ECNU at SemEval-2017 Task 8: Rumour Evaluation Using Effective Features and Supervised Ensemble Models

Feixiang Wang[1], **Man Lan**[1,2*], **Yuanbin Wu**[1,2]
[1]Department of Computer Science and Technology,
East China Normal University, Shanghai, P.R.China
[2]Shanghai Key Laboratory of Multidimensional Information Processing
51151201049@stu.ecnu.edu.cn, {mlan, ybwu}@cs.ecnu.edu.cn

Abstract

This paper describes our submissions to task 8 in SemEval 2017, i.e., *Determining rumour veracity and support for rumours*. Given a rumoured tweet and a plethora of replied tweets, subtask A is to label whether these tweets are *support*, *deny*, *query* or *comment*, and subtask B aims to predict the veracity (i.e., *true*, *false*, and *unverified*) with a confidence (in range of 0-1) of the given rumoured tweet. For both subtasks, we adopted supervised machine learning methods incorporating rich features. Since the training data is imbalanced, we specifically designed a two-step classifier to address subtask A .

1 Introduction

With the rapid development of social media in recent years, people cannot only stay abreast of ongoing events and breaking news, but also express their own views freely. News can spread quickly in social media platforms through a large amount of users, whilst those pieces of unverified information often spawn rumours. The RumourEval (Derczynski et al., 2017) task aims to identify how users in social media networks regard the originating rumours and reply to them, as well as analysis and determining veracity of rumoured tweets. The organizer provides tree-structured conversations that are associated with breaking news and consisting of originating rumoured tweets and tweets replying to them.

There are two subtasks in RumourEval. The propose of subtask A is, given the related breaking news, to predict the class (i.e., *support*, *deny*, *query*, and *comment*) of the originating rumoured tweet (i.e., source tweet) and reactions (i.e., replied tweets). The goal of subtask B is to determine the veracity and confidence of the given rumoured tweet, participants are required to return a label of rumour as *true*, *false* or *unverified*, with a confidence value in the range of 0-1.

We treated the two subtasks as multi-classification problems, and designed multiple effective natural language processing (NLP) features to build classifiers to make predictions. Besides, rumour detection is relevant to sentiment analysis, for example, *support* and *deny* can be viewed as positive and negative sentiment respectively. Therefore, we solved the problem with the aid of a number of sentiment-related features. Due to the imbalanced characteristic of the training data, we specifically adopted a two-step classifier to deal with subtask A. Firstly, tweets would be separated into two categories: *comment* and *non-comment*, then the tweets labeled as *non-comment* would be classified as *support*, *deny* or *query*. On the other hand, we directly adopted a three-classification system for subtask B to label rumoured tweets as *true*, *false* or *unverified* along with confidence.

2 System Description

For both subtask, we extracted rich features from the training data and then built classifiers to make predictions. For subtask A, we designed a two-step classification system. The first step (1-step) classifier is to discriminate *comment* tweets from *non-comment* tweets. And the second step (2-step) classifier is to identify whether a tweet is *support*, *deny* or *query* towards the rumour if the tweet was labeled as *non-comment* in the 1-step classification. The 1-step can be viewed as determining whether a tweet is *objective* (*comment*) or *subjective* (*non-comment*). The 2-step is actually to classify a *non-comment* tweet that expresses positive (*support*), negative (*deny*) or doubtful (*query*)

Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017), pages 491–496,
Vancouver, Canada, August 3 - 4, 2017. ©2017 Association for Computational Linguistics

sentiment. While for subtask B, we simply implemented a three-classification system to determine whether the given rumoured tweet is *true*, *false* or *unverified* and returned a confidence of label.

2.1 Feature Engineering

In this section, we give the detail of feature engineering. Five types of NLP features are designed to capture effective information from the given tweets.

Linguistic-informed Features

- **Word N-grams:** We extracted word n-grams features ($n = 1, 2$) from tweets. However, a word has various forms, therefore we also constructed lemmatization and stem word n-grams features ($n = 1, 2$). To accomplish that, we acquired the lemmatization and stem of words from the pending sentences, using the *Stanford CoreNLP tools*[1].

- **NER:** There are different types of words in tweets, such as a tweet "*Gunman Takes Hostages In Sydney Cafe*" that has useful information like person and location to help to detect rumours. NER feature can effectively express aforesaid information. The 12 types (i.e., *DURATION, SET, NUMBER, LOCATION, PERSON, ORGANIZATION, PERCENT, MISC, ORDINAL, TIME, DATE, MONEY*) named entities are labeled by *Stanford CoreNLP tools*. We used a 12-dimensions binary feature to indicate the entities in tweet.

There are some particular elements in tweets, that can help to predict labels of tweets. For instance, hashtag and mentioned entity (e.g., "#semeval", "@YouTube") express the topic information of the tweets, and several special punctuation and emotions (e.g., "!", "?", and ":)") reveal the sentiment information of users.

Tweet domain Features

We collected all the hashtags and mentioned entities appeared in training tweets, using unigram feature to imply whether a tweet contained such information.

- **Punctuation:** Considering that users often use exclamation marks and question marks to express strongly surprised and questioned feelings, we extracted 7-dimensions punctuation features by recording rules of punctuation marks in the tweets (i.e., whether there is one or more question marks or exclamation marks, whether there is a question mark or an exclamation mark in the end of sentence).

- **Emoticon:** We collected 67 emoticons labeled with positive and negative scores from the Internet[2], and used a 67-dimensions feature to record the sentiment score of the emoticon in tweets.

- **Event:** Training data consists of plenty of tree-structured conversations that cover eight breaking news. We gathered several keywords[3] about these events from the Internet to extract corresponding unigram feature.

Metadata contains important information and can indicate the popularity of a tweet and the credibility of the author of a tweet. For example, features like "*favorite_count: 1340*", "*retweet_count: 500*" may indicate whether the tweet is being watched; "*verified: false*", "*protected: true*" perhaps imply whether the author is trustworthy.

Tweet metadata Features

We extracted two types of metadata information:

- **Tweet metadata:** We designed a 5-dimensions feature that consists of *tweet favorite count*, *retweet count*, *pre-retweet count* (i.e., the retweet count of the last replied tweet), *create time gap* (i.e., the time interval between the tweet and previous replied tweet) and *tweet level* (i.e., the layer of the tweet in a tweet conversation flow). These numerical characteristics are normalized by *0-1* normalization.

- **User metadata:** In addition to the metadata of a tweet, users also have some instrumentally valuable metadata as follows: *list count*, *followers count*, *user favourites count*, *friends count*, *verified*, *protected*, *default profile*, *profile use background image*, and *geo*

[1] http://stanfordnlp.github.io/CoreNLP/

[2] https://github.com/haierlord/resource/blob/master /Emoticon.txt

[3] We enter the hashtag of source tweet on the Internet, to collect keywords from the headlines of relevant news. For example, we manually extracted "charlie", "hebdo", "attack" and "terror" from the title "Charlie Hebdo attack: Three days of terror - BBC News".

enabled. The first four are numerical features that need normalization, and others are binary features. The aforementioned features form a 9-dimensions feature.

Word Vector Features

A lot of recent studies on NLP applications are reported to have good performance using word vectors, such as ducument classification (Sebastiani, 2002), parsing (Socher et al., 2013), and question answering (Lan et al., 2016a), We adopted two widely-used word vectors, i.e., GoogleW2V (Mikolov et al., 2013) and GloVe (Pennington et al., 2014). However, semantic word vectors find similar words with similar context rather than similar sentiment information. Several recent works focused on sentiment word vectors using neural network based models (Lan et al., 2016b). In this work, we also adopted two sentiment word vectors, one is SSWE (Tang et al., 2014) and the other is a home-made sentiment word vector from our previous work. To obtain the representation of a tweet, for each word in a tweet, we concatenated the maximum, minimum and mean of each dimension as a tweet vector [*min-max-mean*].

- **GoogleW2V:** We adopted the pre-trained available 300-dimensions word vectors that were trained on 100 billion words from Google News by word2vec tool[4]

- **GloVe:** The 100-dimensions word vectors we used were trained on 2 billion tweets and supplied in *GloVe*[5].

- **SSWE:** The sentiment-specific word embeddings were trained by using multi-hidden-layers nerual network with a vector size of 50.

- **ZSWE:** The 200-dimensions home-made sentiment word vectors were trained with NRC140 tweet corpus by the *Combined-Sentiment Word Embedding* Model.

Word-cluster Feature

To further group similar words into a small set and to make better use of word semantic information, we clustered all the words of tweets by k-means algorithm. The pending words were firstly represented as 300-dimensions word vectors by looking up pre-trained GoogleW2V, then grouped into 80 clusters. Thus we adopted 80-dimensions binary feature to mark whether the words of a certain cluster appeared in the tweet.

2.2 Learning algorithms and Evaluation metrics

Based on above multiple features, we explored several learning algorithms to build classification models, e.g., Logistic Regression (LR), supplied in *liblinear tools*[6], Support Vector Machines (SVM), Decision Trees (DT), Random Forests (RF), AdaBoost (ADB), and Gradient Tree Boosting (GDB), implemented in *scikit-learn*[7]. We also ensembled the effective learning algorithms using majority vote strategy.

The official evaluation measure for both subtasks is *accuracy*.

3 Experiments and Results

3.1 Datasets

The statistics of the datasets provided by SemEval 2017 task 8 are shown in Table 1.

Subtask A	**support(%)**	**query(%)**	**deny(%)**	**comment(%)**
train	841(19.8)	330(7.8)	333(7.9)	2,734(64.5)
dev	69(24.6)	28(10.0)	11(3.9)	173(61.6)
test	94(9.0)	106(10.1)	71(6.8)	778(74.2)
Subtask B	**true(%)**	**false(%)**	**unverified(%)**	-
train	127(46.7)	50(18.4)	95(34.9)	-
dev	10(40.0)	12(48.0)	3(12.0)	-
test	8(28.6)	12(42.9)	8(28.6)	-

Table 1: Statistics of training (train), development (dev) and testing (test) data sets in SemEval 2017 Task 8.

The train and dev sets are associated with eight different breaking news in English, i.e., *charliehebdo*, *ebola-essien*, *ferguson*, *germanwings-crash*, *ottawashooting*, *prince-toronto*, *putinmissing*, and *sydneysiege*. They are made up of 297 Twitter conversations including $4,519$ tweets in total. Apart from the eight original breaking news, the test set adds two new, i.e., *hillaryshealth* and *save-marinajoyce*, and it contains 28 conversations and $1,049$ tweets. This corpus is collected using the method described in (Zubiaga et al., 2016).

3.2 Data Preprocessing

To deal with the informal characteristic of tweets, we performed tweet normalization to convert elon-

[4] https://code.google.com/archive/p/word2vec

[5] http://nlp.stanford.edu/projects/glove/

[6] https://www.csie.ntu.edu.tw/ cjlin/liblinear/

[7] http://scikit-learn.org/

Subtask		Subtask A									Subtask B			
		1-step				2-step					-			
Algorithm		LR	SVM	DT	ADB	LR	SVM	RF	ADB	GDB	LR	SVM	RF	GDB
Tweet domain	Hashtag					√				√	√	√	√	√
	Mentioned_entity	√	√	√		√	√			√			√	
	Punctuation	√	√	√	√	√	√	√		√		√	√	√
	Emoticon		√	√		√	√					√		√
	Event										√		√	√
Metadata	Tweet metadata	√	√	√	√	√						√	√	
	User metadata	√	√			√	√		√					
Word-cluster	Word-cluster		√		√	√			√		√	√	√	
Linguistic	unigram			√				√	√					
	unigram_lemma						√			√	√	√	√	
	unigram_stem					√								√
	bigram													
	bigram_lemma							√				√		
	bigram_stem					√								
	ner							√			√		√	√
Word Vector	GoogleW2V					√								
	GloVe	√	√				√							
	SSWE									√				
	ZSWE								√					
Accuracy (%)		80.07	81.14	79.36	81.14	**81.49**	81.14	80.07	80.43	81.39	70.03	70.70	64.98	67.34
Ensemble (%)		**83.99**				80.07					**71.04**			

Table 2: Results of feature and algorithm selection experiments for both Subtask A and Subtask B. 1-step, 2-step represent the first and second classification of subtask A respectively,

gated words and slang words into original word. For elongated word (e.g., "*sooo*"), we implemented a home-made application to transform it into "*so*", and for slang words, we collected a big dictionary[8] from the Internet to convert "*LOL*" into "*laugh out loud*". Then we conducted tokenization, lemmatization and stemming with the aid of *Stanford CoreNLP tools*[9].

3.3 Experiments on training data

The Table 2 lists the results of the best feature set with respect to top learning algorithms on two subtasks. Note that for subtask A, we adopted a two-step classification. The accuracy of 1-step is calculated on two classes (i.e., *comment* and *non-comment*), and that of 2-step is calculated on four classes (i.e., *support*, *deny*, *query* and *comment*). Since the dev set of subtask B is not enough (only 25 samples), we combined train and dev sets and performed a 2-fold cross-validation. Furthermore, we also performed ensemble to combine the results of top learning algorithms with their optimum feature sets, which are shown as the last row in Table 2.

From Table 2, we observe the findings as follows:
(1) Among 7 algorithms, LR and SVM consistently perform well in the three classifications. Besides, ADB does a good job in two classifications in subtask A, RF and GBD have a good performance in 2-step of subtask A and subtask B.
(2) Generally, Tweet domain, metadata and Word-cluster features make a considerable contribution for both subtasks, and they can achieve promising performance with different algorithms. The possible reasons are: (a) Tweet domain features not only contain sentiment information (e.g., Punctuation and Emoticon), but also include topic information (e.g., Hashtag, Mentioned_entity, and Event). (b) The numerical characteristic (e.g, tweet favorite count, retweet count, etc) of metadata can indicate that whether a tweet is being closely watched and worthy of commenting. Binary features (e.g., friends count, is-verified, is-protected, etc) reveal that whether the author of a tweet is trustworthy. (c) The Word-cluster feature provides semantic information.
(3) The performance of Linguistic-informed and Word vector features in three classifications is mixed. The Linguistic-informed features do not work in the 1-step, however they contribute to the 2-step classification and subtask B. By observation, the lemmatization and stem n-gram outperform the original n-gram probably because that lemmatization and stem unify the form of words, thus reducing the dimension of feature and unnecessary noise. For Word vector, GloVe slightly outperforms other word vectors.
(4) From the algorithm comparison experiments, the ensemble models for 1-step of subtask A and subtask B are superior to the models using single algorithms, different learning algorithms

[8] https://github.com/haierlord/resource/blob/master/slangs
[9] http://stanfordnlp.github.io/CoreNLP/

contribute differently to the classification performance, that is why we conduct majority vote to ensemble those effective learning algorithms. However, we directly use the LR algorithm in 2-step on account of its best performance.

3.4 System Configuration

Based on the above experimental results, we constructed our submissions as follows: For subtask A, we employed an ensemble model incorporating LR, SVM, DT, and ADB for 1-step classification, while used LR directly for 2-step classification. For subtask B, we also adopted an ensemble model with LR, SVM, RF, GDB to predict labels of rumoured tweets, and probabilities of labels returned as confidence values. The parameters of every algorithms are listed as follows: LR with $c = 1$, SVM with *kernel=linear*, $c = 0.1$, RF with *n_estimators*= 10, ADB with *n_estimators*= 100, GDB with *n_estimators*= 100, and DT with default parameters.

3.5 Results on test data

Tabel 3 shows the officially-released results of our models and top-ranked teams. We ranked the third for both subtasks in terms of *accuracy*, the second for subtask B on the *RMSE* evaluation (a higher *accuracy* is better, while a lower *RMSE* is better). The predict results of test data are inferior to the results of dev set, especially for subtask B. we partly blame it for two reasons: (1) The addition of two breaking news (i.e., *hillaryshealth* and *save-marinajoyce*). The feature set used in subtask B can not capture unseen words in new topics, so the model may have a limited generalizability. (2) The test set is too small (only 28 samples).

Subtask	System	Accuracy(%)	RMSE
Subtask A	**ECNU**	**77.8(3)**	-
	Turing	78.4(1)	-
	Uwaterloo	78.0(2)	-
Subtask B	**ECNU**	**46.4(3)**	**0.736(2)**
	NileTMRG	53.6(1)	0.672(1)
	IKM	53.6(2)	0.763(3)

Table 3: Performance of our models and top-ranked teams on both two subtasks. The numbers in the brackets are the official rankings.

4 Conclusion

For both subtasks, we adopted supervised machine learning methods incorporating rich features. We adopted a two-step classifier to address subtask A to solve the imbalance of training data, and a simplified three-classification for subtask B. We originally thought that features with good generalization performance, such as Linguistic-informed and Word vector features would perform well in both subtasks, but in fact that was not the case. On the contrary, good performance can be achieved with several features like Tweet domain and Metadata features closely related with the tweets. From the final results in test data, in the future work, we need to build a topic independent model to achieve better generalizability.

Acknowledgments

This research is supported by grants from Science and Technology Commission of Shanghai Municipality (14DZ2260800 and 15ZR1410700), Shanghai Collaborative Innovation Center of Trustworthy Software for Internet of Things (ZF1213) and NSFC (61402175).

References

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. In *Proceedings of SemEval*. ACL.

Man Lan, Guoshun Wu, Chunyun Xiao, Yuanbin Wu, and Ju Wu. 2016a. Building mutually beneficial relationships between question retrieval and answer ranking to improve performance of community question answering. In *Neural Networks (IJCNN), 2016 International Joint Conference on*. IEEE, pages 832–839.

Man Lan, Zhihua Zhang, Yue Lu, and Ju Wu. 2016b. Three convolutional neural network-based models for learning sentiment word vectors towards sentiment analysis. In *Neural Networks (IJCNN), 2016 International Joint Conference on*. IEEE, pages 3172–3179.

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

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*. volume 14, pages 1532–1543.

Fabrizio Sebastiani. 2002. Machine learning in automated text categorization. *ACM computing surveys (CSUR)* 34(1):1–47.

Richard Socher, John Bauer, Christopher D Manning, and Andrew Y Ng. 2013. Parsing with compositional vector grammars. In *ACL (1)*. pages 455–465.

Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. Learning sentiment-specific word embedding for twitter sentiment classification. In *ACL (1)*. pages 1555–1565.

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.