



Deep Recurrent Neural Network and Data Filtering for Rumor Detection on Sina Weibo

Yichun Xu^{1,*}, Chen Wang¹, Zhiping Dan¹, Shuifa Sun^{1,2} and Fangmin Dong^{1,*}

- ¹ College of Computer and Information, China Three Gorges University, Yichang 443002, China; wangchen@ctgu.edu.cn (C.W.); zp_dan@ctgu.edu.cn (Z.D.); watersun@ctgu.edu.cn (S.S.)
- ² Department of Electrical Engineering and Computer Science, University of Wisconsin-Milwaukee, Milwaukee, WI 53201, USA
- * Correspondence: xuyichun@ctgu.edu.cn (Y.X.); fmdong@ctgu.edu.cn (F.D.)

Received: 12 October 2019; Accepted: 13 November 2019; Published: 15 November 2019



MDF

Abstract: Social media makes it easy for individuals to publish and consume news, but it also facilitates the spread of rumors. This paper proposes a novel deep recurrent neural model with a symmetrical network architecture for automatic rumor detection in social media such as Sina Weibo, which shows better performance than the existing methods. In the data preparing phase, we filter the posts according to the followers of the user. We then use sequential encoding for the posts and multiple embedding layers to get better feature representation, and multiple recurrent neural network layers to capture the dynamic temporal signals characteristic. The experimental results on the Sina Weibo dataset show that: 1. the sequential encoding performs better than the term frequency-inverse document frequency (TF-IDF) or the doc2vec encoding scheme; 2. the model is more accurate when trained on the posts from the users with more followers; and 3. the model achieves superior improvements over the existing works on the accuracy of detection, including the early detection.

Keywords: rumor detection; Sina Weibo; Deep Neural Network; recurrent neural network (RNN); long short-term memory (LSTM); gated recurrent unit (GRU)

1. Introduction

Social media based on the internet is becoming an important news source for many people. It was reported that in 2017 about two-thirds of Americans got their news from Facebook, YouTube, and Twitter, etc. [1]. Sina Weibo, a Chinese microblogging platform, declared that there were over 100 million people discussing the news of the FIFA 2018 World Cup [2].

It is very convenient for people to publish and consume news on social media, but at the same time, the rumors or fake news spread more easily there. Sina Weibo reported that about 28 thousand rumors were debunked in 2017 [3]. The social media rumors are very harmful to our community, for example, on 23 April, 2013, the official Twitter account of Associated Press was hacked to publish the fake news that two explosions happened in the White House and the President was injured, which resulted in a dramatic crash of the stock market immediately [4]. To prevent rumors from spreading, the organizations such as Associated Press, FactCheck.org and Snopes.com provide functions to check the facts and alert the public. Sina Weibo also set up an account @weibopiyao for users to report the possible fake news. These efforts actually work, yet they depend on the intervention of the users and the experts.

In order to improve the efficiency of rumor detection, the researchers applied artificial intelligence to the problem. Automatic rumor detection is treated as a pattern recognition problem, since the veracity of the news in social media can be judged by the features extracted, such as the texts, the behaviors of the authors, the propagation modes, and the images or videos [5–8]. Many works [9–16] were based on traditional supervised learning methods, in which feature engineering is very important but painstakingly labor intensive.

In recent years, the deep neural networks have made significant progresses in pattern recognition problems, such as the deep convolutional neural network (CNN) for image processing [17] and short text classification [18,19]. The neural networks were also applied in the area of rumor detection. Ma et al., proposed a tree structure recursive neural network to model the propagation of rumors [20]. Nguyen et al. [22] combined CNN and the RNN to obtain a credit score for a single post. Compared to CNN, RNN is more suitable for modeling the temporal characteristics of rumor spreading. Some researchers proposed RNN models for rumor detection in the event level and they obtained good results and avoided the manual feature engineering [21,23].

However, the data representation is still an important factor which affects the performance of existing RNN models. The authors of [23] chose TF-IDF to represent the texts, which is good for the global statistical information of the words, but misses the structural information in the sentences. The authors of [21] used doc2vec technology for the text, which can reserve the structural information, but needs much time and space for the model.

To address the above problems, in this paper we propose a novel deep recurrent neural network (DRNN) model for rumor detection based on the textual information. The DRNN model includes eight layers totally. The input layer receives the stream of posts related to an event, and the sequential coding scheme is applied. The first three hidden lays, including a normalization layer and two fully-connected layers, are inserted for the better representation. The next two hidden layers are RNN layers, which can capture the dynamic temporal signals characteristic in the post stream. Lastly, a fully-connected layer outputs the probability of being a rumor. The proposed model is symmetric in architecture, that each cell in the upper layer is fully connected to the cells in the layer below, and each cell in the layer below is also fully connected to the cells in the upper layer.

Considering that comments and replies to the posts are often very short and contain only similar words, we propose a sequential encoding scheme for the texts. The experiments show that it has better performance than the popular TF-IDF and the doc2vec encoding.

Another contribution of this paper is that we propose a data filtering technique in the data preparation stage. We assume that the users of social media with many followers usually know more than others in certain fields, and in order to maintain the followers, they will be more careful to speak the right things when they make assertions about something. So, we can depend more on their opinions when making the rumor classifications. Our work proves that the models are actually more accurate if they are trained on the posts submitted by the users with more followers.

By experiment, the proposed system achieves superior improvements over the existing works on the accuracy of detection, including the early detection.

1.1. RNN

RNN is the type of neural network with recurrent connections that can make predictions by time serials. Given an input sequence ($x_1, x_2, ..., x_t, ...$), simple RNN units can general the hidden states ($h_1, h_2, ..., h_t, ...$):

$$h_t = \tanh(Wx_t + Uh_{t-1} + b). \tag{1}$$

where W and U are the weight matrices, b is the bias, and tanh is the hyperbolic tangent non-linearity function.

It was found that in the training process, the long-term information often vanished gradually with the time step. Then, LSTM units were proposed to memorize more long term information [24]:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f). \tag{2}$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i). \tag{3}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o).$$
 (4)

$$c_t = f_t * c_{t-1} + i_t * \tanh(W_c x_t + U_c h_{t-1} + b_c).$$
(5)

$$h_t = o_t * \tanh(c_t). \tag{6}$$

where σ is the logistic sigmoid function, f_t , i_t , and o_t are the activation vectors of the forget gate, input gate, and output gate, c_t is the cell state vector, and * denotes the Hadamard product.

GRU is a simpler version of LSTM units with fewer parameters [25]:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z). \tag{7}$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r). \tag{8}$$

$$h_t = z_t * h_{t-1} + (1 - z_t) * \tanh(W_h x_t + U_h(r_t * h_{t-1}) + b_h).$$
(9)

where z_t and r_t are the vectors of update gate and reset gate.

In many cases, the recurrent neural network with LSTM or GRU units has better performance than the simple RNN.

The RNN states can be decomposed into multiple layers [26]. It was further suggested that the separate multiple layer perceptron can be inserted before the input, the hidden, or the output layer of the RNN for better performance [27], which can lead to "deeper" RNN (DRNN).

1.2. Automatic Rumor Detection

Some automatic rumor detection works [9–16] on microblogging platforms were based on the traditional machine learning methods, in which the feature engineering is painstakingly labor intensive.

Ma et al. [23] firstly proposed RNN models on the rumor detection problem to avoid the feature engineering, where the messages were organized as a time series input. Ruchansky et al. [21] proposed an RNN model combining the message text and the user information. Nguyen et al. [22] used CNN and RNN to get a credit score for each post and then combined with other hand-craft features. Ma et al. [20] applied the recursive neural network on the problem by analysis of the tree structure of the propagation modes.

Our work is mostly related with [21,23], yet there are three major differences. The first difference is that they used TF-IDF [23] and doc2vec [21] to encode the input, while our work directly used the simple sequential encoding scheme. The TF-IDF information only keeps the frequency of text, but loses the structural information in the sentences. The doc2vec scheme needs much more training to get suitable vectors. The second difference is that our RNN model is much deeper than theirs. The new architecture can prevent the system from under fitting and over fitting. The third difference is that we filtered the dataset and used the posts from users with more followers to train the model.

2. Materials and Methods

Although our methods are mainly proposed to detect the rumors on Sina Weibo, we still applied them on a public Twitter dataset to check the performance. So, in this section, we first describe a Sina Weibo dataset and a Twitter dataset, and then explain the details of the methods to train and evaluate the models on the datasets to detect rumors.

2.1. Dataset

2.1.1. Sina Weibo dataset

We evaluate the proposed DRNN models on the dataset provided in [23], which is crawled from Sina Weibo. The details about the dataset are listed in Table 1.

In the dataset, there are totally 4664 events. In average, there are about 816 posts related to an event and a user contributed 1.3 posts in the dataset. For a detailed look, we drew two histograms to show the distribution of posts as in Figure 1.

# of rumor	2313
# of non-rumor	2351
# of posts	3,752,459
# of users	2,819,338
Post-based features	post time, text, repost count, comment count, etc.
User-based features	registered time, location, follower count, friend count, post count, etc.

Table 1. Dataset from Sina Weibo.



Figure 1. Number of posts published at different times (**a**) and number of posts published by the users with different numbers of followers (**b**).

Figure 1a is about the number of posts published at different times. The publishing speed is the highest in the first 4 h, and then slows down very quickly with time. About 62% of the posts are published during the first 12 h. Figure 1b is about the number of posts published by different users. We can see that the users with more followers publish fewer posts.

2.1.2. Twitter Dataset

We also evaluated our model on a public Twitter dataset released in [12], which provides all the information we need: the text, the post time, and the follower count. The dataset includes 111 events (60 rumors and 51 non-rumors), and 192,350 posts altogether.

2.2. Methods

2.2.1. Problem Definition

An event in a microblogging platform is defined as $e = \{w_0, r_1, r_2, ..., r_p\}$, where w_0 is the original post published by a user, $r_1, r_2, ..., r_p$ are the *p* posts commented or reposted by other users and sorted by the publication time. An event can be labeled as a rumor or not. We should train a DRNN model to predict the label of an event.

We only use the text body of each post as the input of DRNN, so in the following sections, post also means its text body.

2.2.2. Text Vectorization

(1) Word Segmenting and Sequential Encoding

Due to the fact that in Chinese sentences the words are adjacent and there is no blank to separate two words, we firstly use a natural language processing tool named Jieba (https://pypi.org/project/jieba/) to segment each post into a list of words. Replacing each word with its index in the dictionary, we sequentially encode the post to be a one-dimensional integer vector. Unlike the TF-IDF scheme in [11], our scheme reserves the sentence structure and the semantic information, which should be important for the classification task.

Comparing to the doc2vec scheme, our sequential encoding scheme needs less time and space expenses. The shortcoming of the sequential encoding is that the distance of two words cannot be measured so it is hard to express the homoionym; however in the microblogging system, the comments of posts are usually very short and resemble one another, so the network can memorize most of them.

(2) Post Division Series

In general *p* is usually very large (on average, an event may contain above 816 posts in the Sina Weibo dataset), but it is complex to design an RNN to support so many time steps, so we equally partition the comments and reposts into *t* groups according to the amount of them. With k = p/t posts in a group, we get $e = \{(w_0), (r_1, r_2, ..., r_k), (r_{k+1}, r_{k+2}, ..., r_{2k}), ..., (r_{(t-1)k+1}, r_{(t-1)k+2}, ..., r_p)\}$. Each group will be fed to DRNN as a time step. Note that w_0 separately becomes a group because it is more important than other comments and reposts.

We do not make the partition according to time division as [11,13] because of the uneven distribution of the post in the time dimension, which will be discussed in Section 4.1.

(3) Data Filtering

We infer that if a microblog user has more followers, he perhaps knows more about how to judge the truth of news. Furthermore, he will be more careful when making assertions in order to maintain followers. We sort the posts in each group by the number of followers in descending order, and then retain the posts from the user with more followers and filter out what is left.

(4) Concatenating and Fixing

At last, we concatenate the remaining posts in each group into a long vector, and then fix the elements of the long vector by reserving the first *m* elements if its length is greater than *m*, or padding some zeros behind if its length is less than *m*. Thus, we have a final sequence $\{x_0, x_1, ..., x_t\}$, where each x_i is an *m*-element integer vector.

2.2.3. DRNN Architecture

We propose a deep RNN architecture with eight layers as Figure 2 and the structural information is listed in Table 2.



Figure 2. Deep recurrent neural network (DRNN) architecture.

Table 2. The description of the architecture.

Layer	Name	Description
1	Input	The input layer accepts x_0, x_1, \ldots, x_t sequentially and the dimension of x_i is 3000.
2	Norm	This layer normalizes the inputs
3	Fully-Conn1	A fully-connected layer with ReLU activation, transforms the dimension of the data to 800.
4	Fully-Conn2	A fully-connected layer with ReLU activation, transforms the dimension of the data to 256.
5	RNN1	RNN layer, transforms the dimension of the data to 32.
6	RNN2	The second RNN layer, keeps the dimension of the data.
7	Fully-Conn3	A fully-connected layer with Sigmoid activation, transforms the dimension of the data to 1
8	Prob	The output layer, which is the probability of an event being a rumor.

The first layer is the input layer and it accepts $x_0, x_1, ..., x_t$ sequentially, which are prepared as the descriptions in Section 2.2.2. In our work, *m* was set to 3000, so the dimension of the first layer is 3000.

The second layer is a normalization layer, which applies a transformation on each x_i to make the mean close to 0 and the standard deviation close to 1 [28]. This layer can accelerate the training of the network. With a batch of training samples, this layer outputs:

$$\widetilde{h}_t = \frac{x_t - E(x_t)}{\sqrt{\operatorname{var}(x_t) + \varepsilon}}.$$
(10)

$$h_t^2 = \gamma \widetilde{h}_t + \beta. \tag{11}$$

where γ and β are parameters to be learned and ε is a hyper parameter. The superscript denotes the layer index.

The third and the fourth layer are two fully-connected layers with ReLU activation. The purpose of these two layers is to reduce the dimension of the input data flow from 3000 to 800 and then to 256. We use two layers for the advantage of "deep". The outputs of these two layers are:

$$h_t^3 = \operatorname{Re}LU(W^3h_t^2 + b^3) \text{ and } h_t^4 = \operatorname{Re}LU(W^4h_t^3 + b^4).$$
 (12)

The fifth and sixth layers are two RNN layers. They can be simple RNN or LSTM or GRU as introduced in Section 1.1. We set the dimensions of the two layers to 32 and get h_t^5 and h_t^6 .

The seventh layer is a fully-connected layer that transforms the output from the sixth layer into a real number, and then we use the sigmoid function to get a probability value *p* in the last layer:

$$p = \sigma(W^7 h_t^6 + b^7). \tag{13}$$

3. Results

In the experiments, we set t to 32 and m to 3000. We train the proposed DRNN models with the cross entropy loss function and the Adagrad optimizer with learning rate 0.001. 80% of posts are randomly chosen as the train set and the others as the test set. The metrics such as Accuracy, Precision, Recall and F1 measure [29] are used to evaluate the model performance:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, Precision = \frac{TP}{TP + FP}, Recall = \frac{TP}{TP + FN}, F1 = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}}$$
(14)

where TP, TN, FP, FN are, respectively, the number of rumors correctly detected, number of non-rumors correctly detected, number of non-rumors but wrongly detected as rumor, and number of rumors but wrongly detected as non-rumors.

With three kinds of RNN layers in the DRNN architecture, we abbreviate the corresponding DRNN models as DSRNN (simple RNN), DLSTM, and DGRU in the following texts, tables and graphics to avoid confusion.

The experiments in Sections 3.1–3.3 are on Sina Weibo dataset, and the last experiment in Section 3.4 is on Twitter dataset.

3.1. Filtering Data by the Followers

In this experiment, we study the relationships between the accuracy of models and the followers of users. At first, we extract four subsets from the original Sina Weibo dataset. The four subsets are limited that the followers of the user are in intervals of [0,100), [100,300), [200,500), and $[500,\infty)$. As shown in Table 3, the posts in each subset are about 24% to 32% of the total (There are overlaps between the subsets [100,300) and [200,500), but we prioritized to maintain nearly 30% of posts per subset. The overlaps do not affect the conclusion that the accuracy gets better when training on the

subset with more followers.). Next, we train three kinds of DRRN models on each subset. The accuracy of the models is shown in Figure 3.



Table 3. Data subsets.

Figure 3. The accuracy of three models on different data subsets.

3.2. Comparisons with Different Models

We then perform a comparison of our results with [23] and [21] on the Sina Weibo dataset in Table 4. The results of three kinds of DRNN models are on interval of $[500,\infty)$.

Literature	Model	Accuracy	Precision	Recall	F1
[23]	SVM-TS	0.857	0.839	0.885	0.861
	GRU-2	0.910	0.876	0.956	0.914
[21]	CI	0.928	-	-	0.927
Proposed	DSRNN	0.926	0.947	0.898	0.918
	DGRU	0.942	0.968	0.916	0.940
	DLSTM	0.954	0.975	0.935	0.953

Table 4. Performance of different models.

The results of SVM-TS and GRU-2 are from [23], where SVM-TS is a support vector machine model based on time series and hand-craft features, and GRU-2 is an RNN model with two layers of GRU. The results of CI are from [21], where the authors used an LSTM model.

3.3. Early Detection

It is important to detect the rumors in the early stage of their spreading. As indicated in Figure 1a, the early detection is possible because most of the posts are published in the early stage. We firstly filter the Sina Weibo dataset by the publishing time and get the data subset in the first 4, 8, 12, 24, 48, 72, and 96 h, and then train the proposed DRNN models on each subset with a further filtering which requires users to have more than 500 followers.

The accuracy curves of the proposed DRNN models are presented in Figure 4a accompanied by a curve of GRU-2 from [23]. The loss curves are presented in Figure 4b.



Figure 4. Performance of the proposed models and GRU-2 in [11] for the early detection. (**a**) The accuracy of each model; and (**b**) LSTM and GRU models converge faster than SRNN.

3.4. Extensions to Twitter Dataset

The rumor detection methods are not limited on a certain kind of dataset, so then it is valuable to apply our method to the Twitter dataset. In this experiment, we also want to study the accuracy of our methods and the relationships with the follower count of the user. We first choose 4 subsets in which the follower count of the users is in the interval of [0,80), [80,250), [250,700), and $[700,\infty]$. The posts in each subset account for 21% to 30% of the total. We train and evaluate the DLSTM model on the Twitter dataset with the same parameters. The accuracy of the models is listed in Table 5.

Table 5. Accuracy of DLSTM on the Twitter dataset.

Interval of	[0,80)	[80,250)	[250,700)	(700,∞)	All
followers					data
Accuracy	0.591	0.636	0.682	0.864	0.818

4. Discussion

4.1. Data Filtering

From Table 3, we filter the data by the followers of users and only use about 30% of the data to train the models, which means the data filtering technique can save a lot of training resources, such as memory and CPU time.

From Figure 3, three DRNN models all show that they can get better accuracy when trained with data from users with more followers. The accuracy increases about 3% when trained in the interval of $[500,\infty)$ compared to [0,100). Moreover, the accuracy in this interval is even better than the results from the whole dataset. This can be explained by that users of social media with many followers usually know more than others in certain fields, and in order to maintain their followers, they will be more careful in speaking the right things when they make assertions about something. Therefore, we can depend more on their opinions when we are seeking to identify the truth and falseness of social media events.

Table 5 shows that the data filtering technique also works well in the Twitter dataset. The accuracy of the DLSTM models increases when the training data are from the users with more followers. Compared to interval of [0,80), the accuracy of the model increases about 26% when trained in [700, ∞), and it also outperforms the model from the whole dataset.

4.2. Sequential Encoding

In Table 4, there are 5 kinds of RNN models, where GRU-2 in [23] uses TF-IDF to vectorize the text, CI in [21] uses doc2vec, and our proposed three models use sequential encoding. The results show that the sequential encoding has the best performance in detecting rumors in Sina Weibo. All the proposed three models outperform GRU-2 in accuracy, precision and F1 measure. The DLSTM and DGRU outperform CI in accuracy and F1 measure.

Among the five RNN models, the GRU-2 model in [11] is with the worst performance, which indicates that the TF-IDF encoding limits the classification ability of the model. TF-IDF uses frequency of words as a text feature, but it loses structural information between words in the sentence, and the structural information between sentences.

The doc2vec scheme is a distributed representation of text, and it can measure the distance or similarity of the texts, while the sequential encoding does not have this advantage. However, it is interesting that the sequential encoding also outperforms the doc2vec scheme in the experiments. The results can be explained by the dataset. In social media, the comments and replies to the posts are often very short and contain only similar words. In Sina Weibo, a post usually contains Chinese words below 140 characters, and the comments are even shorter than the original post. For example, most of the comments when reposted are fix words like "reposting" given by the system. The short and similar posts in the dataset mean that the proposed model can memorize most of the information. The doc2vec is a powerful representation of the text, but sometimes it loses information.

4.3. Performance of Different Models

In the experiments on the Sina Weibo dataset, the proposed DRNN models show good performance. From Figure 3, DLSTM is the best model in accuracy. Its best result is about 1.5% higher than the DGRU model and 3% higher than the DSRNN. From Table 4, The DLSTM outperforms the GRU-2 model in [23] and CI model in [21], which are also two kinds of RNN models. Furthermore, the RNN models are about 10% higher than the SVM model, which shows that the RNN models have greater advantages over traditional methods.

DLSTM also shows good performance on the Twitter dataset. From Table 5, DLSTM gets the best accuracy of 0.864, while in [6], the author reported a random forest method with the accuracy of 0.84.

4.4. Early Detection

It is important for the models to detect the rumor as early as possible. In Figure 4a, both the proposed DLSTM and DGRU models show better performance than [23] in all the time intervals. The proposed DLSTM model gets accuracy about 90.6% in the first 4 h, and then raises to 92.4%, 94.1%, and 94.5% in the following 12, 24, and 48 h when the posts gain more and more responses.

The loss curves presented in Figure 4b can explain why the DSRNN models are not as good as the DLSTM and the DGRU. In the first 4 h, the loss of SRNN drops to 0.29 after 80 epochs, but the loss of DLSTM and DGRU drops to 0.25 in the first 40 epochs. The convergence of the DSRNN is improved with the time interval getting large. It can be inferred that the convergence of the DSRNN model depends more on the number of posts.

5. Conclusions

We present a novel deep RNN model for rumor detection in the social media platform, Sina Weibo. This model uses two RNN layers to capture the dynamic temporal signals characteristic in the stream of the post, before which there is a normalization layer and two fully-connected layers for constructing a better representation of the input. By experimental results, we find that: 1. The sequential encoding scheme is suitable for text representation in Sina Weibo; 2. Filtering the dataset and using the posts from users with more followers to train the model can improve the model accuracy; 3. The proposed DRNN model shows better performance than the existing methods on the detection tasks (including

the early detection); and 4. Among three kinds of DRNN models, the model with LSTM layers achieves the best performance.

Future improvements could focus on new encoding schemes and data filtering. Recent development of NLP models can be incorporated into the model to represent the text. The filtering of the dataset may be based on other information, such as the educational background, the occupation, etc.

Another possible improvement is to introduce CNN into the proposed architecture. Precious works [18,19] showed CNN can be used to classify the short sentences, where the sentences are convolved with the templates of words and then the structural patterns are detected. It is possible to use CNN to extract features from the posts before the data enters the cells of RNN layers, so that the structural information and the temporal information are fully utilized.

Author Contributions: Conceptualization, F.D. and S.S.; methodology, Y.X. and Z.D.; software, C.W.; validation, Y.X. and S.S.; formal analysis, Z.D. and C.W.; writing—original draft preparation, Y.X.; writing—review and editing, F.D.; funding acquisition, F.D.

Funding: This work was funded by the National Natural Science Foundation of China, Grant number U1703261, and partially by the National Key Research and Development Program of China, Grant number 2016YFB0800403.

Acknowledgments: We would like to thank Prof. Juan Cao for her helpful suggestions for this paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Pew Research Center. Available online: http://www.journalism.org/2017/09/07/news-use-across-socialmedia-platforms-2017/ (accessed on 31 October 2019).
- 2. Sohu-INC. Available online: http://www.sohu.com/a/246000912_393779 (accessed on 31 October 2019).
- 3. Sina Weibo. Available online: https://weibo.com/1866405545/FFwn4EnNV?type=comment# _rnd1543636471604 (accessed on 31 October 2019).
- 4. CNBC. Available online: https://www.cnbc.com/id/100646197 (accessed on 31 October 2019).
- Zubiaga, A.; Aker, A.; Bontcheva, K.; Liakata, M.; Procter, R. Detection and Resolution of Rumours in Social Media: A Survey. ACM Comput. Surv. 2018, 51. [CrossRef]
- Cao, J.; Guo, J.; Li, X.; Jin, Z.; Guo, H.; Li, J. Automatic Rumor Detection on Microblogs: A Survey. *arXiv* 2018, arXiv:1807.03505.
- 7. Chen, F.; Ji, R.; Su, J.; Cao, D.; Gao, Y. Predicting microblog sentiments via weakly super-vised multi-modal deep learning. *IEEE Trans. Multimed.* **2017**, *20*, 997–1007. [CrossRef]
- 8. Ji, R.; Chen, F.; Cao, L. Cross-Modality Microblog Sentiment Prediction via Bi-Layer Multimodal Hypergraph Learning. *IEEE Trans. Multimed.* **2019**, *21*, 1062–1075. [CrossRef]
- 9. Qazvinian, V.; Rosengren, E.; Radev, D.R.; Mei, Q. *Rumor has it: Identifying Misinformation in Microblogs*; EMNLP 2011: Edinburgh, UK, 2011; pp. 1589–1599.
- 10. Castillo, C.; Mendoza, M.; Poblete, B. Information Credibility on Twitter. In Proceedings of the International Conference on World Wide Web 2011, Hyderabad, India, 28 March–1 April 2011.
- 11. Yang, F.; Yu, X.; Liu, Y.; Yang, M. Automatic Detection of Rumor on Sina Weibo. In Proceedings of the SIGKDD Workshop on Mining Data Semantics, Beijing, China, 12–16 Augest 2012.
- 12. Kwon, S.; Cha, M.; Jung, K. Rumor detection over varying time windows. *PLoS ONE* 2017, 12. [CrossRef]
- Liang, G.; He, W.; Xu, C.; Chen, L.; Zeng, J. Rumor Identification in Microblogging Systems Based on Users' Behavior. *IEEE Trans. Comput. Soc. Syst.* 2015, 2, 99–108. [CrossRef]
- 14. Wu, K.; Yang, S.; Zhu, K. False Rumors Detection on Sina Weibo by Propagation Structures. In Proceedings of the ICDE2015, Seoul, Korea, 13–17 April 2015.
- 15. Ma, J.; Gao, W.; Wei, Z.; Lu, Y. Detect Rumors Using Time Series of Social Context In-formation on Microblogging Websites. In Proceedings of the CIKM2015, Melbourne, Australia, 19–23 October 2015.
- Vosoughi, S. Automatic Detection and Verification of Rumors on Twitter. Ph.D. Dissertation, MIT, Cambridge, MA, USA, 2015.
- 17. Mousavi Kahaki, S.M.; Nordin, M.J.; Ahmad, N.S.; Arzoky, M.; Ismail, W. Deep convolutional neural network designed for age assessment based on orthopantomography data. *Neural Comput. Applic.* **2019**. [CrossRef]

- Kim, Y. Convolutional Neural Networks for Sentence Classification. In Proceedings of the EMNLP 2014, Doha, Qatar, 25–29 October 2014.
- 19. Qin, P.; Xu, W.; Guo, J. An Empirical Convolutional Neural Network approach for Semantic Relation Classification. *Neurocomputing* **2016**, *190*, 1–9. [CrossRef]
- 20. Ma, J.; Cao, W.; Wong, K. Rumor Detection on Twitter with Tree-structured Recursive Neural Networks. In Proceedings of the ACL 2018, Melbourne, Australia, 15–20 July 2018.
- 21. Nguyen, T.N.; Li, C.; Niederée, C. On Early-Stage Debunking Rumors on Twitter: Leveraging the Wisdom of Weak Learners. *LNCS* 2017, 10540, 141–158.
- 22. Ma, J.; Cao, W.; Mitra, P.; Kwon, S.; Jansen, B.J.; Wong, K.-F.; Cha, M. Detecting Rumors from Microblogs with Recurrent Neural Networks. In Proceedings of the IJCAI2016, New York, NY, USA, 9–15 July 2016; pp. 3818–3824.
- 23. Natali, R.; Seo, S.; Liu, Y. CSI: A Hybrid Deep Model for Fake News Detection. In Proceedings of the CIKM2017, Singapore, 6–10 November 2017.
- 24. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. Neural Comput. 1997, 9, 735–1780. [CrossRef]
- 25. Cho, K.; van Merrienboer, B.; Bahdanau, D.; Bengio, Y. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. In Proceedings of the Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST-8), Doha, Qatar, 25–29 October 2014.
- 26. Graves, A.; Mohamed, A.R.; Hinton, G. Speech Recognition with Deep Recurrent Neu-ral Networks. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–30 May 2013.
- 27. Pascanu, R.; Gülçehre, C.; Cho, K.; Bengio, Y. How to Construct Deep Recurrent Neural Networks. In Proceedings of the ICLR 2014, Banff, AB, Canada, 14–16 April 2014.
- 28. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In Proceedings of the ICML2015, Lille, France, 6–11 July 2015.
- 29. Mousavi Kahaki, S.M.; Nordin, M.J.; Ashtari, A.H.; Zahra, J.S. Invariant Feature Matching for Image Registration Application Based on New Dissimilarity of Spatial Features. *PLoS ONE* **2016**, *11*, e0149710. [CrossRef] [PubMed]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).