eventAI at SemEval-2019 Task 7: Rumor Detection on Social Media by Exploiting Content, User Credibility and Propagation Information

Quanzhi Li, Qiong Zhang, Luo Si
Alibaba Group, US
Bellevue, WA 98004, USA
{quanzhi.li, qz.zhang, luo.si}@alibaba-inc.com

Abstract

This paper describes our system for SemEval 2019 RumorEval: Determining rumor veracity and support for rumors (SemEval 2019 Task 7). This track has two tasks: Task A is to determine a user's stance towards the source rumor, and Task B is to detect the veracity of the rumor: true, false or unverified. For stance classification, a neural network model with language features is utilized. For rumor verification, our approach exploits information from different dimensions: rumor content, source credibility, user credibility, user stance, event propagation path, etc. We use an ensemble approach in both tasks, which includes neural network models as well as the traditional classification algorithms. Our system is ranked 1st place in the rumor verification task by both the macro F1 measure and the RMSE measure.

1 Introduction

Social media platforms, such as Twitter, Reddit and Facebook, do not always poses authentic information. Rumors sometimes may spread quickly over these platforms (Castillo et al., 2011; Derczynski and Bontcheva, 2014; Qazvinian et al., 2011). A rumor may be defined as a statement whose truth value is unverified or deliberately false (Qazvinian et al., 2011). Rumors usually spread fear or even euphoria, and they may confuse people and cause them to make wrong decisions. Therefore, rumor detection has gained great interest recently. In this paper, we describe the approaches we used in SemEval 2019 RumourEval: Determining rumor veracity and support for rumors (SemEval 2019 Task 7). This task has two subtasks: Task A - user stance classification and Task B - rumor verification.

Stance classification is to determine the attitude of the author of a post message towards a target (Mohammad et al., 2016). In task A, we focus on stance classification of messages towards the truthfulness of rumors in Twitter or Reddit conversations. Each conversation is defined by a source post that initiates the conversation, and a set of replies to it, which form a conversation thread. The goal is to classify each post into one of the four categories: supporting, denying, querying or commenting (SDQC). For this task, we use an ensemble approach, which combines the prediction results from both the traditional learning models, such as SVM, and a neural network model, using the language features extracted from the message text. Task B predicts the veracity of a rumor: true, false, or unverified (i.e., its veracity cannot be verified based on the given information). Each rumor consists of a source post that makes a claim, and a set of replies, directly or indirectly towards the source post. We also employed an ensemble approach on this task, which uses multiple models together to do the veracity prediction.

For more details about these two tasks, please check the task description paper from the task organizers (Gorrell et al., 2019).

2 Related Studies

Rumor detection on social media has been a popular research topic in recent years. The early exploration of this issue started from two special case studies on rumor propagation during natural disasters like earthquakes and hurricanes (Gupta et. al., 2013; Mendoza et al., 2010). Many existing algorithms (Liu et al., 2015; Wu et. al., 2015; Yang et. al., 2012) for debunking rumors followed the work of Castillo et al. (2011). They studied information credibility and proposed a set of features that are able to retrospectively predict if an event is credible.

Stance classification is also an active research area that has been studied in previous work (Lukasik et al., 2016; Zubiaga et al., 2016; Kochkina et al., 2017). A time sequence classification technique has been proposed for

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 855–859
Minneapolis, Minnesota, USA, June 6–7, 2019.

detecting the stance against a rumor (Lukasik et al., 2016). Zubiaga et al. (2016) used sequence of label transitions in tree-structured conversations for classifying stance.

Several studies have applied neural networks on the verification of rumors (Ma et al., 2016; Kochkina et al., 2017; Ma et al., 2017); They mainly focus on analyzing the information propagation structure, and have not utilized much information on user credibility. User stance plays an important role in rumor detection. Recent works have employed multi-task learning approaches to jointly learn stance detection and veracity prediction, in order to improve classification accuracy by utilizing the interdependence between them (Ma et al.,; 2018; Kochkina et al., 2018).

3 System Description

We first describe the data set, the word embedding, our message representation method, and then our systems for the two tasks.

Data set quality: Regarding the annotation of the data set, as the task description already pointed out: the overall inter-annotator agreement rate of 63.7% showed the task to be challenging, and easier for source tweets (81.1%) than for replying tweets (62.2%). This means that there are many conflicting or inconsistent labels. This will confuse the learning based models, and make the model and prediction result unstable. When we analyzed the training data set, we found many such examples. To make the labels more consistent, we run an analysis to find the posts that are basically the same or highly similar, but their labels are different. We then mark these posts, and use the same label, the one labeled on the majority of these posts, on them during training. The similarity between two posts is calculated by cosine measure, and the post/message embedding is used in the calculation. The similarity threshold for being considered as similar posts is empirically set as 0.75.

Word embeddings: A popular word embedding data set used by many previous studies is one that is created from Google news articles using word2vec (Mikolov et al., 2013). Because the data in this task are social media messages, mainly tweets, we think a embedding model built specifically from tweets will be more appropriate. Therefore, we used tweets collected from Twitter to train a word embedding model. Only English tweets are used, and about 200 million tweets are used to build the embedding model. Totally, 3 billion words are processed, and word embeddings are generated for 3.5 million unique terms using the word2vec model (Mikolov et al., 2013) and data from (Li et al, 2017). In this task, although some messages are from Reddit, they are similar to tweets, in terms of language style and message structure, since both are social media messages.

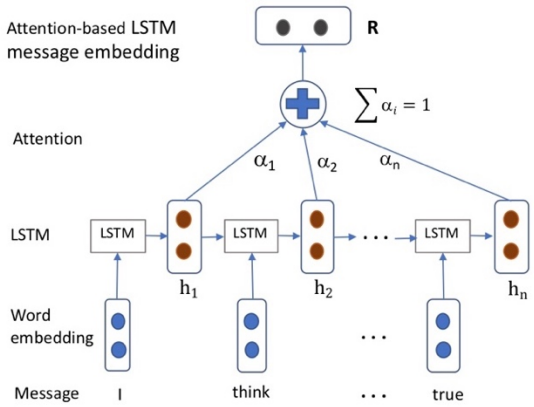

Figure 1. Message embedding based on the attention-based LSTM

Message representation: A tweet (or Reddit message) is usually very shot, consisting of only one sentence. In our models where a whole post is used, such as the stance detection, a message embedding is used as the message representation. We generate the message representation through an attention-based LSTM network for messages that have only one sentence. A post is first preprocessed, such as removing URLs, before it is fed into the LSTM network. Figure 1 shows the network structure for generating the message embedding.

3.1 Stance Detection

Similar to (Kochkina et al., 2017), we use a set of features that include the word embeddings and features generated from the message. They are listed below:

- **Message role:** it is the source message or a reply.
- **Message embedding**: this the message representation presented in Figure 1.
- **Presence of link**: has at least one link or not
- **Link type**: the types of the links. Types include image, video, and article.
- **Relation to source message**: whether this is a reply to the source message
- **Stance of parent message**: if this is a reply message, then the stance of the message it

replies to. In other words, we also check the stance of a message's parent in the propagation path.

- **Similarity with the source message**: Measured by cosine similarity using message embedding.
- **Punctuations**: we check if it has a question mark or exclamation mark.
- **Content length**: the message length after removing links and mentions
- **Mentions:** if the mentions are special accounts, e.g. @cnn, @theonion
- **Hashtags:** if it contains some special hashtags, e.g. #fakenews, #lying

These features are used by the following classification modules: A neural network model. The message embedding and other features are concatenated together and fed into the neural model, which consists of two fully connected layers and a softmax layer for the final label output; Models based on SVM, Random Forest and Logistic Regression, respectively; A rule based model.

The rule-based model handles some special case. Two examples are: 1. For source tweet stance: by default, label the source message as *support*, except A. if it is a Reddit message and it has "debunk this", then label it as *query*. B. if the source message has a question mark, and the sentence has the pattern of asking a question, label it as *query*. For example, the Yes/No questions: "did, do, does, have, has, am, is, are, can, could, may, would, will", and the WH-questions: "what, why, how, when, where", and the as well as their negations. 2. If the message is very short, and mainly contains a couple of keywords or hashtags expressing a very strong opinion, e.g. *#fakenews* or *"not true"*, then a corresponding label is assigned to it.

3.2 Rumor Verification

Our approach for rumor verification utilizes rumor information from several dimensions: text content, user credibility information, rumor propagation path, user stance, etc. Text content is utilized by almost all the previous studies on rumor verification. According to the deception style theory, the content style of deceptive information that aims to deceive readers should be somewhat different from that of the truth, e.g., using exaggerated expressions. An early study from (Castillo et al., 2011) uses many text features in their model. These features and other additional text features are also used in other studies (Liu et al., 2015; Enayet and El-Beltagy, 2017; Ma et al., 2017). Many previous studies have shown that user credibility information is very important in rumor verification (Castillo et al., 2011; Yang et al., 2012; Gupta et al., 2012; Liu et al., 2015; Li et al., 2016; Liu and Wu, 2018). Based on 421 false rumors and 1.47 million related tweets, Li et al. (2016) study various semantic aspects of false rumors, and analyze their spread and user characteristics. Their study shows that user credibility information is a good indicator for judging the credibility of a rumor story. To verify a rumor, we analyze the information from several dimensions:

- **Source content analysis**: whether the source message has links pointing to an article, video or image; length of the source message after removing URLs and mentions; number of named entities, verbs and nouns in the source message; whether the source message contains time expression.
- **Source account credibility:** the following information are considered: is a news agent account; profile has link pointing to top domains; account type: individual or organization (company, government agent, organization); profile has location information; profile has description; profile has image; profile has profession information; is verified user; number of followers, number of posts authors, account age, etc. How they are generated is similar to (Liu et al, 2015).
- **Source account credibility score**: calculated from the information listed in the *Source account credibility* category, and normalized to 0 to 1 (Liu et al, 2015).
- **Reply account credibility:** the profile information to check are similar to the ones in the Source account credibility category.
- **Reply account credibility score:** calculated from the information in the *Reply account credibility category*, and normalized to 0 to 1.
- **Stance of the source message:** get from Task A
- **Stance of replies**: percentage of each stance type; the overall stance score for each stance class, calculated by integrating each reply's account credibility score with its stance.
- **Rumor topic domain**: the topic area of the rumor, e.g. politics, business, science, etc. (Liu et al., 2015; Li et al., 2016).

These data are fed into different models for veracity prediction.

Propagation path analysis: Rumors spread through social media in the form of shares and re-shares of the source post and shared posts, resulting in a diffusion cascade or tree. Each source message has many replies, and they are either direct replies, or replies to other messages in the conversion thread. Take the rumors on Twitter as an example, the training data set contains 327 tweet rumors, and these rumors have 4017 branches and 5570 tweets, which means we have quite a lot replies that are not toward directly to the source message. The structure of the conversion thread is important for understanding the real stance of the user of a reply. For example, given a message "This is fake" and a reply to this message " I totally agree", if we do not know the reply is toward to the first message, then we will give a wrong label "support" to this reply. But actually this reply is denying the rumor claim. This is very important in rumor verification.

Models: We also use the ensemble approach in this task, where multiple models are used to predict rumor veracity, and stacking is employed for the final decision. Similar to the stance detection, we have following classification modules: A neural network model. All the features described above are concatenated together and fed into the neural model, which consists of two fully connected layers and a softmax layer for the final veracity prediction; Three models based on SVM, Random Forest and Logistic Regression, respectively; and a post-processing module that is described below.

Post-processing module

In this component, we consider some special characteristics of rumors and the data sets. We built some simple modules in this components, which may change the prediction results in previous steps. Some of them are described below:

Topics with multiple rumors: Some topics have multiple rumors, and in most cases, the rumors from the same topic have the same veracity, i.e. they are all true, false or unverified. To utilize this characteristic, for a given topic, after each rumor is processed and a prediction is generated, we use this post-process to re-evaluate their veracities, to see if any of them needs to be changed. The rumors from the same topic may or may not talk about the same claim, although in many cases, they are. We calculate the cosine similarity between two source posts, if it is greater than the threshold (set as 0.65 empirically), then the two rumors are considered as basically talking about the same claim, and their veracities are set as the same value. In similarity calculation, a source message is represented by its embedding, already described in previous section. The final veracity is chosen based on the distribution of the veracity values in these rumors, and their corresponding confidence scores.

Rumors originated from special accounts: due to the limited size of training data and annotation quality, some patterns or knowledge may not be caught by the trained models. But when we analyzed the data set, we found some patterns or signals that can provide very strong evidence on rumor veracity, especially for the false rumors. For example, TheOnion is a website usually publishing satirical news and opinions. A source message from this account usually is a false rumor. We check the source post and also the replies in the conversation thread, to see if there is evidence that the claim is from certain special accounts. Another example: if a rumor is from a very credible source, such as news agency NPR or a government agency, then it is very likely its veracity is true.

Rumors debunked or endorsed by special accounts: Similar to the last point, we also check the replies of a source messages, to see if some special accounts have expressed very strong opinion on that claim. For example, if a response is from Snopes.com, a rumor verification website, and it says "#fakenews", or its response is cited by a post in the conversation thread, we can confidently classify this rumor as false. The account information are obtained by analyzing the training data.

Rumor Veracity	Precision	Recall	F measure
True	0.733	0.710	0.721
False	0.828	0.600	0.696
Unverifies	0.227	0.500	0.313
Average	0.596	0.603	0.577

Table 1. Rumer verification result

4 Experiments and Results

The evaluation metric of Task A is the average macro F measure of the four stance categories. Task B uses two evaluation metrics: the average macro F measure of the three veracity types, and also RMSE. Regarding the model training, for the neural network model, the stochastic gradient descent, shuffled mini-batch, AdaDelta update, back-propagation and dropout are used. The word

embeddings were fine-tuned during the training process.

For the stance detection task, the result of our system is 0.578 , based on the macro F measure of SDQC. Table 1 shows the rumor detection result of our system. It shows that the unverified category got a very low precision, 0.227, and consequently, its F value is also pretty low, which is 0.313. And this value drags down the average F value of the three categories to 0.577.

References

C. Castillo, M. Mendoza, and B. Poblete. Information credibility on twitter. In Proc. WWW 2011

Ju-han Chuang and Shukai Hsieh. 2015. Stance classification on ptt comments. In Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation.

Genevieve Gorrell, Kalina Bontcheva, Leon Derczynski, Elena Kochkina, Maria Liakata, and Arkaitz Zubiaga, 2019, RumourEval 2019: Determining Rumour Veracity and Support for Rumours. SemEval 2019

Gupta, H. Lamba, P. Kumaraguru, and A. Joshi. Faking sandy: characterizing and identifying fake images on twitter during hurricane sandy. WWW 2013

Elena Kochkina, Maria Liakata, Isabelle Augenstein, 2017, Turing at SemEval-2017 Task 8: Sequential Approach to Rumour Stance Classification with Branch-LSTM, SemEval 2017

Quanzhi Li, Sameena Shah, Xiaomo Liu, Armineh Nourbakhsh, Rui Fang, TweetSift: Tweet Topic Classification Based on Entity Knowledge Base and Topic Enhanced Word Embedding, CIKM 2016

Quanzhi Li, Xiaomo Liu, Rui Fang, Armineh Nourbakhsh, Sameena Shah, User Behaviors in Newsworthy Rumors: A Case Study of Twitter. The 10th International AAAI Conference on Web and Social Media (ICWSM 2016)

Quanzhi Li, Sameena Shah, Xiaomo Liu, Armineh Nourbakhsh, 2017, Data Set: Word Embeddings Learned from Tweets and General Data, The 11th International AAAI Conference on Web and Social Media (ICWSM-2017).

Xiaomo Liu, Armineh Nourbakhsh, Quanzhi Li, Rui Fang, Sameena Shah, 2015, Real-time Rumor Debunking on Twitter, CIKM 2015.

Yang Liu and Yi-fang Brook Wu. 2018. Early Detection of Fake News on Social Media Through Propagation Path Classification with CNN. AAAI 2018

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 the 54th Meeting of the Association for Computational Linguistics. Association for Computer Linguistics, pages 393–398.

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

Jing Ma, Wei Gao, Kam-Fai Wong, 2017, Detect rumors in microblog posts using propagation structure via kernel learning, ACL 2017

Jing Ma, Wei Gao, Kam-Fai Wong, Detect Rumor and Stance Jointly by Neural Multi-task Learning, WWW 2018

M. Mendoza, B. Poblete, and C. Castillo. Twitter under crisis: Can we trust what we rt? In Proc. First Workshop on Social Media Analytics, 2010.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality. In Proceedings of NIPS, 2013.

Saif M Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016. Semeval-2016 task 6: Detecting stance in tweets. In Proceedings of the International Workshop on Semantic Evaluation, SemEval . volume 16.

V. Qazvinian, E. Rosengren, D. R. Radev, and Q. Mei. Rumor has it: Identifying misinformation in microblogs. EMNLP 2011

Sarvesh Ranade, Rajeev Sangal, and Radhika Mamidi. 2013. Stance classification in online debates by recognizing users' intentions. In Proceedings of the SIGDIAL 2013 Conference . pages 61–69.

Arkaitz Zubiaga, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Peter Tolmie. 2016. Analysing how people orient to and spread rumours in social media by looking at conversational threads. PloS one 11(3):e0150989.

K. Wu, S. Yang, and K. Q. Zhu. False rumors detection on sina weibo by propagation structures. In IEEE International Conference of Data Engineering, 2015.

F. Yang, Y. Liu, X. Yu, and M. Yang. Automatic detection of rumor on sina weibo. In Proc. of the ACM SIGKDD Workshop on Mining Data Semantics, page 13, 2012.