Rumor Stance Classification via Machine Learning with Text, User and Propagation Features

Kaizhou Xuan, Rui Xia
School of Computer Science and Engineering
Nanjing University of Science and Technology
Jiangsu, China
kaizhouxuan@gmail.com, rxia@njust.edu.cn

***Abstract*—Stance classification of user responses plays a key role in rumor detection on social media platforms. Stances are commonly divided into four categories: *support, deny, query* and *comment*, where the first three are particularly important for determining the confidence of rumors. Since people seldom express definite stance under rumors with unclear authenticity, it is difficult to judge such stances. Machine learning approaches have been proposed to address this problem with either manually designed features or automatically extracted features. However, the problem is related to many aspects of attributes and it is somehow difficult to summarize a comprehensive feature template or to learn an effective feature representation. In this work, we conduct an in-depth study on the feature engineering for this task. We screen out 18 salient features in three aspects including text, user and propagation. The experimental results on the RumorEval dataset show that coupled with these 18 features, a traditional logistic regression classifier even achieves the state-of-the-art performance and outperforms some complex neural networks such as long-short term memory networks that uses the same feature template or automatic feature extraction.**

***Index Terms*—Stance classification, rumor, feature engineering, machine learning**

I. Introduction

In recent years, social media platforms have become more active, and netizens are used to disseminate news and events and express their views. The convenience of the network also provides an ideal environment for the dissemination of some rumors, for example, the rapid spread of misinformation in a short period of time sometimes causes adverse effects. However, it is usually difficult to accurately judge the authenticity of rumors simply through their text contents. So, people has began to pay attention to the stances reflected in the relative responses [1]. Knowing the stances that users hold in responses to some rumors give useful insights, and some denying or querying voices reveal false rumors [2]–[4].

Stance classification is the problem of identifying the attitude taken by a user in a short piece of response text under a rumor microblog [5]. Typical stances include showing supporting and denying [6]. However, for a rumor whose authenticity is uncertain, people seldom express definite stances. RumorEval provided a dataset [7] which tagged users' stance as four categories - *support, deny, query* and *comment* (SDQC) [1], [8]:

- **Support:** the author of the response clearly supports the veracity of the rumor.
- **Deny:** the author of the response clearly denies the veracity of the rumor.
- **Query:** the author of the response could not confirm the veracity of the rumor, and seeks for additional evidence.
- **Comment:** the author of the response makes his/her own comment without a clear contribution to assessing the veracity of the rumor.

Among them, the main stances that can influence the veracity of a rumor are *support* and *deny*, while *query* can also partly show that the rumor might not be true. However, in the RumorEval dataset, the proportion of *comment* responses is over 66%. In addition, the source rumors are also marked with SDQC, and the publishers almost support the information he/she posted. Therefore, it is especially meaningful to detect the *deny* and *query* responses which have a small amount of training and testing data. We call all source rumors and responses as post, and do stance classification on the post level.

Rumor stance classification is a complex task, which is related to many aspects of attributes, including text, user, propagation, etc. However, it is somehow difficult to summarize a comprehensive feature template manually or to automatically learn an effective feature representation by machine learning models. For one thing, some of the previous work extracted a large number of artificial features and covered many aspects of microblogs [4], [9]. However, some features might not have a clear association with the class labels and sometimes counteract the classification effect. For another, it is difficult for deep learning models to extract explicit features in the aspects of user and propagation just based on post texts [5], [10]. The automatic extraction of text features is often insufficient to represent the posts.

In this paper, we screen out several social media features that are effective for stance classification, and classify them into text features, user features and propagation features. We initially examined on more than 40 features and conducted an in-depth study by feature engineering. Then, according to the distribution of each feature in the training set and its performance in the development set, we finally hold only 18 features. By combining these 18 features with traditional machine learning classifiers, we have achieved better results on RumorEval dataset than the previous work. The traditional logical regression (LR) classifier gives the state-of-the-art performance and improves the classification F1-score to 0.574

2375-9259/19/$31.00 ©2019 IEEE
DOI 10.1109/ICDMW.2019.00085

TABLE I
EXAMPLES OF RUMORS PUBLISHING AND RESPONSES.

SDQC stance classification.

Example 1:
user1: We understand there are two gunmen and up to a dozen hostages inside the cafe under siege at Sydney.. ISIS flags remain on display #7News [**support**]
user2: @user1 not ISIS flags [**deny**]
user3: @user1 sorry - how do you know it's an ISIS flag? Can you actually confirm that? [**query**]
user4: @user3 no she can't cos it's actually not. [**deny**]
user5: @user1 More on situation at Martin Place in Sydney, AU –LINK– [**comment**]
user6: @user1 Have you actually confirmed its an ISIS flag or are you talking shit [**query**]

Example 2:
user1: These are not timid colors; soldiers back guarding Tomb of Unknown Soldier after today's shooting #StandforCanada –PICTURE– [**support**]
user2: @user1 Apparently a hoax. Best to take Tweet down. [**deny**]
user3: @user1 This photo was taken this morning, before the shooting. [**deny**]
user4: @user1 I don't believe there are soldiers guarding this area right now. [**deny**]
user5: @user4 wondered as well. I've reached out to someone who would know just to confirm that. Hopefully get response soon. [**comment**]
user4: @user5 ok, thanks. [**comment**]

and accuracy to 0.805. In particular, we pay special attention to the categories with fewer samples, *deny* and *query*, and have made great improvements on the predicting results of these two categories. Furthermore, we also find that it is difficult to learn better representation of posts based on these features when using deep learning models.

II. RELATED WORK

The purpose of rumor stance classification task is to determining the type of orientation that each individual post expresses toward the disputed veracity of a rumor [11]. This is very meaningful for the judgement of true rumor and false rumor. In the early online debates, stances were divided into "for" and "against" [12]. Unlike stance classification in online debates, the stances of rumor responses are usually divided into four categories: *support, deny, query* and *comment*. Mendoza et al. labelled responses under several source rumors as *support, deny* and *query* [13]. For the rumors which release true information, they found that almost all the responses express supporting, and few express denying or doubt. But even for false information, only 50% of the responses show disagreement.

As early as in some rumor detection work, researchers have found the validity of some features for sentiment analysis of rumor texts [14]. Qazvinian et al. designed a large number of features to improve the effectiveness of the traditional classifiers [15]. Their pragmatic features included named entities, events, sentiment, and emoticons. Liu et al. proposed a rule-based approach, which performed better than that adopted in previous work [16]. Zeng et al. further enriched the feature sets from the aspects of Linguistic Inquiry and Word Count dictionaries [17].

Rumour stance classification task began to be widely studied in the RumourEval shared task at SemEval 2017 [8], [11]. Turing group extracted the relationship between responses and established a response sequence for each post [5]. They used a combination of word embeddings from Google News [18] and a small number of artificial features to represent each post. Then, long short-term memory network (LSTM) was selected for modeling sequential relationship, which can capture the dynamic changes of responses. UWaterloo group proposed a model using topic independent features from two categories: cue features and message specific features [19]. The main parts of the cue features were "belief", "report", "doubt", "knowledge" and "denial". For example, the presence of belief or knowledge words could be indicative of a response where the author expresses his/her support. The message specific features provided information about the writing style, such as the presence of punctuation marks, Twitter-specific characters (such as #, @) and number of words/characters in the post. We also draw lessons from the construction method of some features. ECNU group proposed a two-step classifier to solve the class imbalance problem [20]. The first-step classifier was designed to discriminate *comment* posts from non-comment posts. And the second-step classifier was designed to identify whether a post is *support, deny* or *query* towards the source rumor if the post was labeled as non-comment in the first step. In fact, they still used the combination of features and traditional classifiers to solve the problem. Mama Edha group tried to solve this classification task by building three different classifiers and combining the predictions with an ensemble method [10]. In their experiments, the single convolution neural network (CNN) provided the best accuracy, and the integrated method helped to improve the macro F1-score. NileTMRG group extracted a large number of features from rumor data manually, and applied them to many traditional classifiers [9]. Aker et al. further extended and described the feature set in detail [4]. In comparison, our work pays more attention to the selection of effective features instead of finding more features.

TABLE II
STATISTICS OF RUMOREVAL DATASET.
THE PERCENTAGE (%) OF EACH CATEGORY IS SHOWN IN PARENTHESES.
(DEV.:DEVELOPMENT)

	Posts	Support	Deny	Query	Comment
Training	4,238	841(20.0)	333(7.9)	330(7.8)	2,734(64.5)
Dev.	281	69(24.6)	11(3.9)	28(10.0)	173(61.6)
Testing	1,049	94(9.0)	71(6.8)	106(10.1)	778(74.2)
All	5,568	1,004(18.0)	415(7.5)	464(8.3)	3,685(66.2)

Subsequently, more and more work began to use deep neural networks to solve the stance classification problem. Poddar et al. used CNN to encode the text of each posts, and introduced LSTM and attention mechanism to judge the stance [21]. Ma et al. fused stance classification and rumor detection tasks into a multi-task model [22]. The model utilized two LSTM layers to model the propagation sequence of two tasks separately, and provided shared parameters through another LSTM layer. Their method did improve the accuracy of prediction, but the complexity of the model was a bit high.

SemEval-2019 Task 7 provides a similar extension task [23]. Many works has again utilized a large number of features to enhance the effectiveness of machine learning or deep learning models [24]–[27]. We hope to extract the most effective part from tremendous number of features.

III. TASK DESCRIPTION

A. Dataset

RumorEval dataset D, released by SemEval-2017 [8], originally came from real Twitter data and was tagged as SDQC [7], [28]. The dataset consists of rumors from eight real emergencies: Ottawa shooting, Ferguson riots, Germanwings crash, Charlie Hebdo, Sydney siege, Ebola essien, Prince Toronto and Putin missing. Each emergency contains a different number of rumors, where each disseminates some information and expresses a stance (SDQC). So the dataset D is usually written as a set of rumours $D = \{R_1, \ldots, R_n\}$, The responses t_i below a rumor R_i have been collected and also marked with stance annotations (SDQC), so a rumor can be considered as a set of posts including itself t_0: $R_i = \{(t_0^i, y_0^i), (t_1^i, y_1^i), \ldots, (t_{|R_i|}^i, y_{|R_i|}^i)\}$, where y_k^i represents the stance label of each post. We call all tagged rumors or responses as *post*.

Two sample structures of rumors and their responses are shown in Table I. The dataset has been divided into three parts: training set, development set and testing set, the detailed statistics of each part are shown in Table II. Obviously, more than 66% posts are marked as *comment*, while *deny* or *query* posts take only about 8% separately. This puts forward higher requirements for stance classification, and we not only focus on the overall prediction accuracy, but also on the accuracy and F1-score of each category.

B. Problem Statement

In this paper, a post t_k^i is represented as a set of features $t_k^i = \{f_1, f_2, \ldots, f_m\}$. Each feature is an independent value, so $t_k^i \in \mathbb{R}^m$. The goal of stance classification is to establish a classifier C between post representation and stance label $C(t_k^i) \rightarrow y_k^i$. We normalize the features and try to use some machine learning models to construct the classifier C.

TABLE III
FEATURES.

Class	Feature	Value
Text	URL	$\{0, 1\}$
	Topic	$\{0, 1\}$
	Question mark	$\{0, 1\}$
	Exclamatory mark	$\{0, 1\}$
	Number of negative words	$[0, +\infty]$
	Number of dirty words	$[0, +\infty]$
	Surprise score	$[0, 1]$
	Doubt score	$[0, 1]$
	Number of words	$[0, +\infty]$
	Sentiment score	$[0, 1]$
	Source Rumor Similarity	$[0, 1]$
User	Is user verified	$\{0, 1\}$
	Is source user verified	$\{0, 1\}$
	User registration time	$[0, +\infty]$
	Number of followers	$[0, 5)$
Propagation	Number of repost	$[0, +\infty]$
	Number of like	$[0, +\infty]$
	Is source rumor or response	$\{0, 1\}$

IV. FEATURES

Previous work has proposed many features that might be effective [5], [9], [14], [19], [20]. We further analyze and organize these features and divide them into text features, user features and propagation features. We have tested over 40 different features from these aspects, checking their distribution on the training set and performance on the development set.

Some invalid features were removed during testing. For example, picture existence, a feature used to record whether a post contains additional pictures, was removed because the differences of its proportions in different categories are very small. In addition, name entity, a feature used to check whether a post contains Person, Organization, Date, Location and Money tags, was removed because it is not easy to find and has no impact on the development set. In particular, we no longer use word embeddings as features. It is sensitive to different pre-training results, and results in the decline of the predicting on development set when occupying a large dimension space.

The valid 18 features are illustrated in the Table IV.

A. Text Features

Text features are the most important features. However, previous work has proved that the usage of high-dimensional word embedding representation could not effectively capture the features of the categories with rare samples. Therefore, we select the features from a more fine-grained perspective, and pay particular attention on how to capture the features of *deny* and *comment* posts.

- **URL:** We retrieve whether the post text contains web links (URL), and assign "1" if it contains any URL, or "0" otherwise. We find that the probability of giving URL

is significantly high in *support* posts, which indicates that such users are more inclined to give an external link to illustrate.

- **Topic:** Some hot events might be marked as topics, and the prominent sign is the appearance of "#". This feature will be marked as "1" if there are pairs of "#". Topics appear frequently in *deny* posts, which is beneficial to the judgement of misinformation.
- **Question mark:** Question mark is mainly used to express people's doubts. According our statistics, the probability of its appearance in *query* posts is as high as 75%. At the same time, it often appears in *deny* posts, too. Therefore, it plays a critical role in stance classification. Regardless of the number of times, we assign "1" if a post contains a question mask.
- **Exclamatory mark:** Exclamation mark is also an important symbol for people to express strong emotions, but it does not have clear emotional tendency. We hope that this feature can reinforce some certain emotions.
- **Number of negative words:** In order to increase the ability of the model to infer negative categories, we sort out some words expressing negative emotions and count the number of occurrences of these words in each post. First, all the common words used to deny are taken into account, such as "not", "never", "seldom", "unrealistic", "unreasonable", "unproved" and "wrong". Negative words commonly found on microblog platforms are then added, like "anger", "annoy", "upset" and "weird". In addition, since the usage of negative words is more complex, we only give them a statistical weight of 0.5. Finally, we find that the built vocabulary covers the *deny* posts more comprehensively.
- **Number of dirty words:** Some netizens use dirty words when expressing strong personal feelings. According to the usage habits on Twitter platform, we collect some dirty words, such as "idiot", "stupid", "fuck", "LIE" and "SUCK". These words appear in 17% of *deny* posts, much higher than the 3% on *comment* posts. Predictions of rare-data categories might be improved.
- **Surprise score and Doubt score:** Both features are designed to enhance the model's ability to detect *deny* and *query* posts. We find some words that express surprise ("amazing", "marvel", "astonish", "impress", etc.) or doubt ("confusion", "suspense", "uncertain", "suspect", etc.) in the training set to form two vocabulary lists. For each post, we assign the coincidence rate of text and vocabulary list as a feature.
- **Number of words:** In rumor responses, posts marked as *comment* may be written for the purpose of recording or reposting, and they may not have much content. After removing some special contents like URLs and topics, we count the number of words contained in a post as a feature.
- **Sentiment score:** We use the SenticNet 5 [29] to create this feature. The tool returns a range from 0 to 1 with 0 indicating "very negative" and 1 "very positive". We find that the probability of low scores in *deny* category is higher than that in other categories.
- **Source Rumor Similarity:** On the Twitter platform, if a response supports the idea or content of the source rumor, it may duplicate the text of the source rumor. Therefore, we calculate the collinearity of words between a post and its source rumor as this feature. Facts have proved that the value of *support* posts are generally 50% higher than those of other categories.

B. User Features

User features focus on determining the user's credibility. Users with high credibility tend to give more convincing *support* or *deny* views.

- **Is user verified and Is source user verified:** Verified users usually have a certain degree of social recognition in some aspect, so they would not easily make uncertain statements to avoid the impact of spreading misinformation. Conversely, once they explicitly support or oppose a rumor, their voices are more credible. We take whether the current user is verified as a feature to adjust the credibility of he/she. At the same time, whether the user who published the source rumor is verified or not is taken as an another feature to improve the possibility of *support* or *deny* responses.
- **User registration time:** The time of user registration contributes to the its credibility. Users who have been active on Twitter platforms for a long time will be responsible for their accounts, so they would be more careful about what they say. Some newly registered users might be zombie-users, which are used to maliciously spread misinformation. We assign this feature with the number of years that users have registered so far.
- **Number of followers:** Similar to previous user features, the number of followers can also affect users' credibility. Users with more followers or friends tend to have a certain influence and seldom disseminate misinformation. In addition, we find that the average number of followers from *support* posts is 20 times that of other categories. It is unbelievable that there is such a big gap between them. At the same time, the numbers of followers between users varies greatly, some are millions, and some are only a few. We set five grades by zero (0), one thousand (1), ten thousand (2), one hundred thousand (3), and one million (4), and calculate the decimals by equal proportions within each interval. This can reduce the error caused by the feature itself.

C. Propagation Features

Propagation features refer to the dynamic effects of a rumor after its release. Typically, posts with clear stance are easier to get people's attention and interaction.

- **Number of repost and number of like:** The numbers of reposts and likes of a rumor are the decisive indicators to measure their influence. Usually, support or denying responses from influential users are reposted more. In

TABLE IV
PREDICTING RESULTS OF DIFFERENT CLASSIFIER.
(ACC.: ACCURACY, S: SUPPORT, D: DENY, Q: QUERY, C: COMMENT)

Classifier	Acc.	F1-score				
		Macro	S	D	Q	C
LR	**0.805**	**0.574**	**0.439**	**0.130**	**0.549**	**0.885**
DT	0.774	0.496	0.419	0.027	0.546	0.867
RF	0.787	0.538	0.403	0.130	0.453	0.875
SVC	0.773	0.489	0.408	0.111	0.468	0.869
NB	0.757	0.397	0.413	0.027	0.000	0.863

our statistics, these features are related to the number of users' followers, and their impact on *support* category is particularly important.

- **Is source rumor or response:** This feature can distinguish the distribution differences between the source rumors and the responses. More than 90% of all source rumors express *support* for the text contents, while only 10% express *deny* or *comment*. This is because the earliest people who spread the message often express trust in the information. Therefore, for responses, the model can focus more on the possibilities of other categories.

V. EXPERIMENT

A. Experiment Setting

The RumorEval dataset has been split into training, development and test sets. We use the training set to train the model, and determine the hyper-parameters via testing on the development set. For the final predicting results on test set, we calculate the accuracy and F1-score for each category and compare them with previous work.

B. Classifier Comparison

We use several classical machine learning classifiers based on scikit-learn to fit these features [30], including logistic regression (LR), decision tree (DT), random forest (RF), linear support vector machine (SVC) and naive bayes (NB). The results of each classifier are shown in Table IV.

For *deny* category, many previous work could not perform effective feature capture and prediction. This situation has been improved with our feature selection. Based on our 18 features, each model has valid predictions under this category.

On the whole, LR outperforms other models in all the metrics.

C. Overall Comparison

We compare with the following stance classification methods.

- **Turing:** A method that modeling the response sequence with LSTM. A few features are also added to the original text representation of the posts to improve the effect of the model. [5]
- **UWaterloo:** An approach makes use of topic independent features from two categories, namely cue features and message specific features to fit a gradient boosting classifier. [19]

TABLE V
PREDICTING RESULTS OF ALL THE MODEL.
(ACC.: ACCURACY, S: SUPPORT, D: DENY, Q: QUERY, C: COMMENT)

Model	Acc.	F1-score				
		Macro	S	D	Q	C
Turing	0.784	0.434	0.403	0.000	0.462	0.873
UWaterloo	0.780	0.450	0.397	0.052	0.494	0.869
ECNU	0.778	-	-	-	-	-
MamaEdha	0.752	0.319	0.403	0.000	0.018	0.856
NileTMRG	0.760	-	-	-	-	-
Aker	0.790	-	-	-	-	-
Ours	**0.805**	**0.574**	**0.439**	**0.130**	**0.549**	**0.885**

- **ECNU:** A two-step traditional classifier based on text vectors and other manually features. It employs an ensemble model incorporating LR, SVM, DT, and ADB (AdaBoost) for the first step classification, while using LR directly for the second step. [20]
- **MamaEdha:** A ensemble classification approach of combining CNN (Convolutional Neural Networks) with both automatic rule mining and manually written rules. [10]
- **NileTMRG:** A method based on a large number of artificial features. The best result is achieved on LR model. [9]
- **Aker:** A decision tree model which is supplemented by more designed features. [4]

Table V shows the results of overall comparison. Turing won the first place in the SemEval-2017 competition. They introduced the idea of sequence modeling and achieved good results. Among their features, word embedding accounts for a large proportion. However, these features do not make the model achieve any effect on the judgment of the *deny* category. The features of UWaterloo work well, and the macro F1-score of their model is also higher than Turing's. They have made breakthroughs in the *deny* category and maintained the high level of the other three categories.

We compare the results of LR classifier with other models. Our features significantly improve the stance classification results on *deny* and *query* categories. At the same time, our attention to specific categories does not reduce the performance on *comment* category which has the largest amount of posts.

D. Discussion on Features

We try to remove some of the current 18 features to further detect the effects of these features. Part of the results are shown in Table VI.

In our experiments, we find that text features play a decisive role. After removing all the text features, other features can hardly capture the effective information of the first three categories. Among all the text features, whether a post has a question mark is a very important feature. As we have mentioned in Section IV, question mark is a key signal of *query*. Without this feature, the model can hardly detect any *query* posts. Similarly, the feature of dirty word is no less important than question mark. Dirty words can be interpreted as a sign of the *deny* stance. The surprise score is more like a reinforcing feature. It cannot control whether a category

TABLE VI
FEATURE REMOVAL TEST.
(ACC.: ACCURACY, S: SUPPORT, D: DENY, Q: QUERY, C: COMMENT, PROP.: PROPAGATION, FEA.: FEATURE)

Feature test	Acc.	F1-score				
		Macro	S	D	Q	C
baseline	**0.805**	**0.574**	**0.439**	**0.130**	**0.549**	**0.885**
-Question mark	0.764	0.419	0.428	0.080	0.000	0.862
-Dirty word	0.797	0.495	0.439	0.000	0.520	0.880
-Surprise score	0.792	0.535	0.443	0.104	0.438	0.878
-All text fea.	0.710	0.280	0.041	0.000	0.000	0.852
-Is user verified	0.792	0.542	0.428	0.105	0.444	0.878
-Registration time	0.792	0.554	0.421	0.130	0.506	0.877
-All user fea.	0.783	0.506	0.427	0.100	0.414	0.873
-Repost number	0.802	0.563	0.439	0.105	0.526	0.884
-All prop. fea.	0.797	0.549	0.400	0.105	0.490	0.881

TABLE VII
FEATURE ADDITION TEST.
(ACC.: ACCURACY, S: SUPPORT, D: DENY, Q: QUERY, C: COMMENT, PROP.: PROPAGATION, FEA.: FEATURE)

Feature test	Acc.	F1-score				
		Macro	S	D	Q	C
baseline	0.805	**0.574**	0.439	**0.130**	**0.549**	**0.885**
+Word Embedding	0.724	0.352	0.214	0.000	0.121	0.840
+Picture existence	0.795	0.549	0.439	0.105	0.458	0.880
+Positive word	0.797	0.549	0.400	0.105	0.490	0.881
+Response time	**0.808**	0.572	**0.440**	0.130	0.542	0.881

is learned, but it has a good reinforcement effect on *deny* and *query* judgement. The model can be corrected from the tendency of predicting all the posts as *comment*.

In line with the original intention of our design, user features and propagation features also enhance the predicting confidence. Removing these features will not bring a huge weakening of the model, but it will affect the confidence of the model in category judgment. Such kind of features should not be too many, otherwise, it is difficult for the model to learn the decision of predicting a post as a rare category.

In addition, we add some removed features to the feature template to observe their performance on the testing set. Table VII shows the results of these experiments. Word embedding feature uses the mean value of all the word embeddings of words in a post to construct. If we don't put the word embeddings into the deep neural network for further encoding, but using them as features alone, it would seriously interfere on rumor stance classification. The picture existence feature marks whether a post using an additional picture as its content, while the positive word feature counts the collinearity of a post text and some given positive words. None of these features will have a large impact on the final result, but they will affect the confidence of the model on rare-sample categories like *deny* and *query*.

In particular, there is a feature that improves the performance of the testing set-response time. Its value depends on the publish time difference between the response post and the source rumor. The addition of this feature increases the predicting results of the *support* posts. Usually, as time goes by, the authenticity of a rumor is gradually confirmed, and people's stances are more clear.

TABLE VIII
EXPERIMENTS ON DEEP LEARNING MODELS.
(ACC.: ACCURACY, S: SUPPORT, D: DENY, Q: QUERY, C: COMMENT, PROP.: PROPAGATION, FEA.: FEATURE)

Model	Acc.	F1-score				
		Macro	S	D	Q	C
baseline	0.805	**0.574**	**0.439**	**0.130**	0.549	**0.885**
LSTM & all features	0.799	0.551	0.436	0.054	**0.554**	0.882
LSTM & word embedding	0.744	0.350	0.353	0.000	0.036	0.852
LSTM & all above	0.783	0.506	0.427	0.100	0.414	0.873

E. Comparisons with Deep Learning Models

In many previous work, deep learning models were introduced to model the posts or propagation sequences [5], [10], [31]. We try to use LSTM to model the response sequences of a source rumor. We extract each response sequence and ensure that the last post in the sequence is a post that needs to be predicted. Therefore, each post is converted to a sequence. LSTM is used to model each sequence, and the output of the last step is used as the final representation. We use a combination of linear layer and softmax for classification.

As shown in the table VIII, we first take the 18 features to build the representation of a post in the sequences and use them as the input of the LSTM model. Although the model consumes more space and time for training, it can not show better performance than the simple LR, and there is a decline in *deny* categories. Furthermore, the addition of word embedding has obviously dealt a serious blow to *support* and *query* categories. This is similar to what we observed in Table VII.

VI. CONCLUSION AND FUTURE WORK

Rumor stance classification usually relies on large-scale feature templates or complex neural networks, but has difficulty making an effective feature representation. In this work, we further study and screen out only 18 features and apply them to traditional machine learning classifiers. These features are summarized from three aspects of text, user and propagation, and pay more attention to the categories with a small number of training samples. Based on these features, simple logistic regression classifier achieves the state-of-the-art performance and outperforms some previous methods with complex features or models. In particular, the improvements of the performances on *deny* and *query* posts further prove the effectiveness of our feature template.

For future work, we want to test more datasets to find more features and verify their validity, and improve the generalization performance of the feature templates.

ACKNOWLEDGMENTS

The work was supported by the Natural Science Foundation of China (No. 61672288), and the Natural Science Foundation of Jiangsu Province for Excellent Young Scholars (No. BK20160085). Rui Xia is the corresponding author of this paper.

REFERENCES

[1] R. Procter, F. Vis, and A. Voss, "Reading the riots on twitter: methodological innovation for the analysis of big data," *International journal of social research methodology*, vol. 16, no. 3, pp. 197–214, 2013.

[2] M. Lukasik, T. Cohn, and K. Bontcheva, "Classifying tweet level judgements of rumours in social media," *arXiv preprint arXiv:1506.00468*, 2015.

[3] L. Derczynski, K. Bontcheva, M. Lukasik, T. Declerck, A. Scharl, G. Georgiev, P. Osenova, T. P. Lobo, A. Kolliakou, R. Stewart, *et al.*, "Pheme: Computing veracity—the fourth challenge of big social data," in *Proceedings of the Extended Semantic Web Conference EU Project Networking session (ESCW-PN)*, 2015.

[4] A. Aker, L. Derczynski, and K. Bontcheva, "Simple open stance classification for rumour analysis," in *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pp. 31–39, 2017.

[5] E. Kochkina, M. Liakata, and I. Augenstein, "Turing at semeval-2017 task 8: Sequential approach to rumour stance classification with branch-lstm," *arXiv preprint arXiv:1704.07221*, 2017.

[6] S. Mohammad, S. Kiritchenko, P. Sobhani, X. Zhu, and C. Cherry, "Semeval-2016 task 6: Detecting stance in tweets," in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pp. 31–41, 2016.

[7] A. Zubiaga, M. Liakata, R. Procter, G. W. S. Hoi, and P. Tolmie, "Analysing how people orient to and spread rumours in social media by looking at conversational threads," *PloS one*, vol. 11, no. 3, p. e0150989, 2016.

[8] L. Derczynski, K. Bontcheva, M. Liakata, R. Procter, G. W. S. Hoi, and A. Zubiaga, "Semeval-2017 task 8: Rumoureval: Determining rumour veracity and support for rumours," *arXiv preprint arXiv:1704.05972*, 2017.

[9] O. Enayet and S. R. El-Beltagy, "Niletmrg at semeval-2017 task 8: Determining rumour and veracity support for rumours on twitter.," in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pp. 470–474, 2017.

[10] M. G. Lozano, H. Lilja, E. Tjörnhammar, and M. Karasalo, "Mama edha at semeval-2017 task 8: Stance classification with cnn and rules," in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pp. 481–485, 2017.

[11] A. Zubiaga, A. Aker, K. Bontcheva, M. Liakata, and R. Procter, "Detection and resolution of rumours in social media: A survey," *ACM Computing Surveys (CSUR)*, vol. 51, no. 2, p. 32, 2018.

[12] M. A. Walker, P. Anand, R. Abbott, J. E. F. Tree, C. Martell, and J. King, "That is your evidence?: Classifying stance in online political debate," *Decision Support Systems*, vol. 53, no. 4, pp. 719–729, 2012.

[13] M. Mendoza, B. Poblete, and C. Castillo, "Twitter under crisis: Can we trust what we rt?," in *Proceedings of the first workshop on social media analytics*, pp. 71–79, ACM, 2010.

[14] C. Castillo, M. Mendoza, and B. Poblete, "Information credibility on twitter," in *Proceedings of the 20th international conference on World wide web*, pp. 675–684, ACM, 2011.

[15] V. Qazvinian, E. Rosengren, D. R. Radev, and Q. Mei, "Rumor has it: Identifying misinformation in microblogs," in *Proceedings of the conference on empirical methods in natural language processing*, pp. 1589–1599, Association for Computational Linguistics, 2011.

[16] X. Liu, A. Nourbakhsh, Q. Li, R. Fang, and S. Shah, "Real-time rumor debunking on twitter," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pp. 1867–1870, ACM, 2015.

[17] L. Zeng, K. Starbird, and E. S. Spiro, "# unconfirmed: Classifying rumor stance in crisis-related social media messages," in *Tenth International AAAI Conference on Web and Social Media*, 2016.

[18] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[19] H. Bahuleyan and O. Vechtomova, "Uwaterloo at semeval-2017 task 8: Detecting stance towards rumours with topic independent features," in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pp. 461–464, 2017.

[20] F. Wang, M. Lan, and Y. Wu, "Ecnu at semeval-2017 task 8: Rumour evaluation using effective features and supervised ensemble models," in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pp. 491–496, 2017.

[21] L. Poddar, W. Hsu, M. L. Lee, and S. Subramaniyam, "Predicting stances in twitter conversations for detecting veracity of rumors: A neural approach," in *2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 65–72, IEEE, 2018.

[22] J. Ma, W. Gao, and K.-F. Wong, "Detect rumor and stance jointly by neural multi-task learning," in *Companion Proceedings of the The Web Conference 2018*, pp. 585–593, International World Wide Web Conferences Steering Committee, 2018.

[23] G. Gorrell, E. Kochkina, M. Liakata, A. Aker, A. Zubiaga, K. Bontcheva, and L. Derczynski, "Semeval-2019 task 7: Rumoureval, determining rumour veracity and support for rumours," in *Proceedings of the 13th International Workshop on Semantic Evaluation*, pp. 845–854, 2019.

[24] Q. Li, Q. Zhang, and L. Si, "eventai at semeval-2019 task 7: Rumor detection on social media by exploiting content, user credibility and propagation information," in *Proceedings of the 13th International Workshop on Semantic Evaluation*, pp. 855–859, 2019.

[25] B. Ghanem, A. T. Cignarella, C. Bosco, P. Rosso, and F. M. R. Pardo, "Upv-28-unito at semeval-2019 task 7: Exploiting post's nesting and syntax information for rumor stance classification," in *Proceedings of the 13th International Workshop on Semantic Evaluation*, pp. 1125–1131, 2019.

[26] R. Yang, W. Xie, C. Liu, and D. Yu, "Blcu_nlp at semeval-2019 task 7: An inference chain-based gpt model for rumour evaluation," in *Proceedings of the 13th International Workshop on Semantic Evaluation*, pp. 1090–1096, 2019.

[27] A. Janchevski and S. Gievska, "Andrejjan at semeval-2019 task 7: A fusion approach for exploring the key factors pertaining to rumour analysis," in *Proceedings of the 13th International Workshop on Semantic Evaluation*, pp. 1083–1089, 2019.

[28] Z. Tufekci, "Big questions for social media big data: Representativeness, validity and other methodological pitfalls," in *Eighth International AAAI Conference on Weblogs and Social Media*, 2014.

[29] E. Cambria, S. Poria, D. Hazarika, and K. Kwok, "Senticnet 5: Discovering conceptual primitives for sentiment analysis by means of context embeddings," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[30] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, *et al.*, "Api design for machine learning software: experiences from the scikit-learn project," *arXiv preprint arXiv:1309.0238*, 2013.

[31] Y.-C. Chen, Z.-Y. Liu, and H.-Y. 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)*, pp. 465–469, 2017.