over the original dataset

ding covers the higher dimensional semantic space.

5 Conclusion and Future Work

In this paper, we have presented a rumor detection framework that uses a graph-based approach to leverage tweets and reactions for extracting rumorous patterns. A graph-based learning representation is used to capture contextual information. The inquisitive, skeptical, sentimental, and emotional natures of social media users are identified through their writing styles in the reactions to a tweet. The semantic relations are preserved through semantic vectorization, based on word embedding. The hybrid top-k patterns are extracted from all three categories – rumor, emotional, and skeptical that are used to train the rumor detection model. The experimental results explained that our proposed approach significantly improves the rumor detection task and outperforms the state-of-the-art methods. As a result, it is concluded that utilizing emotional and skeptical words makes the detection system more effective. It can also be said that the pattern-based approaches result in more representative and smaller size models than the word-based approaches. The patterns also retain the syntactic sense. In this study, we have considered three-words patterns because long patterns are infrequent. The proposed work can be extended to improve the pattern ranking mechanism to maximize the coverage of patterns.

References

Muhammad Abulaish, Nikita Kumari, Mohd Fazil, and Basanta Singh. 2019. A graph-theoretic embedding-based approach for rumor detection in twitter. In *Proceedings of the 18th IEEE/WIC/ACM International Conference on Web Intelligence, Thessaloniki, Greece*. Association for Computing Machinery, 466–470.

Oluwaseun Ajao, Deepayan Bhowmik, and Shahrzad Zargari. 2019. Sentiment aware fake news detection on online social networks. In *Proceedings of the 44th IEEE International Conference on Acous-*

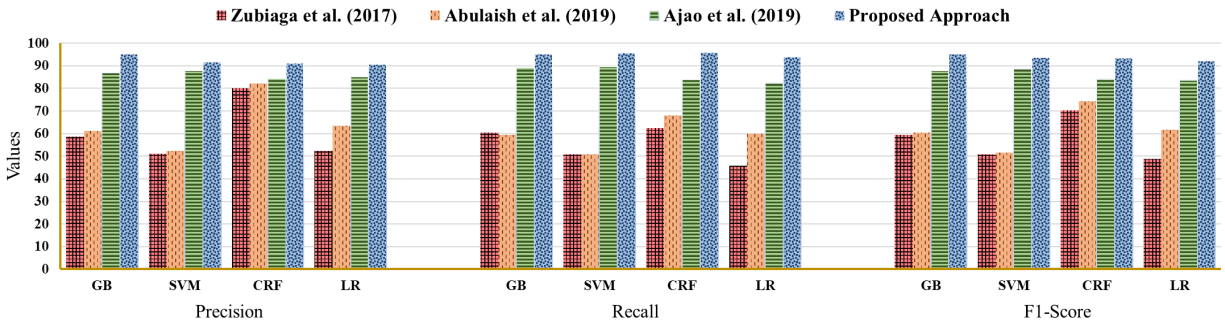

Figure 4: Comparative performance evaluation analysis over the balanced dataset

tics, Speech and Signal Processing (ICASSP), Brighton, UK. IEEE, 2507–2511.

Na Bai, Fanrong Meng, Xiaobin Rui, and Zhixiao Wang. 2021. Rumour Detection Based on Graph Convolutional Neural Net. *IEEE Access* 9 (2021), 21686–21693.

Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. 2011. Information credibility on twitter. In *Proceedings of the 20th International Conference on World Wide Web, Hyderabad, India*. Association for Computing Machinery, 675–684.

Ming Dong, Bolong Zheng, Nguyen Quoc Viet Hung, Han Su, and Guohui Li. 2019. Multiple rumor source detection with graph convolutional networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China*. Association for Computing Machinery, 569–578.

Elena Kochkina, Maria Liakata, and Isabelle Augenstein. 2017. Turing at SemEval-2017 Task 8: Sequential Approach to Rumour Stance Classification with Branch-LSTM. In *Proceedings of the 11th International Workshop on Semantic Evaluation, Vancouver, Canada*. Association for Computational Linguistics, 475–480.

Jing Ma, Wei Gao, and Kam-Fai Wong. 2019. Detect rumors on twitter by promoting information campaigns with generative adversarial learning. In *Proceedings of 28th The World Wide Web Conference, San Francisco, CA, USA*. Association for Computing Machinery, 3049–3055.

Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016. Semeval-2016 Task 6: Detecting stance in tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation, San Diego, California, USA*. Association for Computational Linguistics, 31–41.

Saif M. Mohammad and Peter D. Turney. 2013. Crowdsourcing a Word-Emotion Association Lexicon. *Computational Intelligence* 29, 3 (2013), 436–465.

Van-Hoang Nguyen, Kazunari Sugiyama, Preslav Nakov, and Min-Yen Kan. 2020. Fang: Leveraging social context for fake news detection using graph representation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Virtual Event, Ireland*. Association for Computing Machinery, 1165–1174.

Nir Rosenfeld, Aron Szanto, and David C Parkes. 2020. A kernel of truth: Determining rumor veracity on twitter by diffusion pattern alone. In *Proceedings of 29th The Web Conference, Taipei Taiwan*. Association for Computing Machinery, 1018–1028.

Kefei Tu, Chen Chen, Chunyan Hou, Jing Yuan, Jundong Li, and Xiaojie Yuan. 2021. Rumor2vec: a rumor detection framework with joint text and propagation structure representation learning. *Information Sciences* 560 (2021), 137–151.

Soroush Vosoughi, Deb Roy, and Sinan Aral. 2018. The spread of true and false news online. *Science* 359, 6380 (2018), 1146–1151.

Zhe Zhao, Paul Resnick, and Qiaozhu Mei. 2015. Enquiring minds: Early detection of rumors in social media from enquiry posts. In *Proceedings of*

the 24th International Conference on World Wide Web, Florence Italy. Association for Computing Machinery, 1395–1405.

Xinyi Zhou, Jindi Wu, and Reza Zafarani. 2020. SAFE: Similarity-Aware Multi-modal Fake News Detection. *Advances in Knowledge Discovery and Data Mining, Springer* 12085 (2020), 354.

Arkaitz Zubiaga, Ahmet Aker, Kalina Bontcheva, Maria Liakata, and Rob Procter. 2018. Detection and resolution of rumours in social media: A survey. *ACM Computing Surveys (CSUR)* 51, 2 (2018), 1–36.

Arkaitz Zubiaga, Maria Liakata, and Rob Procter. 2017. Exploiting context for rumour detection in social media. In *Proceedings of International Conference on Social Informatics, Oxford, UK*. Springer, 109–123.