Detecting Breaking News Rumors of Emerging Topics in Social Media

Sarah A. Alkhodair

Concordia Institute for Information Systems Engineering, Concordia University, Montreal, Canada, H3G 1M8

Steven H. H. Ding

School of Information Studies, McGill University, Montreal, Canada, H3A 1X1

Benjamin C. M. Fung

School of Information Studies, McGill University, Montreal, Canada, H3A 1X1

Junqiang Liu

School of Information and Electronic Engineering, Zhejiang Gongshang University, Hangzhou, China

Abstract

Users of social media websites tend to rapidly spread breaking news and trending stories without considering their truthfulness. This facilitates the spread of rumors through social networks. A rumor is a story or statement for which truthfulness has not been verified. Efficiently detecting and acting upon rumors throughout social networks is of high importance to minimizing their harmful effect. However, detecting them is not a trivial task. They belong to unseen topics or events that are not covered in the training dataset. In this paper, we study the problem of detecting breaking news rumors, instead of long-lasting rumors, that spread in social media. We propose a new approach that jointly learns word embeddings and trains a recurrent neural network with two different objectives to automatically identify rumors. The proposed strategy is simple but effective to mitigate the topic shift issues. Emerging rumors do not have to be false at the time of the detection. They can be deemed later to be true or false. However, most previous studies on rumor detection focus on long-standing rumors and assume that rumors are always false. In contrast, our experiment simulates a cross-topic emerging rumor detection scenario with a real-life rumor dataset. Experimental results suggest that our proposed model outperforms state-of-the-art methods in terms of precision, recall, and F1.

Keywords: Breaking news, machine learning, micro-blogs, recurrent neural networks, rumor detection, social media.

Email addresses: sa_alkho@ciise.concordia.ca (Sarah A. Alkhodair), steven.h.ding@mail.mcgill.ca (Steven H. H. Ding), ben.fung@mcgill.ca (Benjamin C. M. Fung), jjliu@alumni.sfu.ca (Junqiang Liu)

1. Introduction

Twitter has been considered one of the most widely adopted social media platforms for spreading breaking news worldwide. In fact, a recent survey from Pew Research Center stated that “As of August 2017, two-thirds (67%) of Americans get news from social media” and that “about three-quarters (74%) of Twitter users have reported getting news on the site” [22]. The importance of social media, especially Twitter, as a major source of up-to-date information arises from the fact that anyone can instantly post, share, and gather information related to breaking news. This flexibility of sharing and exchanging information comes with a drawback of overwhelming readers with a huge volume of new information every second. Unfortunately, the information is not always trustworthy. This nature of social media provides a fertile ground for rumormongers to post and spread rumors that may result in major chaos and unpredictable reactions from involved individuals.

A real-life example is the single tweet reporting an “Explosion at White House” in 2013. Although this rumor was debunked very fast, tweets about it spread to millions of users causing an intense impact and a dramatic plunge in the stock market within only six minutes[1]. Such a major impact could have been avoided, or at least minimized, if there were a way to flag that single tweet as a rumor. This example as well as many other real-life examples show how the explosive spread of rumors in social media can lead to extremely damaging impacts on people and society.

Different definitions of rumors have been used in the literature. However, one of the most adopted definitions is in [2] where a rumor is defined as “a story or a statement whose truth value is unverified”. Definitions of rumors in major dictionaries also coincide with that. According to these definitions, rumors do not have to be false; they can be deemed later to be true or false. The main characteristic of a rumor is that its truth value is unverified at the time of posting. In relevant studies, there are two types of rumors on social media based on the temporal characteristic: long-standing rumors and breaking news rumors [26]. Long-standing rumors are well-discussed for long periods of time, and one can easily collect related training data under the given topics. In contrast, breaking news rumors generally have not been observed before and require zero-shot learning for real-time detection.

Breaking news refers to “newly received information about an event that is currently occurring or developing”[2]. Most regular news evolves slowly, and more details are expected to be revealed over time. In contrast, breaking news is often unexpected events that evolve dramatically fast without many details on what happened or what will happen next. It covers an unexpected sequence of sub-topics that mostly do not occur in existing data. A typical example was the earthquake of magnitude 9.0 that happened in

[1] Source: `http://www.cnbc.com/id/100646197`, retrieved on April 2, 2018

[2] Source: `https://en.oxforddictionaries.com/definition/breaking\_news`, retrieved on April 3, 2018

2011, followed by a tsunami and the failure of three nuclear reactors in Fukushima. This severe consequence is outside of most people's expectations. The nature of breaking news associates it with a lot of rumors on social media. In fact, the volume of rumors is directly proportional to the importance of and interest in the topic to individuals [2]. Therefore, sensitive topics and breaking news tend to be associated with a huge volume of rumors. This is especially true in the early stages of diffusion when the topic is hot, unclear, and attracting a lot of attention.

Real-life incidents of damage and chaos, caused by the spread of rumors in social media during breaking news, have highlighted the urgent need of automatically identifying rumors and verifying their contents. Rumor detection is the task of determining which pieces of information spreading in the social media have unverifiable truth values at the time of posting. This is a crucial and non-trivial task. For long-standing rumors, one can detect or fact-check the incoming text with a training dataset that covers the related events. For breaking news rumors, this data is non-existed and requires zero-shot learning with respect to its temporally evolving topics. It is more challenging to detect breaking news rumors than long-standing ones. First, breaking news covers topics and events that we may not find in the training dataset, which requires a cross-topic consideration in supervised learning. Otherwise, the detection model will very likely overfit the training dataset. Second, breaking news tends to contain new words such as new hashtags or entity names that do not exist in the training dataset. The issue of *Out-of-vocabulary* (OOV) words is another challenge. Emerging rumors contain words that are not in the training samples, especially for the hashtags. Using pre-trained word embedding cannot address this issue because of the new terms that have not been observed before. Moreover, the same terms may have very different meanings when compared to the past, given their context.

To address these challenges, we jointly train a *word2vec* [19] model with an unsupervised objective to learn the word embedding and train a recurrent neural network model with a supervised objective of rumor detection. We propose to train a word2vec model on the fly with the input of a recurrent neural network. Typically, one uses the recurrent neural network to update the word embedding layer. In contrast, we keep a word2vec model parallel to the recurrent neural network and use it to update the embedding space. In this way, our model can incrementally learn the distributed vector representations of words in the input text, capture the deep latent features and their correlations from it, and use them to build a detection model of breaking news rumors. Furthermore, learning the distributed vector representations of terms allows our model to better handle new OOV words of emerging topics of breaking news that were not seen during the training process. We find such a simple design effective to address the aforementioned challenges.

Related research mostly focuses on long-standing rumors. These rumors usually spread on social media websites for a while, causing streams of posts questioning their truth and looking for a confirmation. Thus, long-standing rumors are already known to be rumors and detecting them is relatively straightforward. For that reason, existing work handling long-standing rumors aims at tracking the diffusion of rumors, classifying opinions expressed toward them, or predicting their veracity [26]. In contrast, we aim at detecting emerging rumors of breaking news, which is more challenging. During early stages of breaking news diffusion, when the topic is still hot, emerging rumors spread very fast in social media, with not many posts discussing their truth. In contrast, people tend to spread these rumors and act upon them immediately, which can be extremely damaging. Furthermore, because breaking news tends to generate new unseen topics, work on detecting emerging rumors of breaking news has to be able to handle topic shift issues.

Most existing studies on rumor detection also suffer from another issue: they assume that rumors are always false and aim at predicting these false rumors [26]. This is demonstrated by the design of their experiments where they train their detection models on datasets of long-lasting rumors with the objective of detecting false rumors. This assumption is invalid because rumors are not always false. The term 'rumor' refers to unverified information that can be deemed later to be true or false. Instead, we aim at detecting emerging rumors regardless of their truth value. The goal is to flag micro-posts as rumors, i.e., micro-posts that contain unverified information during the rapid diffusion, and thus minimize their harmful consequences.

In this paper we study the problem of automatically identifying rumors spreading in social media during breaking news diffusion. We propose a new method that incorporates deep learning and representation learning algorithms to automatically identify rumors in social media. The main contributions of this work can be summarized as follows:

- We propose a new semi-supervised learning solution for breaking news rumor detection by combining an unsupervised learning objective with a supervised learning objective. To the best of our knowledge, this is the first work that employs representation learning with a deep learning model for the purpose of emerging breaking news rumor detection on social media

- We propose a new strategy to update word embeddings on the fly with the training process to mitigate the cross-topic and OOV issue in breaking news rumor detection. In contrast to existing work, we do not train our model based on hand-crafted features. Instead, our proposed model learns distributed representations on parallel to the supervised training.

- Experimental results on real-life datasets suggest that our proposed method outperforms the state-of-the-art sequential classifier [28], as well as other classifiers in terms of precision, recall, and F1.

There is very limited work targeting the challenge of identifying unverified information circulating social media. The work in [25] is one of the earliest works in this category. The authors proposed to first identify "signal tweets" based on a hand-crafted list of regular expressions. Our proposed model learns the features automatically rather than using a predefined hand-crafted regular expressions list. Recently, a sequential classifier model based on the *Conditional Random Fields (CRF)* was proposed to learn the context of an event from the sequence of tweets seen so far and use it to classify the current tweets [28]. Our model predicts the class of the micro-post solely based on its text. It does not need historical data or a trail of micro-posts regarding the information in question.

The rest of the paper is organized as follows. Section 2 covers related work. Section 3 provides an important background knowledge followed by an overview of the proposed model. Section 4 describes the experiments and provides a detailed discussion of the results. Finally, Section 6 concludes this work.

2. Related Work

This section provides an overview of some important related work in the literature.

2.1. Rumor Detection and Analysis

There has been an increasing interest in rumor detection and analysis in social media in the last few years. The nature of the textual data and how fast it spreads in social media raised the need of building tools capable of automatically identifying rumors and assessing their veracity. Work in this field falls in one of four categories: rumor detection, rumor tracking, rumor stance classification, and rumor veracity classification[26]. To illustrate the difference, this section briefly describes each of them and covers some representative work.

Rumor detection is the first and most important task in which unverified information spreading across social media is identified. All subsequent tasks rely heavily on it. Improving the accuracy of this task can indirectly improve the result of the subsequent tasks. However, there has been very little work in this category. Zhao et al. [25] proposed the first method that starts by identifying "signal tweets" that are then grouped into different clusters, each representing a rumor. Next, the summary of each cluster is used to retrieve more related tweets. Finally, the clusters are ranked in the order of their likelihood of being rumors. The proposed framework is entirely based on a list of user-defined regular expressions. This list needs to be periodically revised and updated in order for the model to better handle new unseen stories. On the other hand, our proposed model employs representation learning that learns and exploits the lexical and temporal features of rumor micro-posts in a completely unsupervised manner. Another work in this category is the one presented in [28]. The authors proposed a rumor detection model based on a sequential classifier where a tweet is classified as a potential rumor or non-rumor based on previous data. Although this method achieves higher performance than previous work, it suffers from the cold start problem [28], meaning that the performance of the proposed sequential classifier depends on the sequences of tweets encountered so far. Our proposed model overcomes this problem by classifying each micro-post solely based on its features. It does not need sequences of micro-posts and it performs better than [28] in detecting breaking news rumors in terms of precision, recall, and F1.

Rumor tracking also gains limited attention in the literature. The research problem here is to determine if a given micro-post is related to one of the rumors known in advance. The first work in this category was proposed in [20]. The authors proposed a supervised machine learning approach to judge the relevance of new tweets to the known set of rumors. In [7], the authors proposed a tweet latent vector representation of tweets and used the *Semantic Textual Similarity (STS)*[6] to assess the relevance of new tweets to the known rumors.

Rumor stance classification is a well-studied problem: given a set of micro-posts related to a rumor, classify the orientation expressed in the text of a micro-post as supporting, denying, or questioning the rumor. Most existing works in this category are supervised learning where a predictive model is trained based on different features. The first and most cited work is [20], which proposed several content-based, network-based, and micro-blog-based features. There is a family of works [7, 14, 15, 28] that focus on introducing new features and studying their performance with different classifiers.

Rumor Veracity classification is another well-studied problem in the literature. Most existing works in this category [13, 24, 14, 17] also employ supervised learning where predictive models are trained based on different features to determine the veracity of rumors spreading in social media. Unsupervised methods were recently proposed to tackle this problem, including *recurrent neural networks (RNN)* [16] and *recurrent neural networks with attention mechanism* [10, 3]. This problem is sometimes referred to as rumor detection, where authors of such works adopt an invalid definition of rumors as being "false" pieces of information. Thus, the goal of these proposed methods is to predict the truth value of an unverified story rather than detecting these unverified stories.

Although approaches proposed in the last three categories are beneficial for handling long-standing rumors, their applicability to handle breaking news rumors of emerging topics might be limited. They are based on the assumption that the rumor is already known and a stream of micro-posts about it is available. They skip the first and most important step in the process of detecting and analyzing rumors, which is identifying these rumors in the first place.

2.2. Fake News Detection

In this section, we provide an overview of a closely related problem to rumor detection and analysis known as fake news detection. Fake news refers to news articles that are intentionally written to contain false information. The task of detecting fake news has recently attracted a lot of attention as an emerging research field, especially on social media. The goal of this category of work is to predict if a news article is fake or not. Work in this field can be broadly categorized into two families: identifying check-worthy news articles and predicting the veracity of these articles. This section highlights some of the recent contributions in each of these categories.

Identifying check-worthy news articles is related to the problem studied in this paper. Hassan et al. [8] tackled this problem by proposing a supervised learning method. They first constructed a dataset of spoken sentences labeled as non-factual sentence, unimportant factual sentence, or important factual sentence. Next, *Naive Bayes (NB)*, *Support Vector Machine (SVM)*, and *Random Forest (RF)* multi-class classifiers were used to identify sentences belonging to each of the three categories.

Predicting the veracity of the identified check-worthy news articles is highly related to the problem of rumors veracity classification. It is also a well-studied problem. Different machine learning algorithms were used to tackle this problem such as SVM, *bi-directional long short-term memory networks (Bi-LSTM)*, *convolutional neural networks (CNN)* [23], RNNs [21], *homogeneous credibility networks* [12], and *heterogeneous credibility networks* [11].

3. Deep Learning Model for Breaking News Rumors Detection

This section first formally defines the research problem, presents some background knowledge on recurrent neural networks (RNNs), and, finally, presents the proposed deep learning model for detecting breaking news rumors.

3.1. Problem Statement

The research problem of breaking news rumors detection can be defined as follows: for a given micro-post regarding a specific piece of information, the task is to determine if it is a rumor or not. This problem can be formulated as a binary classification problem as follows: let $w = \langle w_1, ..., w_T \rangle$ be a sequence of words in a micro-post w of length T. Given w as an input, the goal is to classify it as a rumor or not by assigning a label from $L = \{R, NR\}$.

3.2. Recurrent Neural Network

Recurrent Neural Networks (RNNs) represent a rich family of feed-forward neural networks used mainly to handle variable-length sequential or time series data. RNNs have been used for many tasks, including sequence generation and classification. A standard RNN works as follows [5]: Given an input vector sequence x of length T, denoted by $x = \langle x_1, \ldots, x_T \rangle$, for each time step $t = 1$ to T, the algorithm iterates over the following equations to update the hidden states of the network $h = \langle h_1, \ldots, h_T \rangle$ and generate the outputs $o = \langle o_1, \ldots, o_T \rangle$ [16]:

$$h_t = tanh\left(Ux_t + Wh_{t-1} + b\right) \tag{1}$$

$$o_t = Vh_t + c \tag{2}$$

where the terms W, U, and V denote weight matrices connecting hidden to hidden, input to hidden, and hidden to output layers, respectively, and the terms b and c denote bias vectors. The $tanh\left(.\right)$ denotes a hyperbolic tangent non-linear function.

RNNs, especially the above vanilla RNN, are incapable of learning long-distance temporal dependencies. The gradient of current time stamp completely depends on the next time stamp during the back-propagation step, which will cause the gradient to either vanish or explode [16]. This limitation of standard RNNs in storing information about previous input led to extended RNNs architectures designed to store previously seen information in a better way such as *Long Short-Term Memory (LSTM)* [9, 5] and *Gated Recurrent Unit (GRU)*[4].

3.3. Proposed Model

This section provides an overview of the proposed breaking news rumors detection model. The proposed model jointly trains a word2vec model with an unsupervised objective to learn the word embedding and train a recurrent neural network model with a supervised objective of rumor detection. Fig. 1 illustrates the architecture of the proposed model. We will start by describing two main components of our model, namely word2vec and LSTM-RNN, followed by a brief description on how the two models are jointly trained using the input data.

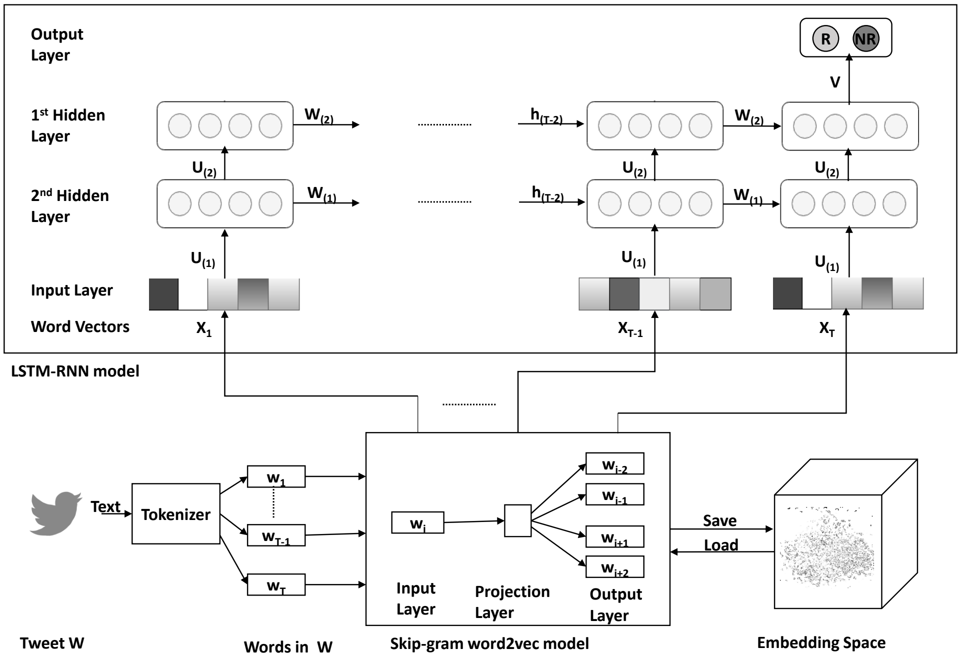

Figure 1: The proposed breaking news rumors detection model. A tweet w is first tokenized into a sequence of words $w = \langle w_1, ..., w_T \rangle$. Next, the word2vec model converts the sequence of words into a sequence of vectors $x = \langle x_1, \ldots, x_T \rangle$ and passes it through weighted connections to the LSTM-RNN model. Finally, the LSTM-RNN model predicts the class as the output vector at the last time step T.

3.3.1. word2vec

A word2vec model is a neural network that takes a text corpus as an input and produces real-valued low-dimensional vector representations for words that appear in that corpus. Thus, it converts textual data into distributed vector representations that can be then fed into deep neural networks for different purposes. These vector representations are called *word embeddings*. In this work, we use a technique called *skip-gram* to train the word2vec model [19] given its better effectiveness compared to the *cbow* model. Given a corpus of text, skip-gram builds the word2vec model as follows. Let w_i be a word in the corpus, and let the set of words surrounding w_i within a specified window size in a sentence be the context of w_i. To build the word2vec model, skip-gram takes each word w_i along with its context words and learns their word representations. The learning objective here is to find useful representations of these words in the embedding space so that the model can, given any other word w_t, predict its surrounding context words with high probabilities and the others with low probability [19]. Formally, given a sequence of words $w = \langle w_1, ..., w_T \rangle$ and a context window of size z, the objective of a skip-gram model is to maximize the following average log probability function:

$$\frac{1}{T} \sum_{t=1}^{T} \sum_{-z \leq j \leq z, j \neq 0} \log p\left(w_{t+j} | w_t\right) \tag{3}$$

where $\log p\left(w_{t+j}|w_t\right)$ is approximated using negative sampling as follows:

$$\log p\left(w_{t+j}|w_t\right) = \log \sigma({v'_{w_{t+j}}}^{\top} v_{w_t}) + \sum_{i=1}^{k} E_{w_i \sim P_n(w_{t+j})}\left[\log \sigma(-{v'_{w_i}}^{\top} v_{w_t})\right] \tag{4}$$

where v_{w_t} and $v'_{w_{t+j}}$ denote the input and output vector representations of words w_t and w_{t+j}; k denotes the number of negative samples for each data sample, and $P_n(w_{t+j})$ denotes the noise distribution [19].

3.3.2. LSTM-RNN

LSTM extends the standard RNNs by introducing a memory cell c_t at each time step t. In this case, the algorithm iterates over the following equations to update the hidden states of the network and generate the outputs[5]:

$$i_t = \sigma\left(W_i x_t + U_i h_{t-1} + V_i c_{t-1} + b_i\right) \tag{5}$$

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + V_f c_{t-1} + b_f\right) \tag{6}$$

$$c_t = f_t c_{t-1} + i_t tanh\left(W_c x_t + U_c h_{t-1} + b_c\right) \tag{7}$$

$$o_t = \sigma\left(W_o x_t + U_o h_{t-1} + V_o c_t + b_o\right) \tag{8}$$

$$h_t = o_t tanh\left(c_t\right) \tag{9}$$

where σ is the logistic sigmoid function, and i, f, o, and c are the input, forget, output gates, and the cell input activation vector, respectively. For each training micro-post, the predicted class is calculated using a softmax layer with the objective of minimizing the following cross entropy loss:

$$L = -\frac{1}{N}\sum_{i=1}^{N}\left[y\log(p) + (1-y)\log\left(1-p\right)\right] \tag{10}$$

where N represents the number of training samples, y represents the actual class, and p represents the predicted class. LSTM provides more channels for the gradient to flow back from time step t to time step $t-1$ by introducing the concept of gates. The gradients do not completely depend on a single time stamp and the vanishing or exploding issues is mitigated by gating.

3.3.3. Model training

We train our model as follows. We first feed the training corpus of micro-posts to the combined skip-gram-word2vec model, which automatically learns the distributed vector representation of each word, i.e., word embedding. This converts the sequence of words in w into a sequence of vectors $x = \langle x_1, \ldots, x_T\rangle$ that is passed through weighted connections to a stack of LSTM hidden layers to compute the hidden vector sequences $h = \langle h_1, \ldots, h_T\rangle$. The predicted class is then calculated as the output vector at the last time step o_T of the LSTM-RNN model.

To help the training process mitigate the cross-topic and OOV issues in breaking news rumor detection, we keep the word2vec model parallel to the recurrent neural network model and use it to update the embedding space on the fly. By designing our model in this way, we incrementally learn the distributed vector representations of words in the input text, capture the latent features and their correlations from the text, and use them to build a detection model of breaking news rumors. We compare the performance of different embedding training strategies in Section 4.4.3. The experimental result shows that this approach significantly outperforms the typical methods of embedding training.

4. Experiment

This section first describes the datasets, baseline methods, features sets, and experimental settings. Next, the obtained results are discussed in detail. Finally, two case studies of real-life breaking news events are presented.

4.1. Datasets

In our experiments, we used five sets of real-life tweets from *PHEME* [27], where each set is related to a piece of breaking news. *PHEME* is publicly accessible. Table 1 summarizes the percentages of rumors and non-rumors tweets in each of them.

Table 1: Percentages of rumors and non-rumors tweets in the PHEME datasets

Breaking News	**Rumors**	**Non-rumors**
Charlie Hebdo	458 (22.0%)	1,621 (78.0%)
Ferguson	284 (24.8%)	859 (75.2%)
Germanwings Crash	238 (50.7%)	231 (49.3%)
Ottawa Shooting	470 (52.8%)	420 (47.2%)
Sydney Siege	522 (42.8%)	699 (57.2%)

4.2. Baselines and Feature Sets

To evaluate our model, we compared it with the state-of-the-art sequential classifier proposed in [28]. We also compared our model with other non-sequential classifiers that were used extensively as baselines in the literature, including *Support Vector Machine (SVM)*, *Naive Bayes (NB)*, *Random Forest (RF)*, and *Maximum Entropy (ME)*.

To train the baseline classifiers, we used the same sets of content-based and social-based features that yielded the state-of-the-art performance in [28]. Table 2 summarizes these two sets of features.

Table 2: Content-based and social-based features

Category	Features
Content-based	word vectors
	Capital ratio: ratio of capital letters
	#Qmark: number of question marks
	#Emark: number of exclamation marks
	#Periods: number of periods
	#Words: number of words
Social-based	#Tweets: number of tweets written by the author
	#Lists: number of lists that include the author's account
	Follow ratio: the following ratio of the author's account
	Age: the age of the author's account
	Verified: whether the account of the author is verified or not

4.3. Experimental Settings

To simulate a real-life cross-topic emerging rumor detection scenario, we performed a 5-fold cross-validation as follows. In each run, we used the datasets of four breaking news stories to train our model as well as the baseline classifiers. The fifth dataset was then used to evaluate the performance of these classifiers in terms of precision, recall, and F1. Thus, in each of the five runs, the dataset used for the evaluation represents breaking news rumors of unseen topics. Furthermore, to ensure the stability of the reported results and get a more robust estimation of the classification performance of our deep learning model, we repeated each run of the 5-fold cross-validation for each model configuration five times. In the rest of this paper, the classification performance of the proposed model is reported as the $mean \pm variance$ of five repetitions of the 5-fold cross-validation instead of a single 5-fold cross-validation run.

The proposed model was implemented using *JetBrains IntelliJ IDEA* [3] development environment and *Deeplearning4j* [4] machine learning library . We ran our experiments on a machine running *Windows server 2016 Datacenter.* The machine has a 32 GB of RAM and is powered by an Intel Xeon E5-1650 v4 processor at 3.60 GHz.

4.4. Results

4.4.1. Comparison with baseline classifiers

To compare the performance of our proposed model with the baseline classifiers, we performed a 5-fold cross validation and reproduced the results of [28], as shown in Table 3. The reported values are the micro-averaged scores across all five runs in terms of precision, recall, and F1 for both classes: rumors and non-rumors. Bold values indicate the best classification performance among all classifiers. For our proposed model, the reported

[3] Source: `https://www.jetbrains.com/idea/`, retrieved on September 28, 2018

[4] Source: `https://deeplearning4j.org/`, retrieved on September 28, 2018

Table 3: Micro-averaged precision (p), recall (R), and F1 scores of detecting rumors and non-rumors across all five runs for baseline classifiers and our proposed model

Classifier	**Features**	**Rumors**			**Non-rumors**			**All Classes**
		P	**R**	**F1**	**P**	**R**	**F1**	**F1**
Support Vector Machine (SVM)	**Content-based**	0.351	0.431	0.387	0.668	0.590	0.626	0.536
	Social-based	0.347	0.479	0.402	0.666	0.536	0.594	0.517
	Combined	0.353	0.457	0.399	0.671	0.569	0.616	0.531
Random Forest (RF)	**Content-based**	0.299	0.092	0.141	0.655	0.889	0.754	0.618
	Social-based	0.343	0.460	0.393	0.662	0.545	0.598	0.495
	Combined	0.326	0.104	0.158	0.658	0.889	0.757	0.622
Naive Bayes (NB)	**Content-based**	0.402	**0.767**	0.527	0.775	0.412	0.538	0.533
	Social-based	0.259	0.011	0.020	0.659	**0.984**	0.789	0.653
	Combined	0.402	0.767	0.527	0.775	0.412	0.538	0.533
Maximum Entropy (ME)	**Content-based**	0.362	0.473	0.410	0.678	0.570	0.619	0.537
	Social-based	0.368	0.495	0.422	0.684	0.563	0.617	0.540
	Combined	0.364	0.472	0.411	0.679	0.575	0.623	0.540
Conditional Random Field (CRF)	**Content-based**	0.687	0.544	0.607	0.788	0.872	0.828	0.761
	Social-based	0.467	0.259	0.333	0.690	0.848	0.761	0.648
	Combined	0.665	0.548	0.601	0.787	0.858	0.821	0.752
Proposed model	words	**0.728** ±0.002	0.706 ±0.0005	**0.716** ±0.001	**0.833** ±0.0003	0.847 ±0.001	**0.839** ±0.0004	**0.795** ±0.001
	Combined	0.619 ±0.005	0.670 ±0.003	0.639 ±0.001	0.821 ±0.0003	0.778 ±0.008	0.796 ±0.003	0.741 ±0.002

values are the micro-averaged ± variance scores across five repetitions of the 5-fold cross-validation. As shown in the table, results for the rumors class suggest that among all baseline classifiers, NB had the best performance in terms of recall, while Conditional Random Fields (CRF) performed the best in terms of precision and F1. This is consistent with the results reported in [28]. Table 3 also shows that our proposed model outperformed CRF in terms of precision, recall, and F1.

For the non-rumors class, similar results were obtained. Among all baseline classifiers, CRF had the best performance in terms of precision and F1, while NB performed the best in terms of recall. Our proposed model also outperformed all baselines in terms of precision and F1. It achieved a high recall as well.

Table 3 also shows that our proposed model had the best overall performance for both classes: rumors and non-rumors, compared to all baseline classifiers in terms of F1. These results suggest that our model outperformed all baseline classifiers, including the state-of-the-art model, in detecting breaking news rumors using *only the text of tweets* as input without any social-based features.

4.4.2. Experimenting with syntactic representations of posts

To further evaluate the classification performance of our model, we experimented with the following syntactic representations of tweets as our input:

- *Part-Of-Speech tags (POS).* Inspired by work on sensitive text detection [18], we wanted to explore whether or not representing a tweet as a sequence of POS tags can lead to better classification performance. We used GATE Twitter part-of-speech

Table 4: Micro-averaged ± variance of precision (p), recall (R), and F1 scores of detecting rumors and non-rumors across all five runs for our proposed model using other syntactic features

Features	**Rumors**			**Non-rumors**			**All Classes**
	P	**R**	**F1**	**P**	**R**	**F1**	**F1**
1-gram words	0.728 ±0.002	0.706 ±0.0005	0.716 ±0.001	**0.833** ±0.0003	0.847 ±0.001	0.839 ±0.0004	**0.795** ±0.001
2-gram words	0.478 ±0.002	0.431 ±0.007	0.447 ±0.002	0.706 ±0.004	0.737 ±0.009	0.719 ±0.005	0.631 ±0.004
3-gram words	**0.884** ±0.0002	**0.740** ±0.001	**0.806** ±0.0002	0.542 ±0.001	0.759 ±0.003	0.632 ±0.002	0.746 ±0.0004
3-gram characters	0.420 ±0.002	0.612 ±0.009	0.494 ±0.002	0.734 ±0.001	0.555 ±0.014	0.626 ±0.007	0.575 ±0.003
5-gram characters	0.496 ±0.003	0.589 ±0.008	0.533 ±0.002	0.778 ±0.001	0.700 ±0.011	0.732 ±0.004	0.662 ±0.003
7-gram characters	0.316 ±0.031	0.199 ±0.022	0.239 ±0.026	0.646 ±0.001	0.788 ±0.004	0.709 ±0.001	0.583 ±0.002
Part Of Speech (POS) tags	0.433 ±0.021	0.154 ±0.004	0.207 ±0.005	0.793 ±0.001	**0.927** ±0.005	**0.853** ±0.0002	0.752 ±0.0004

tagger, known as *"Twittie"*[5], to tag words in our datasets. Then, we replaced each word in every tweet by its POS tag and used the sequences of POS tags as our input.

- *N-gram words and N-gram characters.* We also represented each input tweet as a sequence of N-gram words or N-gram characters to further explore whether or not such representations can improve the classification performance of our model.

In this experiment, we set $N = 1, 2, 3$ for N-gram words and $N = 3, 5, 7$ for N-gram characters. We then performed 5 repetitions of a 5-fold cross validation and evaluated our model using different input representations. Table 4 shows the micro-averaged ± variance scores across five repetitions of the 5-fold cross-validation in terms of precision, recall, and F1 for both classes: rumors and non-rumors. Bold values indicate which input representation yielded the best classification performance of our model. For the rumors class, the results suggest that representing the input tweets as sequences of 3-gram words yielded the best classification performance over all other representations in terms of precision, recall, and F1. 2-gram words also yielded a good classification performance. On the other hand, representing tweets as sequences of N-gram characters did not yield as good of a performance as N-gram words. The results also suggest that using POS tags representations yielded high classification performance in terms of precision, but the recall and F1 scores were low. For the non-rumors class, among all input representations, using the text of the tweet yielded the best classification performance in terms of precision,

[5] Source: `https://gate.ac.uk/wiki/twitter-postagger.html`, retrieved on January 24, 2018

while the POS tags representation yielded the best classification performance in terms of recall and F1.

Table 4 also shows that our proposed model yielded the best overall performance in terms of F1 for both classes when the input is simply the text of the tweets.

4.4.3. Comparison of Different Embedding Training Strategies

To assess if knowledge transfer can help improve the classification performance of our deep learning model, we compared the performance of our model using three different settings for learning the distributed vector representation of words via the word2vec model:

- *Static word2vec model.* In this setting, during the training phase we used the training datasets to jointly learn the word2vec and LSTM-RNN models. Then, to evaluate our model, the word2vec model was used as a lookup table to transform every new tweet in the testing dataset into a sequence of vector representations of its words, which was then fed into the LSTM-RNN model. 14

- *Dynamic word2vec model.* In this setting, during the training phase we used the training datasets to jointly learn the word2vec and LSTM-RNN models. Then, to evaluate our model the word2vec model was incrementally up-trained and updated while classifying every new tweet in the testing dataset.

- *Up-trained Google word2vec model.*[6] In this setting, instead of learning the distributed vector representations of words from scratch, we used a general word2vec model as our initial distributed victor representations of words. This model was trained on Google's news dataset to contain three million words and phrases, each represented as a 300-dimensional vector in the embedding space. During the training phase, Google's word2vec model was first up-trained using our training datasets in parallel with building the LSTM-RNN model. Then, to evaluate our model, this word2vec model was incrementally up-trained and updated while classifying every new tweet in the testing dataset.

Table 5 shows the micro-averaged ± variance scores of our model under each of the three settings across five repetitions of the 5-fold cross-validation in terms of precision, recall, and F1 for both classes: rumors and non-rumors. Bold values indicate which setting yielded the best classification performance of our model. The results suggest that using dynamic word2vec setting yielded a significantly better classification performance than the static word2vec for the rumors class in terms of recall and F1, while it improved the performance on the non-rumors class in terms of precision, recall, and F1. In the experiment the size of the testing dataset is smaller than the training set. Since the quality of the distributed vector representation of words tends to increase significantly with amount of the input data, the dynamic word2vec setting should yield even better classification performance in the long term. The results also suggest that although the

[6] Source: `https://code.google.com/archive/p/word2vec/`, retrieved on May 11, 2018

Table 5: Micro-averaged ± variance of precision (p), recall (R), and F1 scores of detecting rumors and non-rumors across all five runs for our proposed model under different settings of training word2vec model

Word2vec model	**Rumors**			**Non-rumors**			**All Classes**
	P	**R**	**F1**	**P**	**R**	**F1**	**F1**
Static model	0.710 ±0.003	0.696 ±0.002	0.703 ±0.002	0.716 ±0.0005	0.747 ±0.001	0.731 ±0.0002	0.734 ±0.001
Dynamic model	**0.728** ±0.002	**0.706** ±0.0005	**0.716** ±0.001	**0.833** ±0.0003	**0.847** ±0.001	**0.839** ±0.0004	**0.795** ±0.001
Up-trained Google model	0.668 ±0.001	0.552 ±0.0003	0.604 ±0.003	0.751 ±0.002	0.816 ±0.002	0.782 ±0.0007	0.719 ±0.002

idea of transfer knowledge using a pre-trained embedding from Google seems promising, it did not improve the classification performance of our model in terms of precision, recall, or F1. These results suggest that building the word2vec model in parallel with building the LSTM-RNN model helps the rumors detection model learn the latent features and their correlations from the input text. Furthermore, updating the word2vec model incrementally with every new tweet helps the model mitigate the topic-shifts and OOV issues associated with emerging breaking news rumors.

4.4.4. Characterizing datasets

During our experiments we observed that using social-based features in addition to content-based features as our input did not always improve the classifiers. In this section we aim to assess the effect of adding the social-based features to the content-based features of each of the datasets on the classification performance. We started by evaluating the precision of each classifier on each dataset twice: once using only content-based features and another using both social-based features and content-based features as our input. Table 6 shows the obtained results. Bold values indicate cases where the precision of a classifier was improved after adding social-based features. The results show that the precisions of four classifiers were improved after adding the social-based features for the *Ferguson* dataset compared to only one classifier for the rest of the datasets.

These results led us to analyze the social-based and the content-base features of each

Table 6: Precision scores of different classifiers before and after using social-based features associated with each dataset

Dataset	**NB**		**ME**		**RF**		**SVM**		**CRF**		**Proposed Model**	
	Before	**After**	**Before**	**After**	**Before**	**After**	**Before**	**After**	**Before**	**After**	**Before**	**After**
Charlie Hebdo	0.756	0.756	0.568	**0.578**	0.687	0.687	0.571	0.556	0.823	0.807	0.684	0.630
Ferguson	0.253	**0.254**	0.563	**0.578**	0.714	0.704	0.519	**0.543**	0.778	0.773	0.680	**0.714**
Germanwings Crash	0.508	0.508	0.520	0.510	0.484	**0.505**	0.582	0.548	0.731	0.702	0.806	0.802
Ottawa Shooting	0.527	0.527	0.501	0.491	0.478	**0.484**	0.512	0.508	0.697	0.709	0.896	0.844
Sydney Siege	0.428	0.428	0.494	0.488	0.564	**0.582**	0.491	0.488	0.697	0.691	0.803	0.779

Table 7: Importance scores of each of the features in each dataset measured as the gain ratio between this feature and the true class label

Dataset	**Content-based Features**					**Social-based Features**				
	Capital Ratio	**#Qmark**	**#Emark**	**#Periods**	**#Words**	**#Tweets**	**#Lists**	**Follow Ratio**	**Age**	**Verified**
Charlie Hebdo	0.010	**0.032**	**0.024**	0.008	0.005	0.015	0.020	0.000	0.009	**0.038**
Ferguson	0.011	**0.020**	0.005	0.010	**0.018**	0.010	**0.014**	0.000	0.003	0.000
Germanwings Crash	0.010	**0.019**	0.004	**0.029**	0.007	0.004	0.008	0.000	0.005	**0.022**
Ottawa Shooting	**0.031**	**0.127**	**0.054**	0.003	0.019	0.020	0.023	0.000	0.016	0.003
Sydney Siege	**0.054**	0.047	**0.056**	0.004	0.005	0.029	**0.105**	0.000	0.008	0.044

of the datasets. We started by measuring the importance of each of the features in predicting the true class of tweets in each of the datasets using the *gain ratio* feature selection algorithm [1]. Table 7 shows the obtained results. Bold values indicate the top important features in each case. The results show that the number of lists that include the author's account, denoted by *#Lists*, is an important social-based feature for the *Ferguson* and the *Sydney Siege* datasets, while *verified* (whether the author's account is verified or not) is an important social-based feature for *Charlie Hebdo* and *Germanwings Crash* datasets. We further analyzed the social-based features of each of the datasets and used the *Standard Deviation (SD)* to measure the amount of variation in their values. Table 8 shows the obtained results. Bold values indicate cases where the SD value of the feature in a dataset varies significantly from the rest of the datasets. The standard deviation values in the table shows the sparsity in the values of each social-based feature in each one of the five datasets. Each column represents the amount of variation in one social-based feature. The different scales are due to the fact that different features have very different value scales. As shown in the table, among the four datasets with important social-based features, the *Ferguson* dataset can be characterized by the very low SD value of the *#Lists* feature compared to the rest of the datasets. Similarly, the *Sydney Siege* dataset can be characterized by the high SD value of the *#Lists.* On the other hand, the SD values of the *Verified* feature in the *Charlie Hebdo* and *Germanwings Crash* datasets are almost the same as the rest of the datasets, which does not help characterize these datasets.

By comparing our results in Tables 6, 7, and 8, we observed that although the *Ferguson* and the *Sydney Siege* datasets can be distinguished from the other datasets by having a social-based feature with high important score and very different SD value, adding the social-based features improved the classification performance for most classifiers for the first dataset, compared to only one classifier for the second one. The very high SD value of the *#Lists* feature in the *Sydney Siege* dataset suggests much higher sparsity in its values. Consequently, instead of improving the classification performance, adding this feature actually worsened it.

In general, the nature of breaking news and its diffusion patterns reduce the effect of using social-based features to distinguish rumors from non-rumors micro-posts for many reasons. First, breaking news mainly spread on Twitter as trending stories and hashtags. Taking a glance at any trending breaking news hashtag clearly shows the high diversity

Table 8: Standard Deviation values of social-based features for the PHEME datasets

Dataset	**#Tweets**	**#Lists**	**Follow Ratio**	**Age**	**Verified**
Charlie Hebdo	56305.081	33348.537	1.552	1.950	0.498
Ferguson	58165.469	**12054.331**	1.094	1.783	0.483
Germanwings Crash	67650.101	30550.214	1.438	2.158	0.483
Ottawa Shooting	55850.439	32896.770	1.489	1.604	0.468
Sydney Siege	53221.181	**71941.379**	1.549	1.952	0.483

in social-based features of the participants. Furthermore, predefined features are known to be data or domain dependent. Meaning that the effect of different types of features depends on the quality of the dataset and how informative these features are in that specific dataset. For instance, many work in the literature on veracity classification and stance classification of long-standing rumors have experimented with social-based features as well as many other types of features and have reported contrasting results on different datasets. Finally, predefined lists of features need to be periodically revised and updated in order for the model to better handle new data. In the case of emerging breaking news rumors, even when a model is trained on a high quality data where the social-based features are very informative, the model may not perform well with new data. This is a major advantage of our proposed model which will learn the latent features and their correlations from the input text itself, rather than depending on a predefined list of features. Our design also allows the model to automatically learn new features from every new data it receives and dynamically update itself to better handle it.

4.5. Case studies

In this section, two case studies of real-life breaking news events are first presented and followed by a brief discussion of the obtained results[7].

4.5.1. Case Study 1: Detecting rumors of emerging sub-topics of a breaking news

To demonstrate the performance of our model on a real-time Twitter stream of a breaking news sub-topics, we collected tweets about an emerging breaking news story stating that the U.S. government lost track of almost 1,500 unaccompanied immigrant children after placing them in sponsors' homes[8]. This breaking news has recently become viral on Twitter with thousands of people wondering in the hashtag *#WhereAreTheChildren* about many aspects of the news. Although this news was verified in general, many

[7] Labeled data available at: `http://dmas.lab.mcgill.ca/data/RumorsNonRumorsCaseStudyData.zip`.

[8] Source: `https://www.cnn.com/2018/05/26/politics/hhs-lost-track-1500-immigrant-children/index.html`, retrieved on May 28, 2018

Table 9: The classification performance of our model on a real-life breaking news case study in terms of precision (p), recall (R), and F1

	P	**R**	**F1**
Rumor	0.786	0.647	0.710
Non-rumor	0.700	0.824	0.757
Both classes	0.743	0.735	0.757

Table 10: Examples of tweets collected from a real-life breaking news and how it was classified by our model.

Tweet text	**Truth**	**Classified**
So, about that prison bus for babies. . . , it actually takes charter school kids on field trips.	rumor	rumor
This administration is a real beauty. HOW in hades do you lose almost FIFTEEN HUNDRED CHILDREN?	non-rumor	non-rumor
How is it fake news? It's from their website and is literally a prison bus for babies. Why do you think the babies are there?	rumor	non-rumor

tweets are spreading rumors about different aspects and details of the story. These rumors are not yet confirmed nor refuted by the government. We collected 50 tweets about this breaking news and manually fact-checked each of them and kept only the 34 tweets we know belong to one of the two classes: rumors[9] and non-rumors[10]. We then fed those tweets into our model to classify each of them as a rumor or not. Table 10 shows examples of the collected tweets and how they were classified by our model. Table 9 shows the classification performance of our rumor detection model when applied on these tweets in terms of precision, recall, and F1. These results suggest that our model is capable of detecting breaking news rumors of unseen topics with high accuracy.

4.5.2. Case Study 2: Detecting rumors of multiple emerging breaking news topics.

We performed another case study to demonstrate the performance of our model on detecting different emerging topics of multiple breaking news in a real-time Twitter stream. We started by collecting tweets about the following three unverified breaking news stories that have recently emerged and are not yet confirmed nor refuted by the government:

- *"449,000 California residents turned down jury duty because they are not U.S. citizens, despite being registered to vote"*[11]. This news spread very fast in social media

[9] Source: https://www.snopes.com/fact-check/prison-bus-for-babies/, retrieved on May 29, 2018

[10] Source: https://www.snopes.com/fact-check/1475-immigrant-children-missing/, retrieved on May 29, 2018

[11] Source: https://www.snopes.com/fact-check/did-449000-californians-turn-down-jury-duty-because-they-are-undocumented-immigrants/, retrieved on Dec 24, 2018

Table 11: The classification performance of our model on a real-life multiple breaking news case study in terms of precision (p), recall (R), and F1

	P	**R**	**F1**
Rumor	0.810	0.756	0.782
Non-rumor	0.766	0.818	0.791
Both classes	0.788	0.787	0.791

and even more claims were added by users overtime. Nevertheless, this news is not verified yet.

- *"Guatemalan authorities rescued a group of minors from human smugglers in the migrant caravan"*[12]. This news is still unverified regardless of the claims about the existence of exclusive information and photos from high-level Guatemalan government official.

- *"The U.S. Attorney for the Southern District of New York has begun the prosecution of President Trump's inauguration committee as of December 2018*[13]. Although this claim was published by reputable news organizations, it is still unverified and is based only on information from unnamed sources.

Furthermore, to demonstrate a real-life scenario where Twitter streams are not limited to predefined events or topics, we collected general streams of tweets from the following two major sources of breaking news:

- *An official Twitter account of a well-known news agency.* We collected all tweets in the first 2 pages of the timeline of the *CNN's* Twitter account[14]. These tweets represent a real-time stream of micro-posts about unspecified topics of regular as well as breaking news and events currently occurring all over the world.

- *A general all-time trending hashtag.* We also collected all tweets in the first 2 pages of the timeline of a general widely-adopted fashion hashtag, namely #OOTD [15]. We choose this hashtag for two main reasons. First, fashion data in this hashtag represents unseen general topics that are not news-related. This simulates an everyday general real-time Twitter stream. Second, similar to a trending breaking news hashtag, trending fashion hashtags always contain tweets with many new and emerging topics, terms/vocabulary, and named entities.

[12] Source: `https://www.snopes.com/fact-check/guatemala-smugglers-children/`, retrieved on Dec 25, 2018

[13] Source: `https://www.snopes.com/fact-check/trump-entities-criminal-probe/`, retrieved on Dec 20, 2018

[14] Source: `https://twitter.com/CNN`, retrieved on Dec 25, 2018

[15] Source: `https://twitter.com/search?vertical=default\&q=\%23OOTD&src=typd`, retrieved on Dec 25, 2018

Next, we manually fact-checked each of the collected tweets and kept only the 89 ones we know belong to one of the two classes: rumors and non-rumors. We then randomly shuffled these tweets and fed them into our detection model. Table 11 shows the classification performance of our rumor detection model when applied on these tweets in terms of precision, recall, and F1. These results suggest that our model is capable of detecting multiple breaking news rumors of unseen topics in an every day Twitter stream with high accuracy.

4.5.3. Discussion of case studies results

To further understand the obtained results of our rumor detection model, we closely inspected the text of tweets that were correctly classified and compared it with tweets that were misclassified in the two case studies. We had two main observations. First, we noticed a high similarity in the writing styles among most rumor tweets. Similarly, most non-rumor tweets also have its own writing style. This observation can be further inspected in the future by proposing a breaking news rumor detection model that is conditioned on the different writing styles of tweets. Second, we noticed the existence of many new OOV terms and named entities that were not originally trained by our model such as *Inauguration*, *Guatemala*, *smugglers*, *Trump*, *immigrants*, and *outfit*. The results of the case studies suggest that our model can adaptively capture the drift and mitigate the OOV and topic-shift issues in breaking news rumor detection.

5. Limitation

According to our adopted definition where a rumor is defined as "a story or a statement whose truth value is unverified", rumors do not have to be false; they can be deemed later to be true or false. This definition implies that an emerging tweet that was flagged as rumor can later be non-rumor. However, our proposed model does not explicitly model or memorize the facts across time. To address this issue, the proposed model can be combined with a long-lasting rumor detection model. The proposed model is responsible for flagging and storing the emerging rumors, and the long-lasting rumor detection model can be trained when facts are checked.

However, our experiment and case studies show that despite our model does not explicitly model and memorize facts across time, it performs fairly well by just looking at the tweet at current moment. We suspect that there may be two reasons. First, the word2vec model is incrementally updated. It may memorize new concepts and drift over time. Secondly, the proposes model may memorize to distinguish how rumors and non-rumors are conveyed in natural language. They may correspond to a very different writing style, which coincides with our observations in case studies.

6. Conclusion

With the increased adaptation of social media as the main source of breaking news, distinguishing verified information from unverified rumors becomes an extremely difficult and crucial task. Several characteristics of social media facilitate the process of posting information with unestablished truth values and the fast diffusion of them among users all over the world. Breaking news rumors, if not identified as early as possible, may have extremely damaging consequences. In this work, we tackle the problem of identifying breaking news rumors of emerging topics spreading on Twitter by proposing a model that jointly builds the word2vec model and the LSTM-RNN rumor detection model. The proposed model is capable of accurately identifying breaking news rumors based solely on a tweet's text. Our experiments on real-life datasets show that the performance of our proposed model outperforms the state-of-the-art classifier as well as other baseline classifiers in terms of precision, recall, and F1.

Acknowledgments

The first author is supported by King Saud University in Riyadh, Saudi Arabia, and the Saudi Arabian Cultural Bureau in Canada. The third author is supported by the Discovery Grants (RGPIN-2018-03872) from the Natural Sciences and Engineering Research Council of Canada (NSERC) and Canada Research Chairs Program (950-230623). The last author is supported by the Zhejiang Provincial Natural Science Foundation of China (LY17F020004), and the National Natural Science Foundation of China (61272306). We would like to thank the reviewers for the thorough review and valuable comments, which significantly improve the quality of this manuscript.

References

[1] T. Abeel, Y. Van de Peer, and Y. Saeys. Java-ml: A machine learning library. *The Journal of Machine Learning Research*, 10:931–934, June 2009.

[2] G. Allport and L. Postman. *The psychology of rumor.* Russell & Russell, 1965.

[3] T. Chen, L. Wu, X. Li, J. Zhang, H. Yin, and Y. Wang. Call attention to rumors: Deep attention based recurrent neural networks for early rumor detection. *CoRR*, abs/1704.05973, 2017.

[4] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *CoRR*, abs/1409.1259, 2014.

[5] A. Graves. Generating sequences with recurrent neural networks. *CoRR*, abs/1308.0850, 2013.

[6] W. Guo and M. Diab. Modeling sentences in the latent space. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, pages 864–872, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[7] S. Hamidian and M. Diab. Rumor identification and belief investigation on twitter. In *Proceedings of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, 2016.

[8] N. Hassan, C. Li, and M. Tremayne. Detecting check-worthy factual claims in presidential debates. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, pages 1835–1838, New York, NY, USA, 2015. ACM.

[9] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computing*, 9(8):1735–1780, Nov. 1997.

[10] Z. Jin, J. Cao, H. Guo, Y. Zhang, and J. Luo. Multimodal fusion with recurrent neural networks for rumor detection on microblogs. In *Proceedings of the 2017 ACM Conference on Multimedia*, MM '17, pages 795–816, New York, NY, USA, 2017. ACM.

[11] Z. Jin, J. Cao, Y.-G. Jiang, and Y. Zhang. News credibility evaluation on microblog with a hierarchical propagation model. In *Proceedings of the 2014 IEEE International Conference on Data Mining*, ICDM '14, pages 230–239, Washington, DC, USA, 2014. IEEE Computer Society.

[12] Z. Jin, J. Cao, Y. Zhang, and J. Luo. News verification by exploiting conflicting social viewpoints in microblogs. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 2972–2978. AAAI Press, 2016.

[13] S. Kwon, M. Cha, and K. Jung. Rumor detection over varying time windows. *PLOS ONE*, 12(1):1–19, January 2017.

[14] X. Liu, A. Nourbakhsh, Q. Li, R. Fang, and S. Shah. Real-time rumor debunking on twitter. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1867–1870, New York, NY, USA, 2015. ACM.

[15] M. Lukasik, K. Bontcheva, T. Cohn, A. Zubiaga, M. Liakata, and R. Procter. Using Gaussian processes for rumour stance classification in social media. *CoRR*, 2016.

[16] J. Ma, W. Gao, P. Mitra, S. Kwon, B. J. Jansen, K.-F. Wong, et al. Detecting rumors from microblogs with recurrent neural networks. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 3818–3824. AAAI Press, 2016.

[17] J. Ma, W. Gao, Z. Wei, Y. Lu, and K.-F. Wong. Detect rumors using time series of social context information on microblogging websites. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1751–1754, New York, NY, USA, 2015. ACM.

[18] G. McDonald, C. Macdonald, and I. Ounis. Using part-of-speech n-grams for sensitive-text classification. In *Proceedings of the 2015 International Conference on The Theory of Information Retrieval*, ICTIR '15, pages 381–384, New York, NY, USA, 2015. ACM.

[19] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, pages 3111–3119, USA, 2013. Curran Associates Inc.

[20] V. Qazvinian, E. Rosengren, D. R. Radev, and Q. Mei. Rumor has it: Identifying misinformation in microblogs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1589–1599, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[21] N. Ruchansky, S. Seo, and Y. Liu. Csi: A hybrid deep model for fake news detection. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, pages 797–806, New York, NY, USA, 2017. ACM.

[22] E. Shearer and J. Gottfried. News use across social media platforms 2017, September 2017.

[23] W. Y. Wang. "liar, liar pants on fire": A new benchmark dataset for fake news detection. *CoRR*, abs/1705.00648, 2017.

[24] Z. Yang, A. Kotov, A. Mohan, and S. Lu. Parametric and non-parametric user-aware sentiment topic models. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 413–422, New York, NY, USA, 2015. ACM.

[25] Z. Zhao, P. Resnick, and Q. Mei. Enquiring minds: Early detection of rumors in social media from enquiry posts. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1395–1405, Republic and Canton of Geneva, Switzerland, 2015. International World Wide Web Conferences Steering Committee.

[26] A. Zubiaga, A. Aker, K. Bontcheva, M. Liakata, and R. Procter. Detection and resolution of rumours in social media: A survey. *ACM Computing Surveys*, 51(2):32:1–32:36, Feb. 2018.

[27] A. Zubiaga, G. W. S. Hoi, M. Liakata, and R.Procter. Pheme dataset of rumours and non rumours. Nov. 2016.

[28] A. Zubiaga, M. Liakata, and R. Procter. Learning reporting dynamics during breaking news for rumour detection in social media. *CoRR*, 2016.