A Temporal Attentional Model for Rumor Stance Classification

Amir Pouran Ben Veyseh[*]
Department of Computer and Information Science
University of Oregon
Eugene, OR 97403, USA
apouranb@cs.uoregon.edu

Javid Ebrahimi
Department of Computer and Information Science
University of Oregon
Eugene, OR 97403, USA
javid@cs.uoregon.edu

Dejing Dou
Department of Computer and Information Science
University of Oregon
Eugene, OR 97403, USA
dou@cs.uoregon.edu

Daniel Lowd
Department of Computer and Information Science
University of Oregon
Eugene, OR 97403, USA
lowd@cs.uoregon.edu

ABSTRACT

Rumor stance classification is the task of determining the stance towards a rumor in text. This is the first step in effective rumor tracking on social media which is an increasingly important task. In this work, we analyze Twitter users' stance toward a rumorous tweet, in which users could support, deny, query, or comment upon the rumor. We propose a deep attentional CNN-LSTM approach, which takes the sequence of tweets in a thread of conversation as the input. We use neighboring tweets in the timeline as context vectors to capture the temporal dynamism in users' stance evolution. In addition, we use extra features such as friendship, to leverage useful relational features that are readily available in social media. Our model achieves the state-of-the-art results on rumor stance classification on a recent SemEval dataset, improving accuracy and F_1 score by 3.6% and 4.2% respectively.

KEYWORDS

Rumor Stance Classification, Temporal Attention, LSTM, Twitter

1 INTRODUCTION

One of the recent forms of deception on the web is propagating fake news on social media. Given the abundant unfiltered outlets that purport to broadcast news, it is necessary to separate facts from falsehoods in online sources, thereby assessing credibility. One way to improve estimating the veracity of rumors on social media is to study people's conversations on breaking news/rumors. When a rumor is broadcasted, people may deny or support the claim based on their knowledge or even political/ideological affiliation. Additionally, they may query the source about the details or the specifics of the rumor. Rumor stance classification aims to classify responses to a rumor by type (i.e., deny, support, query, comment). In this work, we study stance-labeled conversations on Twitter. Figure 1 shows a labeled tweet conversation initiated by a rumorous tweet.

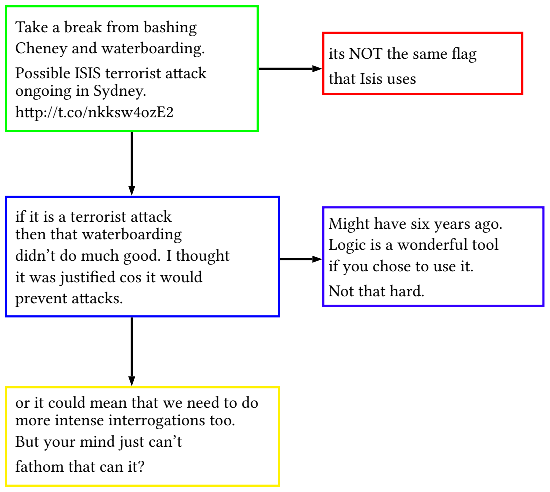
Take a break from bashing Cheney and waterboarding. Possible ISIS terrorist attack ongoing in Sydney. http://t.co/nkksw4ozE2
its NOT the same flag that Isis uses
if it is a terrorist attack then that waterboarding didn't do much good. I thought it was justified cos it would prevent attacks.
Might have six years ago. Logic is a wonderful tool if you chose to use it. Not that hard.
or it could mean that we need to do more intense interrogations too. But your mind just can't fathom that can it?

Figure 1: A sample conversation on a rumorous tweet. Green, red, yellow and blue boxes represent support, deny, query and comment labels respectively.

Rumors are resolved as true or false after a period of time; naturally, users' collective stance towards tweets evolves. For example, we observed that early in the timeline there are more *query* tweets; but as time goes by, and more knowledge about the event is known, the number of *query* tweets decreases. Similarly, the number of *deny* tweets increase over time, which can be due to the rumor being debunked. Following this observation, to learn a better tweet representation, we take the tweet and its neighboring tweets in the timeline as context, and use attention functions to learn the weight for each tweet.

In the past few years, there has been a growing interest in studying rumors in social media [3, 5, 11]. The tree structure of tweets in

[*]The work was done when the author was a MS student at the University of Tehran from 2016 to 2017

CIKM'17, November 6–10, 2017, Singapore.
© 2017 ACM. ISBN 978-1-4503-4918-5/17/11...$15.00
DOI: https://doi.org/10.1145/3132847.3133116

a thread of tweets has been found to improve the performance of an independent classifier [11]. Instead of the whole tree, an LSTM based model [3] was used to classify a sequence (branch) of replies to an orginating tweet. Moreover, the temporal process of tweet generation was studied by posing it as a Hawkes process [5].

We generalize the sequence labeling approach of [3] to incorporate temporal information. Rather than treating the tweet generation process separate from the text [5], we propose a unified model that exploits the textual-temporal aspects of the data. In addition, we incorporate relational features such as friendship for further improvements. Specifically, we introduce a variable to discriminate weight learning for tweets where the user is in a relationship with a person in the thread. Our experiments on a recent SemEval rumor stance classification challenge show that our model is able to achieve state-of-the-art performance with significant improvements in accuracy and F-score over previous best published results.

Our contributions in this paper are:

- Introducing a novel temporal attention mechanism.
- Incorporating relational features in a thread of tweets.
- Achieving state-of-the-art performance on a rumor stance classification dataset.

2 RELATED WORK

The first work that studied machine learning application to rumor stance classification was Qazvinian et al. [8]. Rumor stance classification for Twitter conversations has also been studied in the RumourEval shared task at SemEval 2017 [1], where the winning team used a sequential classifier based on LSTM [3]. Another work that has utilized recurrent neural network in this context is [6]. However, they have used RNN for rumor detection which is different from our purpose in this paper. To incorporate the structure of the conversations, Zubiaga et al. [11] used conditional random fields (CRF) in linear-chain CRF and tree CRF settings. To incorporate temporal ordering of tweets, Gaussian Process [4] and Hawkes process [5] approaches have been investigated. In [7] the authors have used the stance, reliability and trend of different sources on the web to classify and explain credibility of a claim. However they worked on different context, our findings on stance evolution and classification towards rumors in social networks could be incoporated in such models to obtain better results. Ebrahimi et al. in [2] have used a hinge-loss Markov random fields model to classify the stance of tweets. Unlike their work, our model is able to detect the stance of the tweet towards rumorous tweets instead of fixed targets and also utilizes stance evolution. For a complete survey of the previous work, we refer the reader to [10]. In comparison with previous works, our approach is able to use conversation structure combined with attention based temporal information and relational features.

3 MODEL

As previous works have shown [3, 11], the tree structure of replying tweets is helpful for this task. We use the sequence of tweets starting from the rumorous tweet followed by the reply chain, which are fed to a stacked-LSTM of two layers. We use two extra sources of information from training data: temporal dynamism and relational features.

3.1 Temporal Attention

When a rumorous tweet is published, users' stance toward that rumor would change over the time. For example, most of the *query* tweets are posted earlier in the timeline, whereas it takes longer for *deny* tweets to emerge. Figure 3 shows this trend in the training set. So considering some of previous and following tweets helps. In our model, we use a context window of six tweets; three of them are those tweets published right before the current tweet, and the other three tweets are published right after it.

To represent these tweets in vector space, we average the word embeddings in the tweets as follows:

$$\hat{e}_i = \frac{1}{T}\Sigma_{j=1}^{T} e_j \tag{1}$$

In equation 1, T represents the max tweet length and e_j represents the word embedding for word j. $\hat{e}_i$ is the vector representation for tweet i.

Since all of the tweets don't have the same importance, we use an attention mechanism to construct a weighted average of these tweets. Similar to [9] we use a simplified version of attention. Our learnable attention function only depends on the vector representation of the tweet as follows:

$$\beta_i = a(\hat{e}_i), \;\; \alpha_i = \frac{exp(\beta_i)}{\Sigma_{k=1}^{W} exp(\beta_k)}, \tag{2}$$

where a is a learnable attention function (in our model we have used one layer feed-forward network with a sigmoid activation), and W represents the context window size. While this attention mechanism takes only textual information as input, since it performs on the information expressed before and after the current tweet and disregards other structural features we call it temporal attention. We leave a more complicated attention mechanism that works on other temporal information for our future work.

The output of the attention is multiplied with the vector representation of each tweet. Then, we sum these seven weighted tweets in order to have one vector representing the contextual information for the tweet.

$$c_i = \Sigma_{j=1}^{W} \alpha_j \hat{e}_j \tag{3}$$

In our experiments, considering the main tweet, along with the neighboring tweets, in the context window proved to be important. We think this choice helps the network extract more relevant contextual features from the neighboring tweets.

A CNN with one layer of 1-dimensional convolution and one max pooling layer is used to extract features from this final representation.

3.2 Relation Vector

In addition to temporal information, there are other features in a thread of tweets that could be utilized to capture contextual evidence. For each tweet, we add a binary variable which determines whether the tweet is in a *relation*. These relations include friendship[1], user mentions, and user's own previous posting. For example, a tweet posted by a user previously mentioned in another

[1]We download users' friend list through Twitter API, who are the people that a person follows.

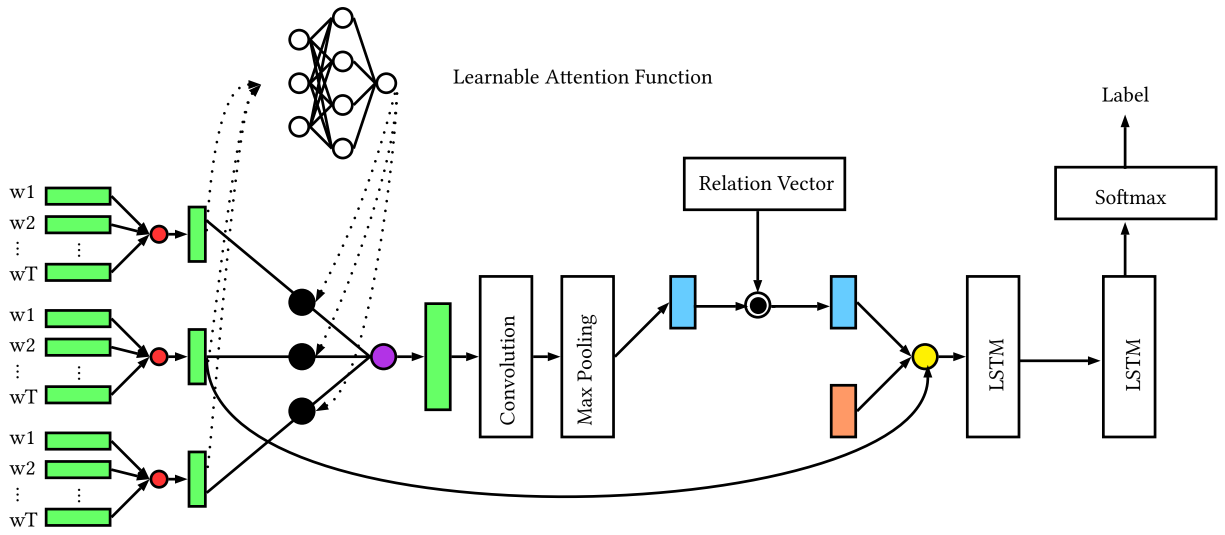

Figure 2: The green rectangles represent word embeddings. The red circles perform mean-pooling. The black circles are the attention functions, which are followed by a weighted sum operation. The output of the CNN and the relation vector are multiplied element-wise. The orange rectangles denote extra features, which are concatenated to the output of the CNN and the tweet vector.

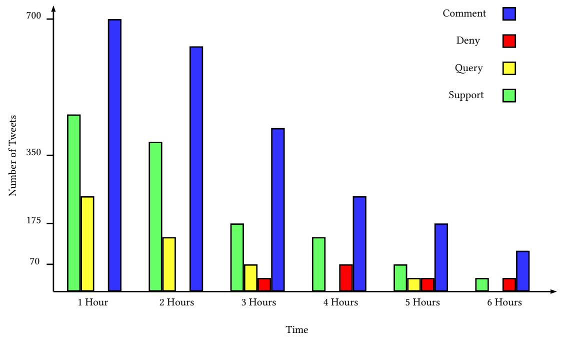

Figure 3: Distribution of Labels as time passes. The horizontal axis displays how much time has passed from the originating rumorous tweet. The vertical axis displays the number of tweets in each class.

Model	Accuracy	$F_1 Macro$
LSTM	78.1	42.1
Turing [3]	78.4	43.4
Tree CRF [11]	NA	44
Ours	**82.0**	**48.2**

Table 1: Performance results

tweet is in a relation. A tweet posted by a user whose friend has also posted on the thread is another example. When such relationships exist, we perform an element-wise multiplication between the context vector and a learnable weight vector. Relations reflect a history of tweeting behavior; the intuition is to incorporate a bias in the tweet vector when such a relationship exists. Note that all these relations are aggregated and only one weight vector is used. The multiplicative operator had the best results in our experiments.

3.3 Overall Model

Our experiments show there are extra features, which are mostly independent of the tweet content that could enhance the model performance. These features include:

- whether the author has retweeted the rumorous tweet;
- whether the tweet contains media or a URL;
- average number of retweets, replies, and favorite count;
- whether the tweet is replying to the rumorous tweet or to the other tweets in the thread;
- average number of favorite count, follower counts, and status counts for the author;
- and whether the tweet contains a hashtag that exists in other tweets replying to the rumorous tweet.

We concatenate these features with those found by the product of relation vector and the CNN output. In addition, the main tweet itself is concatenated with them before being fed to the LSTM. The final label of the tweet is the output of Softmax layer. See Figure 2.

4 EXPERIMENTS

We use the recent SemEval 2017 Task 8.A dataset. This dataset consists of 5568 labeled tweets. One of the most challenging issues in this dataset is that the data is extremely skewed toward the *comment* and it is difficult for the model to correctly classify minority classes; i.e. *deny* and *query*. This dataset contains ten different topics; each of which has several rumorous originating tweets. The tweets are organized in a tree structure based on their reply chain

Model	Sydney Siege		Charlie Hebdo		Ferguson		Ottawa	
	Accuracy	$F_1 Macro$	Accuracy	$F_1 Macro$	Accuracy	$F_1 Macro$	Accuracy	$F_1 Macro$
HP [5]	68.59	32.49	72.93	32.56	68.44	25.99	67.77	32.29
Ours	**73.15**	**40.98**	**77.09**	**40.81**	**72.14**	**36.17**	**74.93**	**43.08**

Table 2: Performance results in leave-one-rumor-out setting

Component	Accuracy	$F_1 Macro$
All	82.0	48.2
CNN-	79.6	46.8
Relation Vector-	80.9	47.3
Temporal attention-	79.8	46.4
Penalizing Minority Misclassification-	81.3	45.5

Table 3: Contribution of model components. Each row shows the result after removing that component.

and the originating rumorous tweet is the root of the tree. In total, the dataset contains 325 rumors. Each rumorous conversation contains 15 tweets in average.

In our experiments we use each path from the root tweet to the leaf as a sequence of tweets. To find the context tweets for each tweet we only use those tweets that are in the tree that the current tweet belongs to. To deal with the minority class problem, we use Keras[2]'s weighted classification to penalize misclassification for the *deny* class. Moreover, we use larger context window (5 tweets before and after current tweet) for this class. Our experiments show that such adaptation could improve the model's F_1 score by 2.7%. For word embeddings, we start with Google News word2vec and fine-tune them using a corpus of 100,000 tweets, which we downloaded using the Twitter API based on a handful of hashtags in the training dataset. To tune the hyper-parameters, we conduct a grid search over (1) attention function hidden size (2) number of CNN filters and filters size (3) number of hidden units in the LSTM (4) and dropout probability between the two LSTM layers, on the validation set.

We compare our results with Kochkina et al. [3] (the winner of SemEval 2017 Task 8.A) and Zubiaga et al. [11]. As another baseline, we use a simple LSTM-based classifier which takes the sequence of words of the tweet as its input, which achieves competitive results. Accuracy and F_1 have been used as performance metrics. The results have been shown in Table 1.

We also compare our model with Lukasik et al. [5]. They try to use temporal information in training data using Hawkes process modeling of tweet generation. In contrast, our model uses an attention mechanism to extract features dependent on temporal relationship among neighboring tweets. Table 2 shows that our model is able to achieve better results in terms of accuracy and F_1 score. Since Lukasik et al. [5] use only four topics in training data and perform a *leave-one-rumor-out* evaluation, we conduct our experiments in this setting too.

In order to study the contribution of each component of our model, we remove each of them individually and report the results on the test set. In Table 3, each row shows the performance of the model when the corresponding component has been removed. Among all components of the model, CNN and the temporal attentional mechanism have the largest impact on the model performance. Also, using the penalization technique for the minority classes substantially improves the F_1 score.

5 CONCLUSION & FUTURE WORK

In this paper we introduced a temporal attention mechanism to utilize the collective stance evolution towards rumorous tweets. We also employed tree structure of tweet conversations, as well as relational features in Twitter. Our model improves the state-of-the-art results significantly in terms of accuracy and F_1 score.

We will investigate our temporal attention mechanism for tasks beyond stance classification to general discourse analysis in Twitter conversations. Also, in order to incorporate the full conversation tree in our modeling, using a tree based deep learning model like recursive neural nets could be useful.

REFERENCES

[1] Leon Derczynski, Kalina Bontcheva, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Arkaitz Zubiaga. 2017. SemEval-2017 Task 8: RumourEval: Determining rumour veracity and support for rumours. (2017).

[2] Javid Ebrahimi, Dejing Dou, and Daniel Lowd. 2016. Weakly Supervised Tweet Stance Classification by Relational Bootstrapping. In *Proceedings of EMNLP 2016*. Austin, Texas, 1012–1017.

[3] Elena Kochkina, Maria Liakata, and Isabelle Augenstein. 2017. Turing at SemEval-2017 Task 8: Sequential Approach to Rumour Stance Classification with Branch-LSTM. (2017). http://arxiv.org/abs/1704.07221

[4] Michal Lukasik, Kalina Bontcheva, Trevor Cohn, Arkaitz Zubiaga, Maria Liakata, and Rob Procter. 2016. Using Gaussian processes for rumour stance classification in social media. *arXiv preprint arXiv:1609.01962* (2016).

[5] Michal Lukasik, P. K. Srijith, Duy Vu, Kalina Bontcheva, Arkaitz Zubiaga, and Trevor Cohn. 2016. Hawkes Processes for Continuous Time Sequence Classification: an Application to Rumour Stance Classification in Twitter. In *Proceedings of ACL*.

[6] Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J. Jansen, Kam-Fai Wong, and Meeyoung Cha. 2016. Detecting Rumors from Microblogs with Recurrent Neural Networks. In *Proceedings of IJCAI 2016, New York, NY, USA, 9-15 July 2016*. 3818–3824. http://www.ijcai.org/Abstract/16/537

[7] Kashyap Popat, Subhabrata Mukherjee, Jannik Strötgen, and Gerhard Weikum. 2017. Where the Truth Lies: Explaining the Credibility of Emerging Claims on the Web and Social Media. In *Proceedings of WWW 2017, Perth, Australia.*

[8] Vahed Qazvinian, Emily Rosengren, Dragomir R Radev, and Qiaozhu Mei. 2011. Rumor has it: Identifying misinformation in microblogs. In *Proceedings of EMNLP*. 1589–1599.

[9] Colin Raffel and Daniel P. W. Ellis. 2015. Feed-Forward Networks with Attention Can Solve Some Long-Term Memory Problems. (2015). http://arxiv.org/abs/1512.08756

[10] Arkaitz Zubiaga, Ahmet Aker, Kalina Bontcheva, Maria Liakata, and Rob Procter. 2017. Detection and Resolution of Rumours in Social Media: A Survey. (2017).

[11] Arkaitz Zubiaga, Elena Kochkina, Maria Liakata, Rob Procter, and Michal Lukasik. 2016. Stance Classification in Rumours as a Sequential Task Exploiting the Tree Structure of Social Media Conversations. In *Proceedings of COLING*.

[2] https://keras.io/