Attention-Residual Network with CNN for Rumor Detection

Yixuan Chen
chenyixuan17@mails.ucas.ac.cn
University of Chinese Academy of Sciences
Beijing, China

Jie Sui*
suijie@ucas.ac.cn
University of Chinese Academy of Sciences
Beijing, China

Liang Hu
huliang13@mails.ucas.ac.cn
University of Chinese Academy of Sciences
Beijing, China

Wei Gong
gongwei11@mails.ucas.ac.cn
University of Chinese Academy of Sciences
Beijing, China

ABSTRACT

Wide dissemination of unverified claims has negative influence on social lives. Rumors are easy to emerge and spread in the crowds especially in Online Social Network (OSN), due to its openness and extensive amount of users. Therefore, rumor detection in OSN is a very challenging and urgent issue. In this paper, we propose an Attention-Residual network combined with CNN (ARC), which is based on the content features for rumor detection. First, we build a data encoding model based on word-level data for contextual feature representation. Second, we propose a residual framework based on fine-tuned attention mechanism to capture long-range dependency. Third, we apply convolution neural network with varying window size to select important components and local features. Experiments on two twitter datasets demonstrate that the proposed model has better performance than other content-based methods both in rumor detection and early rumor verification. To the best of our knowledge, we are the first work that utilize attention model in conjunction with residual network on rumor detection.

CCS CONCEPTS

• **Information systems → Multimedia information systems**.

KEYWORDS

Rumor Detection, Attention Mechanism, Residual Network, Convolution Neural Network

ACM Reference Format:
Yixuan Chen, Jie Sui, Liang Hu, and Wei Gong. 2019. Attention-Residual Network with CNN for Rumor Detection. In *The 28th ACM International Conference on Information and Knowledge Management (CIKM'19), November 3–7, 2019, Beijing, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3357384.3357950

*Jie Sui is the corresponding author.

1 INTRODUCTION

Online Social Network (OSN) provides an access to gather information about societal issues, and to find out about the latest developments during breaking new stories. Especially, social media platform as an powerful internet-connected tool, because its huge users and ease of information sharing. However, while people share thoughts and post microblogs in real-time, these messages of unverified truth may lead to the spread of rumor.

A rumor is defined as an circulating story or report whose veracity status is yet to be verified at the time of posting [1]. The wide dissemination of rumor may influence development of platforms and cause social panic. To combat these damages, recently, there has been lots of rumor detection research. They aim to build a rumor classification model, which identifies microblogs whose veracity status are uncertain.

Some existing researches based on content typically utilize supervised machine learning algorithm based on hand-crafted features. An obvious limitation is that they rely on the quality of artificial features and ignore the dynamic temporal features during rumor circulation. Some researches tend to apply deep learning methods to automatically detect rumors. Especially, by regarding the microblogs in a specific topic as time-series sequence, Recurrent Neural Network (RNN) is widely used to verify rumor by learning latent contextual features. These works greatly improve performance than feature engineering methods. However, RNN has limitations of capturing long-range dependencies effectively, the reason is that a recurrent layer suffers from the vanishing gradient problem when modeling a long sequence series.

Nowadays, end-to-end memory networks based on attention mechanism instead of sequence recurrence have been shown well performance on question answering task [2]. Motivated by the success of attention mechanism on modeling sequence series, we utilize attention model to draw long-range dependencies by relating different position of a sequence. To learn effective features for rumor detection, instead of extracting a single global contextual feature, we remain extract a local feature map.

In this paper, we propose an attention-residual network combined with CNN (ARC), which is based on the content features for rumor detection. First, we propose an encoding model to learn contextual feature initially. The model encode variable-length word-level sequence to fixed-length event-level feature map by shortening path between element position. Second, we build an attention residual framework to exploit latent global correlations. We adapt three strategies to fine tune attention model, then wrap several fine-tuned attention models into residual network. Third, we apply one-dimensional CNN in a sliding window way two times. The first time is to retain crucial local information from output of the encoding model, the second time is to select important information from global features generated by the attention residual framework. Finally, we obtain accurate content representation by concatenating the local and global feature maps and implement it via a non-linear fully connected layer and softmax layer. We evaluate proposed model on two twitter datasets, one is public dataset with structure features and the other is a complex dataset generated by keywords inquiry. Extensive experiments demonstrate that ARC outperforms existing content-based methods and has a good adaptability on noisy data. Moreover, our model achieves good performance on early rumor detection. The contributions of this paper can be summarized as follows:

1) We build an attention-residual network combined with CNN to exploit latent contextual features for rumor detection.
2) We propose three fine-tuned attention mechanism model to capture long-range dependency. To the best of our knowledge, we are the first work that utilize attention model in conjunction with residual network on rumor detection.
3) Abundant experiments validate the effectiveness and adaptability of our model in rumor detection. Moreover, our model outperforms other models on early rumor detection.

2 RELATED WORK

In this section, first, we will briefly review the related research on rumor detection, and then introduce the key algorithm to be used.

2.1 Rumor Detection

Rumor detection task is a binary classification problem essentially, which classifies unmarked raw data into rumor and non-rumor. Some researches use feature-based machine learning algorithms to analyze and extract effective classification features, and construct a classifier to solve rumor detection problem.

Castillo et al. [3] divided all hand-crafted features into four categories including content-based features, user-based features, topic-based features and propagation features. On the basis of research [3], some works [4–6] have extracted new features and combined multi-modal features into machine learning algorithm for rumor detection. Some people focus on the study of content-based features. Zhao et al. [7] found few posts related with a specific rumor used some enquiry phrases, which are used to express doubt. Cai et al. [8] extracted textual features from crowd responses, which composed of reposts and comments, the experiments show that some stop words used to be regarded as noises can improve classifier accuracy to a certain extent.

Some researches regard the microblogs related to a same event as a continuous time series, that is, a microblog information flow. A time series modeling is applied to capture dynamic characteristics [9]. With the development of deep learning, Recurrent Neural Network(RNN) which can learn long-distance connections is used in the field of rumor detection [10]. Subsequently, In order to detect rumor from different aspects of features, many people apply variants of RNN combined with attention mechanism. Chen et al. [11] applied attention mechanism to Bi-GRU for rumor detection based on content features. Guo et al. [12] built a hierarchical network involving Bi-LSTM and attention mechanism to combine contextual and social features. Existing work suggests multi-modal features can improved accuracy by providing extra informations, but the choose of multiple features is artificial and time consuming.

2.2 CNN

Convolution Neural Network(CNN) models have shown to be effective in the tasks of Natural Language Processing(NLP) [13–15]. Chen et al. [16] proposed a fusion network composed of BiLSTM and CNN to improve sentiment analysis by classifying sentence type firstly. Peng et al. [17] used a graph-CNN model to build Graph-word representations, which have a better performance on large-scale hierarchical text classification problem.

In the filed of rumor detection, few research applied a CNN model to identify rumors by extracting local features. Chen et al. [18] build a CNN model composed of several filters with varying window size, which can obtain different aspects of contextual features, then connected with a softmax layer to compute probabilities of different classes. Considering the great success and adaptability of CNN, we assign a one-dimensional CNN to locate and capture local features.

2.3 Attention Residual Network

In the filed of NLP, Bahdanau [19] firstly introduced the attention mechanism to machine translation model. The core equation is calculated as follow:

$$Attention(Q, K, V) = A(Q, K)V \tag{1}$$

where K, Q, V are three matrix packed by a set of input series; A is a function used to calculate the energy between Q and K, which also includes a normalization operation. The attention operation can highlight important features from the overall data by assigning them a higher weight. In order to improve the accuracy of rumor detection, there have been a lot of works of combining attention mechanism with RNN, they learn important features from computation to the output layer and hidden layer. Liu et al. [20] applied a fusion attention model, which consists of content attention and dynamic attention to learn textual and temporal features, respectively. In addition to strengthening feature learning, attention mechanism can also encode interrelationship of elements in a sequence without being limited by the length [21–23]. Transformer was proposed on machine translation task, it merely based on attention mechanism [2].

In the deep neural network, residual connections [24] are often used to address the degradation problem. Wang et al. [25] proposed a residual attention network for object recognition and image classification, experiments on two datasets demonstrates that residual attention network can greatly improve performance again noisy label.

Motivated by the success of Transformer on capturing long-distance dependencies and effectiveness of residual network on solving degradation problem, we apply several attention layers connected by residual blocks.

3 PROBLEM STATEMENT

On the OSN, individual post has natural sparsity, which is due to the limitation of the word number. Therefore, the truth value of individual microblog is difficult to measure, but the collection of individual microblogs with same topic has relatively stable credibility. Therefore, We focus on the collection of microblogs rather than an individual one.

We define a rumor detection dataset as a set of events $\mathbf{E} = \{e_1, e_2, \cdots, e_m\}$, in which each event e_i is a collection containing n relevant microblogs with a specific topic. An event $e_i = \{s_{i,1}, s_{i,2}, \cdots, s_{i,n}\}$ where each $s_{i,*}$ is a related individual microblog, and each microblog $s_{i,*}$ consists of l words $s_{i,*} = \{w_1, w_2, \cdots, w_l\}$. Note that n microblogs in an event have temporal features because of the distribution over time series, and there may exists some structure relations among n microblogs based on their comment or repost relationships.

We formulate the problem of rumor detection as a binary classification problem, and build an attention-residual network combined with CNN to learn a projection $F(e) \longrightarrow (1, 0)$, in which label 1 and 0 indicate a rumor class and a non-rumor class, respectively.

4 METHODOLOGY

In this section, we will describe ARC in detail. The main component of our deep network consists of two branches, one is to capture long-range global features, the other is to retain important local features. The topmost layer is the softmax classification layer.

4.1 Encoding

Content features serve as important clues for rumor recognition. Traditional methods of learning content features are to use embedding layer or construct neural network to generate word vector matrix, which is input of the classification model. However, an event contains variable number of tweets, which will increase the algorithm complexity. Besides, duplicate microblog often appears in an event, which makes it difficult to effectively represent contextual features in word-level.

To overcome these limitations, we design a data encoding model, as described in Algorithm 1. The model contains three different levels contextual representation, which are word vector, sentence vector and event vector, respectively. The architecture is shown in Figure 1.

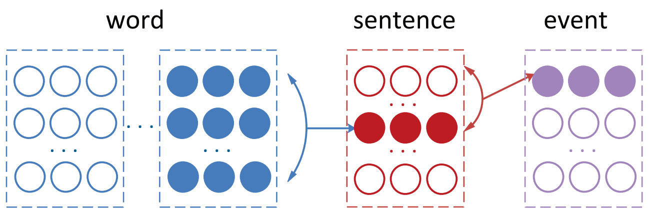

Figure 1: The architecture of encoding model.

First, we remove URLs, user-name with @ tags and topic tags with # to clean data, then segment each microblog content to words. Word embeddings pre-trained by a significant number of unlabeled corpus are more powerful for capturing content properties [26].

Algorithm 1 Data encoding for each event

```
Input: the content series of an event $e = \{s_i\}_{i=1}^n$
       the required fixed length $FN$
Output: $InputV$
 1: for each post $s_i$ do
 2:     $\tilde{s}_i = \{word_{i,1}, \cdots, word_{i,l}\}$ ←Clean and Cut content in $s_i$
 3: end for
 4: Divided content series $\tilde{e} = \{\tilde{s}_i\}_{i=1}^n$ into $FN$ sub-events:
    $e\_s \leftarrow \{\tilde{s}_{(i-1)k+1}, \cdots, \tilde{s}_{ik}\}_{i=1}^{FN}, k \leftarrow \frac{n}{FN}$
 5: for each $e\_s_i$ do
 6:     $sc_i \leftarrow$ Concat all words
 7:     Apply pre-trained word embedding vector with dimension
        $D$ in :
        $\tilde{sc}_i \leftarrow \{v_{sj}\}_{j=1}^p, p \leftarrow len(sc_i)$
 8:     $v_{si} \leftarrow \frac{1}{|p|}\sum_{i\in p} w_i v_i$
 9: end for
10: $InputV \leftarrow [v_{s1}, v_{s2}, \cdots, v_{sFN}]^T \in \mathbb{R}^{FN\times D}$
```

For the representation of word vector, we use the pre-trained word embeddings with dimension D on the Google News dataset [27].

The representation of a sentence vector v_s can be learned by the following equation:

$$v_s = \frac{1}{|l|}\sum_{i\in l} w_i v_i \tag{2}$$

where l denotes the number of word vector v_i in the sentence vector v_s. In this paper, we adopt three weighted functions, each of them corresponds to a calculating method.

The first method is an unweighted average operation (UNW), as described in Equation (3), which has been proved efficient in representing short phrases [28].

$$v_{unws} = \frac{1}{|l|}\sum_{i\in l} v_i \tag{3}$$

The second method is smooth inverse frequency (SIF)[29], which computes weighted average of word vectors and removes the most common component.

$$v_{sif\,s} = \frac{1}{|l|}\sum_{i\in l} \frac{c}{c + p(v_i)} v_i \tag{4}$$

where c is a smoothing parameter, and $p(v_i)$ denotes the word i frequency estimated from public dataset.

The third method is a weighted average operation, where the weight is formulated by TF-IDF. In the Equation (5), IDF_i denotes the inverse document frequency of word i, where each sentence is viewed as a document.

$$v_{unws} = \frac{1}{|l|}\sum_{i\in l} IDF_i v_i \tag{5}$$

Finally, for the representation of event vector, we first fix the length of input matrix as FN, then, encode variable-length sentence-level sequence to the fixed-length FN event-level feature map by average dividing. Thus, we satisfy the requirement of fixed length FN of input matrix to decrease computation complexity.

4.2 Residual Attention Framework

In this work, we use a fine-tuned attention model as a basic unit to construct our residual attention framework. As shown in Figure 2.

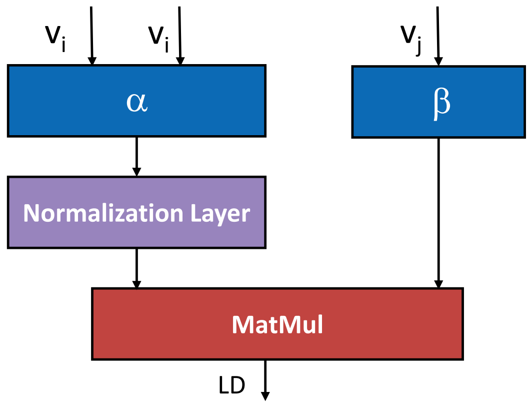

Figure 2: The architecture of a fine-tuned attention model.

Attention mechanism, as described in Section 2.3, is defined as mapping a query and a set of key-value pairs to enlighten relatively important components. In this paper, the attention model is used to capture long-range dependencies by computing a global feature at a position as a sum of all position features, where the query, keys, and values are the same $InputV$ (Algorithm 1). Following the Equation (1), we define fine-tuned attention operation in our paper as:

$$LD_i = \frac{1}{N(v)} \sum_{\forall j} \alpha(v_i, v_j)\beta(v_j) \tag{6}$$

where i means the i-th position of a global feature LD, which is computed by other components; j means the positions of all elements v in the $InputV$; The factor $N(v)$ is a normalization function. The pairwise function α computes the representation of correlation between v_i and v_j, and the function β represents a signal at the position j.

We adapt three algorithms on weighted operation. For simplicity, the function β is implemented as one-dimensional convolution operation, which is set as $\beta(v_j) = Conv1d(v_j)$. Next, we will describe the three choices of the pairwise function α.

4.2.1 Fine-tuned Self-Attention. Following the mean of scaled dot-product attention, we compute the global feature as:

$$LD_i = \sum_{\forall j} softmax(\frac{{v_i}^T v_j}{\sqrt{d_v}})Conv1d(v_j) \tag{7}$$

where the $\frac{1}{N(v)} \sum_{\forall j} \alpha(v_i, v_j)$ is defined by softmax computation along the dimension j for the given position i; d_v is the dimension of v; The normalization parameter D is the length of v, which simplifies gradient computation.

4.2.2 Gaussian. The second definition of function α is Gaussian formulation based on dot-product, we compute the global feature based on Gaussian as:

$$LD_i = \frac{1}{N(v)} \sum_{\forall j} softmax(e^{{v_i}^T v_j})Conv1d(v_j) \tag{8}$$

where a Gaussian function is commonly applied to map the data into a high feature apace. The normalization $N(v)$ is defined as $\sum_{\forall j} e^{{v_i}^T v_j}$.

4.2.3 Embedded Gaussian. Following the Gaussian function choice of function α, a simple extension is to map the high-level feature maps into an embedding space. The function α and the normalization element $N(v)$ in Equation (6) are computed as:

$$\begin{cases} LD_i = \frac{1}{N(v)} \sum_{\forall j} \alpha(v_i, v_j)Conv1d(v_j) \\ \alpha(v_i, v_j) = e^{Conv1d({v_i}^T)Conv1d(v_j)} \\ N(v) = \sum_{\forall j} \alpha(v_i, v_j) \end{cases} \tag{9}$$

where we use a one-dimensional convolution to representation learning of v.

Following the above three candidates of fine-tuned attention models, we propose the residual attention framework, as shown in Figure 3. A building block is defined as:

$$\mathcal{H}(v_i) = \gamma(LD(v_i)) + v_i \tag{10}$$

where γ is a weighted function implemented as one-dimensional deconvolution operation, which is set as $\gamma(LD(v_i)) = DeConv1d(v_i)$. $\mathcal{H}(v_i)$ is the final long-range dependencies computation function. $+v_i$ denotes the residual connection, which neither introduces extra parameter nor increases computation complexity. Note that the fine-tuned attention model maintains the input size that the dimension of $\mathcal{H}(v_i)$ is equal to $InputV$, due to the settings of three functions α, β, γ.

The proposed residual attention framework is powerful for capturing global feature map GF by wrapping several fine-tuned attention models.

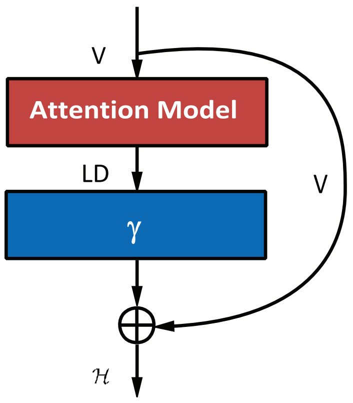

Figure 3: The architecture of a residual block.

4.3 One-Dimensional CNN

The architecture of one-dimensional CNN is shown in Figure 4. Both output matrix of data encoding model and attention residual framework are the input of one dimensional CNN, which is applied for representation learning of local features. Each element $iv_i \in \mathbb{R}^D$ is a D-dimensional vector, which represents i-th position

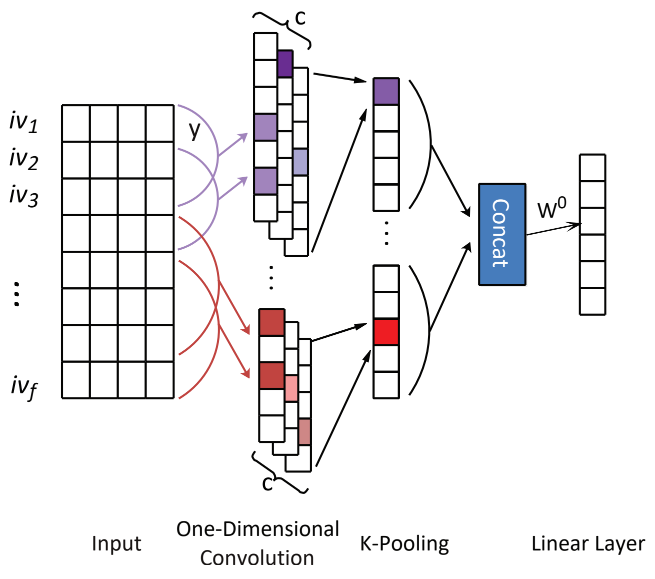

Figure 4: The architecture of a one-dimensional CNN.

in the event-level feature map. A one-dimensional convolution operation uses a filter $f \in \mathbb{R}^{Y \times D}$ to perform collaborative filtering. For instance, a local feature map lf is represented by Equation (11):

$$lf = ReLU\left(f\begin{bmatrix} iv_1 \\ iv_2 \\ \cdots \\ iv_{FN} \end{bmatrix} + b\right) \tag{11}$$

where $lf \in \mathbb{R}^{C\times(FN-Y+1)}$. FN is the fixed-length of input matrix, and C is the number of filters with same window size Y. $ReLU$ is the non-linear function and b is the bias term. Then we assign softmax transformation over the topic variables to ensure maintain the most significant value $\widehat{lf} \in \mathbb{R}^C = softmax(lf)$.

In order to describe local features comprehensively and accurately, we apply multiple filters with varying window size. Finally, we concatenate all maximum values extracted by different filters, and generate the multiple local feature map lf_m as following:

$$LF = concat\left(\widehat{lf}_1, \widehat{lf}_2, \cdots, \widehat{lf}_m\right) \tag{12}$$

where m is the number of filters with different size.

4.4 Pipeline

Take the Embedded Gaussian attention residual framework as a example, we present the pipeline of ARC for rumor detection in Figure 5.

Learning symbolic data represented by distributed vectors is the first step to detect rumor. We take the original word-level embeddings data as input matrix of the encoding model in Section 4.1, which compresses original semantics into feature map $InputV$.

In order to capture different aspects of contextual features, we build two parallelized branches: local branch and global branch.

The local branch involves m one-dimensional CNN models with varying window size, aims to perform spatial feature learning comprehensively. The local feature map LF_L is the concatenation of m dimensional spatial feature vectors.

The global branch consists of a residual framework based on Embedded Gaussian attention and m one-dimensional CNN models. Given the feature map $InputV$, the attention residual framework capture long-range dependencies GF by wrapping several attention model with residual connection. Then m one-dimensional CNN models with varying window size are applied to exploit the compositional semantics on the high-level feature maps. The global feature map LF_G is the concatenation of m dimensional spatial feature vectors.

The configurable one-dimensional convolution operation can be thought of as searching spatial contextual feature maps for latent semantic representations that correspond to specific vector level.

In the process of classification, we concatenation the local features and global features as the final feature map $F_M = (LF_L \oplus LF_G)$, which is the input matrix of a fully connected layer and following softmax layer. The goal is to learn a projection $F(e) \longrightarrow (1, 0)$ with 1 denoting tumor and 0 denoting non-rumor. In the fully connected layer, we adopt non-linear function $ReLU$ to exploit full input field or to focus on fewer elements if needed[30]. During the training process, we assign loss function as the cross-entropy error between prediction distribution l_p and ground-truth l_g series, which is calculated as:

$$L = -\sum_{e\in E}\left[l_g \times log l_p + (1-l_p) \times log(1-l_g)\right] \tag{13}$$

Our training objective is to minimize the loss function given the input e over the entire training set E.

5 EXPERIMENT DESIGN

5.1 Data sets and Data Collection

We study the problem of event-level rumor detection using two Twitter datasets, as shown in Figure 6. The first one is a public twitter dataset proposed by Ma et al. [10], which consists of 992 separate labeled groups. Each group is an event-level collection, containing an initial tweet defined as the specific event topic, all the retweets and comments which are related with the initial tweet. Considering the relation structure among tweets in a group, as shown in Figure 6(a), we call this dataset as TREE. However, the time span between some events in TREE is as long as two years or even longer, which makes the distribution of data over time sparse, and the excessive long time interval leads to the lack of continuous temporal characteristics. In addition, the amount of groups and total tweets also limit the performance of deep neural networks.

To combat the disadvantages, we collect a new dataset by applying keyword-based query method defined by Twitter Monitor [31]. Each keyword represents a specific event topic, which is also the search query. First, we extract keywords from www.snopes.com, which is a confirmed website used to verify credibility of debunking event. Second, We fine tune each keyword by removing some descriptive information while remaining some symbolic participants. Third, we apply Beautiful Soup to pull data out of HTML

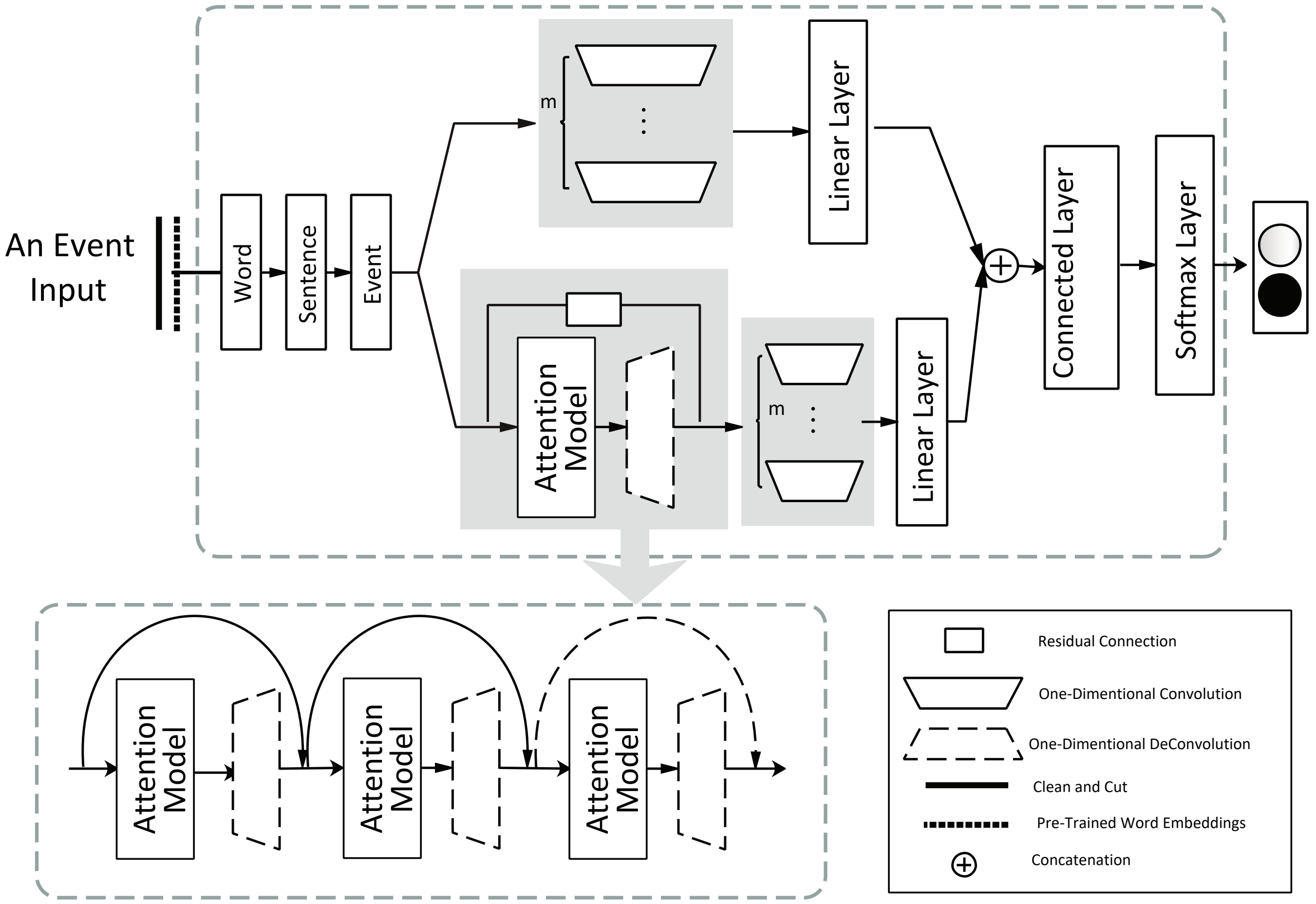

Figure 5: The architecture of Attention-Residual Network combined with CNN (ARC) for rumor detection.

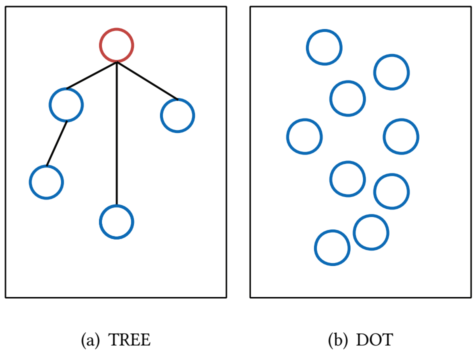

Figure 6: The architecture of two datasets.

results matching the fine-tuned keywords. An event-level collection composes of all relevant tweets without structure relation, and its time span is 30 days from the release time of the initial tweet. Considering that all tweets in each collection may be independent of each other, as shown in Figure 6(b), we call this proposed dataset as DOT. The statistics can be seen in Table 1.

Table 1: Statistic of two datasets.

Dataset	TREE	DOT
Events	992	1,409
Micrologs	581,054	812,862
Max Microlog	38,202	60,122
Min Microlog	4	20
Vocabulary	40,881	71,590
Unknown words	12,032	35,667
Rumor	498	844
Non-Rumor	494	565

5.2 Compared Methods

We compare our model with the following state-of-art models:

- DTR-Enquiry [7] adapted a Ranking model based on Decision Tree to detect rumor, and assigned the enquiry phrases as the classification features. We implement their enquiry phrases and processing of stop words.

- SVM-TS [9] proposed a linear classifier SVM based on time-series structures for rumor detection, and combined three kinds of hand-crafted features. The model greatly improved the performance of machine learning algorithm.
- GRU-2 [10] firstly utilized a RNN model to detect rumor, and designed a time-series data processing algorithm. We implement the 2-layers GRU model which performed best.
- CallAtRumor [11] proposed a Bi-LSTM model combined with attention mechanism to learning content features, and improved accuracy on rumor detection compared with simple RNN.
- AIM [20] used two attention-based models: content attention model used to calculate contextual features, dynamic attention model related to the posting time of each post. This method had a good performance in rumor detection task.
- HSA [12] proposed a hierarchical network based on Bi-LSTM and three-level attention models. This method extracted content feature based on word-level data and fused three kinds of hand-crafted features, and achieved good performance on rumor detection problem. HA-BLSTM is a variant of HSA, which removes social features.

For comprehensive evaluation, we take confusion matrix for validating the proposed ARC model and other compared methods. Especially, DOT dataset does not contain complete hand-crafted features. Thus, when utilize SVM-TS and HSA methods, we just consider content features.

5.3 Experimental Setup

During training process, the L2-regularization term, batch normalization and dropout operation are used to avoid over-fitting problem. All the parameter settings are listed in Table 2:

Table 2: Parameter Settings

Parameter		Value
Word embedding dimension		300
Fixed-length of Input V		200
Filter in CNN	number	4
	window size	(2,4,8,16)
	channels	64
The number of attention model		10

6 RESULT AND ANALYSIS

In this section, we extensively analysis detection performance against compared methods.

6.1 Overall performance

First, we evaluate the accuracy of overall ARC model. For sentence vector learning and residual framework building, we utilize unweighted average method and Embedded Gaussian attention, respectively. Experimental results of all systems are shown in Table 3, according to which we have the following observations:

- ARC outperforms all the compared methods on both datasets. Especially, ARC achieves a accuracy of 87.0% on TREE dataset and 80.7% on DOT data, which demonstrate the great adaptability of our model for applicable rumor detection. The performance of ARC indicates that our model can indeed effectively draw latent features from global level and local level.
- HSA underperforms ARC in two datasets, which demonstrates that fusion of multi-modal features may not be competitive with comprehensive learning of lexical properties, and validates our choice of fundamental contextual feature.
- GRU-2 does not show competitive performance compared to other neural networks, which suggests that content feature captured by RNN cannot benefit to rumor detection. This phenomenon can be attributed to that simple recurrent operation may has limitation on distance of components.
- DTR-Enquiry and SVM-TS underperform neural network models. They resort to combination of multiple hand-crafted features or clues. Hence, they are limited with feature quality and lack of dynamic temporal features. Besides, only 10.9% tweets in TREE dataset and 21.4% tweets in DOT dataset contain enquiry expressions; DOT dataset does not contain required hand-crafted feature. These are why the results of DTR-Enquiry and SVM-TS are not satisfactory.
- DOT dataset is much more noise than TREE dataset, which contains extra structure features. This difference between two datasets may explains the lower baseline accuracies on DOT dataset than on TREE dataset. The proposed ARC model can help overcome the noise by capturing higher-level content features more accurately.

6.2 Further Study

To further verify the effectiveness of our model, we evaluate several internal models. There are three groups of them:

6.2.1 Ablation Study. The first group is simplified variations of ARC by removing some components:

- EC-L : We remove the global feature branch and simply utilize one-dimensional CNN to learn local feature.
- ECC : We just remove the attention residual framework.
- ARC-G : We remove the local feature branch and serve the global feature as classified feature.

For performance evaluation, we assign ARC as compared methods, and confusion matrix computed by overall data as evaluation metrics.

Table 4 shows the performance of all simplified models, we have the following observations: (1) EC-L shows that one-dimensional CNN can exploit local feature effectively, which is significant for rumor detection. (2) Model with the attention residual network performs better, indicating that the fine-tuned attention model can effectively represent accurate feature by capturing long-range dependency. (3) Compared with TREE dataset, performance improvement of ARC-G is better on DOT dataset. This phenomenon indicates that the proposed model indeed effectively help overcome noise and focus on the content.

Table 3: Performance comparison on two datasets: TREE(top) and DOT(bottom).

Dataset	Method	Accuracy	Rumor			Non-Rumor		
			Precision	Recall	F-1	Precision	Recall	F-1
TREE	DTR-Enquiry	0.644	0.638	0.675	0.656	0.652	0.613	0.632
	SVM-TS	0.751	0.805	0.667	0.729	0.713	0.837	0.770
	GRU-2	0.784	0.870	0.670	0.757	0.730	0.899	0.805
	CallAtRumor	0.804	0.821	0.780	0.800	0.789	0.828	0.808
	AIM	0.796	0.746	0.846	0.794	**0.851**	0.754	0.799
	HSA	0.844	**0.948**	0.730	0.825	0.779	**0.960**	0.863
	ARC	**0.879**	0.907	**0.975**	**0.891**	0.844	0.884	**0.864**
DOT	DTR-Enquiry	0.636	0.678	0.740	0.691	0.615	0.508	0.557
	SVM-TS	0.593	0.514	0.644	0.571	0.681	0.556	0.612
	GRU-2	0.679	0.739	0.747	0.743	0.577	0.566	0.571
	CallAtRumor	0.707	**0.886**	0.716	0.792	0.404	0.677	0.506
	AIM	0.735	0.692	0.631	0.660	0.761	0.807	0.782
	HA-BLSTM	0.742	0.692	0.643	0.667	0.772	**0.809**	**0.790**
	ARC	**0.807**	0.807	**0.877**	**0.840**	**0.807**	0.712	0.757

Table 4: Performance of simplified models on rumor detection

Dataset	Model	Accuracy	Precision	Recall	F-1
TREE	EC-L	0.810	0.907	0.778	0.838
	ECC	0.830	0.926	0.794	0.855
	ARC-G	0.820	0.887	0.797	0.839
DOT	EC-L	0.764	0.795	0.824	0.809
	ECC	0.786	0.830	0.830	0.830
	ARC-G	0.771	0.829	0.811	0.820

6.2.2 *Variants of Encoding.* The second group is variants of ARC, which produced by the calculating methods of sentence vectors. They are ARC-UNW, ARC-SIF, ARC-IDF. The weighting parameter c of ARC-SIF is set as 10^{-5} and word frequency is computed by the common dataset enwiki (wikipedia, 3 billion tokens).

Table 5: Performance of variant encoding model on rumor detection

Dataset	Model	Accuracy	Precision	Recall	F-1
TREE	ARC-UNW	0.879	0.907	0.875	0.891
	ARC-SIF	0.817	0.860	0.796	0.827
	ARC-IDF	0.838	0.792	0.894	0.840
DOT	ARC-UNW	0.807	0.807	0.877	0.840
	ARC-SIF	0.757	0.859	0.744	0.798
	ARC-IDF	0.786	0.784	0.863	0.821

Table 5 shows the performance of variant encoding model, we observe that: (1) On two datasets, different methods influence performance to some extent. In summary, the unweighted average method overcome other methods, smooth inverse frequency is not satisfactory. (2) On the DOT dataset, ARC-UNW outperforms other two models, which demonstrates that unweighted average method has a good adaptability on noisy dataset.

6.2.3 *Choice of Attention Residual Framework.* The third group is different choice of attention residual framework:

- ARC-D: Adapt fine-tuned self-attention model.
- ARC-G: Adapt Gaussian attention model.
- ARC-EG: Adapt Embedded Gaussian model.

For performance evaluation, we assign ARC as compared method, and confusion matrix computed by overall data as evaluation metrics.

Table 6: Performance of variant attention residual framework on rumor detection

Dataset	Model	Accuracy	Precision	Recall	F-1
TREE	ARC-D	0.858	0.926	0.833	0.877
	ARC-G	0.849	0.833	0.883	0.857
	ARC-EG	0.879	0.907	0.875	0.891
DOT	ARC-D	0.786	0.864	0.809	0.835
	ARC-G	0.786	0.830	0.830	0.830
	ARC-EG	0.807	0.816	0.866	0.840

Table 6 shows performance of variant attention residual framework on rumor detection, we have the following observations: All kinds of fine-tuned attention model show good performance, the performance of different weighted function changes slightly. This phenomenon demonstrates that attention residual framework is able to capture long-range dependencies without regard to strategy choice; instead, it is more likely that the position encoding behavior is important.

7 EARLY RUMOR DETECTION

Identify rumors in the early stage of propagation is truly important. In this section, we examine the performance of our model on rumor early detection. We take different ratios of tweets as detection deadline, ranging from 20% to 100%. That means during testing process, all information of an event after the deadline are limited; meanwhile, the complete messages of an event is visible. For performance evaluation, we take 4 competitive methods: GRU-2, CallAtRumor, ARC, ARC-G, .

Figure 7 and Figure 8 show the accuracy of compared methods on TREE and DOT dataset, respectively. Experimental results show that our model is superior to other state-of-art methods on detecting rumors at early stage. In most instances, all methods improve performance with more data involved in training. However, we find that GRU-2 does have obvious advantage on very early stage, the reason is that GRU-2 is sensitive to noise data. In particular, the great performances of EC-L and ARC-G demonstrate that not local feature but global feature has a positive implication for detecting rumor.

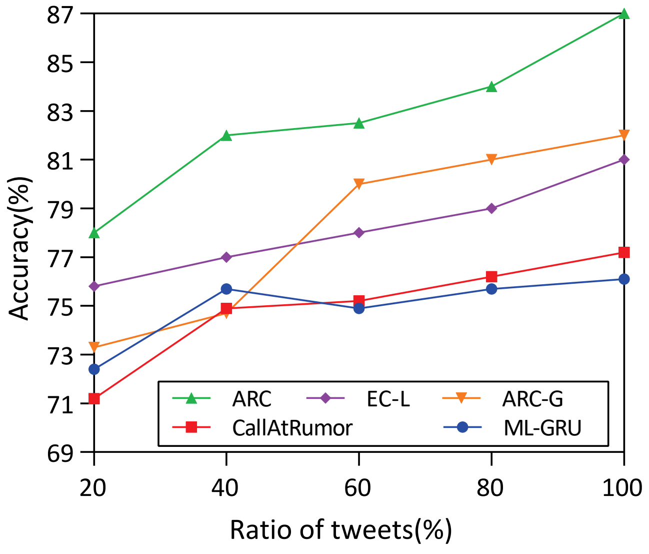

Figure 7: The performance of all methods on TREE.

8 CONCLUSIONS

In this paper, we propose an attention-residual network combined with CNN for rumor detection. In contrast to existing works relying on RNN to capture long-range dependency, we modeled an attention residual framework to relate different position of a sequence. Moreover, we also retained important spacial features. Finally, via a accurate contextual feature generated by the combination of local feature and global feature, our model achieved a good performance on two datasets. Extensive experiments showed that our proposed model (ARC) outperforms other state-of-art models in both rumor detection and early rumor detection. In future work, we will examine effect of the number of attention models for rumor detection.

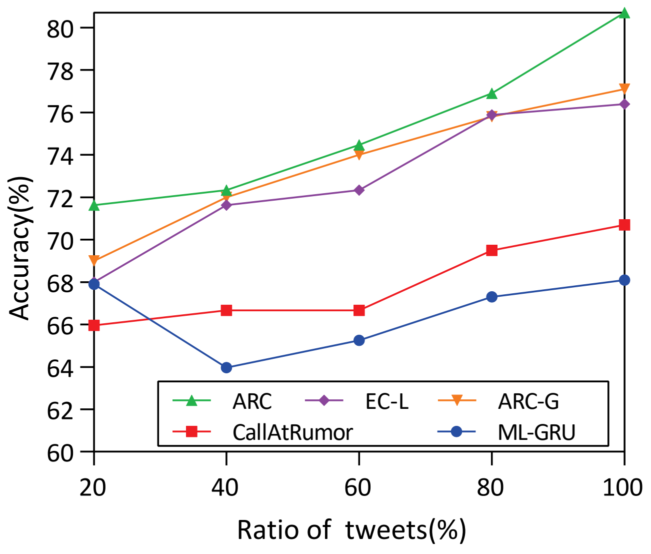

Figure 8: The performance of all methods on DOT.

ACKNOWLEDGMENTS

This work is supported by The National Key Research and Development Program of China (Grant No. 2017YFB0803001) and The National Natural Science Foundation of China (Grant No. 61572459).

REFERENCES

[1] Arkaitz Zubiaga, Ahmet Aker, Kalina Bontcheva, Maria Liakata, and Rob Procter. Detection and resolution of rumours in social media: A survey. *ACM Computing Surveys (CSUR)*, 51(2):32, 2018.
[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. 2017.
[3] Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. Information credibility on twitter. In *Proceedings of the 20th international conference on World wide web*, pages 675–684. ACM, 2011.
[4] Vahed Qazvinian, Emily Rosengren, Dragomir R Radev, and Qiaozhu Mei. Rumor has it: Identifying misinformation in microblogs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1589–1599. Association for Computational Linguistics, 2011.
[5] Alex Hai Wang. Don't follow me: Spam detection in twitter. In *2010 international conference on security and cryptography (SECRYPT)*, pages 1–10. IEEE, 2010.
[6] Fan Yang, Yang Liu, Xiaohui Yu, and Min Yang. Automatic detection of rumor on sina weibo. In *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*, page 13. ACM, 2012.
[7] Zhe Zhao, Paul Resnick, and Qiaozhu 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. International World Wide Web Conferences Steering Committee, 2015.
[8] Guoyong Cai, Hao Wu, and Rui Lv. Rumors detection in chinese via crowd responses. In *Proceedings of the 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 912–917. IEEE Press, 2014.
[9] Jing Ma, Wei Gao, Zhongyu Wei, Yueming Lu, and Kam-Fai 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. ACM, 2015.
[10] Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J Jansen, Kam-Fai Wong, and Meeyoung Cha. Detecting rumors from microblogs with recurrent neural networks. In *IJCAI*, pages 3818–3824, 2016.
[11] Tong Chen, Xue Li, Hongzhi Yin, and Jun Zhang. Call attention to rumors: Deep attention based recurrent neural networks for early rumor detection. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 40–52. Springer, 2018.
[12] Han Guo, Juan Cao, Yazi Zhang, Junbo Guo, and Jintao Li. Rumor detection with hierarchical social attention network. In *Proceedings of the 27th ACM International*

Conference on Information and Knowledge Management, pages 943–951. ACM, 2018.

[13] Xiaoxue Li, Yanan Cao, Yanmin Shang, Yanbing Liu, Jianlong Tan, and Li Guo. Inferring user profiles in online social networks based on convolutional neural network. In *International Conference on Knowledge Science, Engineering and Management*, pages 274–286. Springer, 2017.

[14] Abdallah Yousif, Zhendong Niu, and Ally S Nyamawe. Citation classification using multitask convolutional neural network model. In *International Conference on Knowledge Science, Engineering and Management*, pages 232–243. Springer, 2018.

[15] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[16] Tao Chen, Ruifeng Xu, Yulan He, and Xuan Wang. Improving sentiment analysis via sentence type classification using bilstm-crf and cnn. *Expert Systems with Applications*, 72:221–230, 2017.

[17] Hao Peng, Jianxin Li, Yu He, Yaopeng Liu, Mengjiao Bao, Lihong Wang, Yangqiu Song, and Qiang Yang. Large-scale hierarchical text classification with recursively regularized deep graph-cnn. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, pages 1063–1072. International World Wide Web Conferences Steering Committee, 2018.

[18] Yi-Chin Chen, Zhao-Yang Liu, and Hung-Yu Kao. Ikm at semeval-2017 task 8: Convolutional neural networks for stance detection and rumor verification. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 465–469, 2017.

[19] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[20] Qiang Liu, Feng Yu, Shu Wu, and Liang Wang. Mining significant microblogs for misinformation identification: An attention-based approach. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 9(5):50, 2018.

[21] Yoon Kim, Carl Denton, Luong Hoang, and Alexander M. Rush. Structured attention networks. 2016.

[22] Ankur P. Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. 2016.

[23] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018.

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[25] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3156–3164, 2017.

[26] Dumitru Erhan, Yoshua Bengio, Aaron C. Courville, Pierre Antoine Manzagol, and Samy Bengio. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, 11(3):625–660, 2010.

[27] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[28] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[29] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. A simple but tough-to-beat baseline for sentence embeddings. 2016.

[30] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1243–1252. JMLR. org, 2017.

[31] Michael Mathioudakis and Nick Koudas. Twittermonitor: trend detection over the twitter stream. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pages 1155–1158. ACM, 2010.