IITP at SemEval-2017 Task 8 : A Supervised Approach for Rumour Evaluation

Vikram Singh, Sunny Narayan, Md Shad Akhtar, Asif Ekbal, Pushpak Bhattacharyya
Indian Institute of Technology Patna, India
{vikram.mtcs15,sunny.cs13,shad.pcs15,asif,pb}@iitp.ac.in

Abstract

This paper describes our system participation in the SemEval-2017 Task 8 'RumourEval: Determining rumour veracity and support for rumours'. The objective of this task was to predict the stance and veracity of the underlying rumour. We propose a supervised classification approach employing several lexical, content and twitter specific features for learning. Evaluation shows promising results for both the problems.

1 Introduction

Twitter along with Facebook is widely used social networking site which generates tons of authentic and unauthentic information. The purpose of twitter varies from people to people. Twitter has been greatly used as a communication channel and also as an information source (Zhao and Rosson, 2009). However, Twitter like any other social media platform does not always poses authentic information. It also brings a negative by-product called rumour (Castillo et al., 2011; Derczynski and Bontcheva, 2014; Qazvinian et al., 2011). Rumours are the statement which cannot be verified for its correctness. These rumours may confuse people with the unverified information and drive them in poor decision making. In many organizations(political, administration etc.), detection and support for rumour invites great interest from the concerned authorities.

Recently, researchers across the globe have started addressing the challenges related to rumours. A time sequence classification technique has been proposed for 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. A study on speech act classifier for veracity prediction is proposed in (Vosoughi, 2015). One of the earlier work reported on rumour detection and classification had used twitter specific and content based features for the prediction (Qazvinian et al., 2011).

In this paper we present our proposed system submitted as part of the SemEval-2017 shared task on "RumourEval: Determining rumour veracity and support for rumours". Our system is supervised in nature and uses a diverse set of features (c.f. Section 2.3) for training. The task involves Twitter conversation thread where for every source tweet a number of direct and nested reply tweets are present. An example thread is depicted in Table 1. The task defines two separate sub-problems: A) **S**upport, **D**eny, **Q**uery & **C**omment (SDQC) classification and B) veracity prediction. The first subtask checks the stance of any tweet(source or reply) w.r.t. the underlying rumour. Reply tweet can be direct or nested. Second subtask predicts the veracity of a rumour i.e. *true (rumour)*, *false (not rumour)* or *unverified (its veracity cannot be verified)*. Further, there were two variants of the veracity task: closed and open variants. In closed variant, the veracity prediction has to be made solely from the tweet text only. In addition usage of extra data (Wikipedia article, news article etc.) was allowed for the open variant.

The rest of the paper is organized as follows: Section 2 presents a brief description of the proposed approach. Experimental results and discussion is furnished in Section 3. Finally, we conclude in Section 4.

2 System Overview

We adopted a supervised classification approach for both the tasks. We use Decision Tree (DT), Naive Bayes (NB) and Support Vector Machine

Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017), pages 497–501,
Vancouver, Canada, August 3 - 4, 2017. ©2017 Association for Computational Linguistics

Tweet conversation thread	Stance
Src: Very good on #Putin coup by @CoalsonR: Three Scenarios For A Succession In Russia http://t.co/fotdqxDfEV	Support
Rep1: @andersostlund @CoalsonR @RFERL And how Europe will behave in such a case?	Deny
Rep2: @andersostlund @RFERL Putin'll be made a tsar (and the newborn an heir). Back 2 serfdom as Zorkin suggested.	Comment
Rep3: @andersostlund @CoalsonR @RFERL uhmmm botox sesions far more likely anyway	Comment
Rep4: @andersostlund What are your thoughts on #WhereIsPutin?	Query
Rep5: @tulipgrrl Either a simple flue, more serious illness or serious domestic political problems.	Comment
Rep6: @andersostlund @tulipgrrl :mask:	Deny

Table 1: Twitter conversational thread. Src: Source tweet; Rep#: Replies.

(SVM) as base classifier for prediction of veracity. For stance detection, every instance consists of a pair of source-reply tweet. We extracted features for both the tweets and fed it to the system for the classification. In subsequent subsections we describe dataset, preprocessing and list of features that we use in this work.

2.1 Dataset

The training dataset consists of 272 source tweets for which 3966 replies tweet are present. For tuning the system, validation set contains 256 replies across 25 source tweets. Each source and reply tweet has one of the four label for stance detection namely, *support*, *deny*, *query* and *comment*. For veracity prediction, each of the source tweets belongs to one of the three classes i.e. *true*, *false* and *unverified*. The gold standard test dataset has 28 source and 1021 reply tweets. A detailed statistics is depicted in Table 2.

2.2 Preprocessing

The distribution of different classes in the dataset is very skewed so the first step that we perform is to extract and over sample the under represented class. Classes *support*, *deny* and *comment* were sampled by a factor of 4, 7 and 7 respectively. Afterwards, we perform normalization of urls and usernames in which all urls and username were replaced by keyword someurl and @someuser respectively.

2.3 Features

In this section we describe features that we employed for building the system. We use following set of features for both Subtask A and B.

- **Word Embedding**: Word vectors has been proved to be an efficient technique in capturing semantic property of a word. We use 200-dimension pretrained GloVe model[1] for computing the word embeddings. Sentence embedding is computed by concatenating embeddings of all the words in a tweet. We fix the length of each tweet by padding it to the maximum number of tokens.

- **Vulgar words**: Conversations on Twitter are usually very informal and usage of vulgar words are often in practice. The presence of vulgar words in a sentence declines the orientation of it being a fact, hence, less chances of it being a rumour. We use a list of vulgars words[2] [3] and define a binary feature that takes a value '1' if a token is present in the list, otherwise '0'.

- **Twitter specific features**: We use presence and absence of following twitter specific features in this work.
  - **URL and Media**: The presence of metadata indicates that the user is providing with more authentic information. Hence less chances of it being a rumour. For subtask A, a user reply with metadata suggests it to be a *support* or *deny*.
  - Punctuation, Emoticons and Abbreviation.

- **Word count** : Rumour sentences tend to be more elaborative and hence longer while factual data is generally short and precise. Also, user tends to deny a claim in shorter sentence. We, therefore, define number of words in a sentence (excluding stop words and punctuations), as a feature.

- **POS tag**: We use unigram and bigram POS tags extracted from CMU's ARK[4] tool.

In addition, we also implement few of the task specific features listed below. **Subtask A: SDQC**

[1] http://nlp.stanford.edu/data/glove.6B.zip

[2] http://fffff.at/googles-official-list-of-bad-words/

[3] http://www.noswearing.com/dictionary

[4] http://www.cs.cmu.edu/ ark/TweetNLP/

Dataset	Overall		Subtask A: SDQC				Subtask B: Veracity		
	Source	Reply	Support	Deny	Query	Comment	True	False	Unverified
Train	272	3966	841	333	330	2734	127	50	95
Dev	25	256	69	11	28	173	10	12	3
Test	28	1021	94	71	106	778	8	12	8

Table 2: Distribution of source and reply tweets with their labels in the dataset

- **Negation words**: Presence of negation word in a tweet signals it to be a denial case. Therefore, we use a binary feature indicating the presence of negation words in the tweet. There were 27 negation words taken into account. The following are the list - no, not, nobody, nothing, none, never, neither, nor, nowhere, hardly, scarcely, barely, don't, isn't, wasn't, shouldn't, wouldn't, couldn't, doesn't, hasn't, haven't, didn't, ain't, can't, doesn't and won't.

- ***Wh-* words:** Query usually contains *Wh*-words (What, Where, Which, When, Who, Why, Whom, Whose). We define a binary feature that fires when a tweet contains one of these words.

Subtask B: Veracity prediction

- **Presence of Opinion words**: An opinion carrying sentence cannot be a fact, hence, makes it a probable candidate for rumour. We define two features based on MPQA subjectivity lexicon (Wilson et al., 2005). The first feature takes opinion word count, whereas, the second feature checks the presence of at least one strongly subjective token in a tweet.

- **Number of adjectives**: An interesting relation between presence of adjectives in a sentence and its subjectivity has been explored in (Hatzivassiloglou and Wiebe, 2000). As per (Wiebe, 2000) the probability of a sentence being subjective, given that there is at least one adjective in the sentence, is 0.545. If a sentence is objective then its chances of being a rumour is very low. Therefore, we use a binary feature that denotes presence/absence of adjectives in a tweet.

Since, prediction in close variants has the limitation of using the tweet only, we also extracted 'presence of media' as a binary feature value for the open variant only.

3 Experiments and Results

We use scikit learn machine learning package[5] for the implementation. As defined by shared task, we use classification accuracy and micro-average accuracy as evaluation metrics for SDQC and veracity prediction respectively. For subtask A, we try various feature combinations to train a SVM classifier. Table 3 reports the validation accuracy for SDQC subtasks. As a result we select the feature combination that performs best during the validation phase and submit it for the final prediction on the test dataset. In veracity prediction task, we employ three classifiers i.e. Decision Tree, SVM and Naive Bayes to the evaluate our system. We observe that the among three classifiers performance of Naive Bayes is comparatively better than others as shown in Table 4. For evaluation of test dataset we use our best classifier i.e. Naive Bayes. Our system reports an accuracy of 64.1% for the SDQC classification. For subtask B, we also compute a confidence score for each prediction. We obtain micro-average accuracies of 39.28% and 28.57% respectively for the open and close variants. Reported root mean squared error (RMSE) for the two variants are 0.746 and 0.807. It should be noted that we were the only team which submit-

	Features	Accuracy
A.	Unigram	54.2969%
B.	Unigram + POS	62.1093%
C.	W.E.	61.3281%
D.	(C + POS)	63.2813%
E.	(D + URL and Media)	62.8906%
F.	(E + Twitter Specific)	63.2813%
G.	(F + Negation words)	63.2813%
H.	(G + Wh-Word)	63.6719%
I.	**(H + Vulgar words)**	**64.0625%**
J.	(I + Punctuation)	63.6719%
K.	**(J + Word count)**	**64.0625%**

Table 3: SDQC: Accuracy on Development Set

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

ted their system in open variant category. Table 5 depicts the evaluation result on test dataset.

Classifiers	**Micro-average Accuracy**	
	Open	**Closed**
Decision Tree	58.23%	54.54%
SVM	58.75%	59.09%
Naive Bayes	59.09%	**63.0%**

Table 4: Veracity: Accuracy on Development Set

Task	**Accuracy**	**RMSE**
Subtask A	64.1%	-
Subtask B(Open)	39.28%	0.746
Subtask B(Closed)	28.57%	0.807

Table 5: Evaluation results on test set.

Further, we perform error analysis on the results. Confusion matrix for SDQC classification is depicted in Table 6. We observe that most of the classes were confused with the *comment* class. The possible reason could be the presence of relatively high number instances for the comment 'class'. Similarly, Table 7 & 8 shows confusion matrix for both open and closed variants of subtask B. Recall for 'true' is encouraging i.e. 75% but the problem lies with the precision which is merely 28% & 25% for open and close variants respectively.

	Support	**Deny**	**Query**	**Comment**
Support	**42**	2	2	48
Deny	11	**9**	2	49
Query	9	7	**35**	55
Comment	125	35	32	**586**

Table 6: SDQC: Confusion Matrix on test set (S: support, D: deny, Q: query, C: comment)

	True	**False**	**Unverified**
True	**6**	2	0
False	7	**5**	0
Unverified	8	0	**0**

Table 7: Veracity (Open): Confusion Matrix on test set

	True	**False**	**Unverified**
True	**6**	2	0
False	10	**2**	0
Unverified	8	0	**0**

Table 8: Veracity (Closed): Confusion Matrix on test set

4 Conclusion

In this paper we proposed a supervised approach for determining the support and veracity of a rumour as part of the SemEval-2017 shared task on rumour evaluation. As base classification algorithm we use Naive Bayes, Support Vector Machine and Decision Tree for building the model. In future, we would like to explore deep learning technique and other relevant features to further improve the performance of the system.

References

Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. 2011. Information credibility on twitter. In *Proceedings of the 20th international conference on World wide web*. ACM, pages 675–684.

Leon Derczynski and Kalina Bontcheva. 2014. Pheme: Veracity in digital social networks. In *UMAP Workshops*.

Vasileios Hatzivassiloglou and Janyce M Wiebe. 2000. Effects of adjective orientation and gradability on sentence subjectivity. In *Proceedings of the 18th conference on Computational linguistics-Volume 1*. Association for Computational Linguistics, pages 299–305.

Michal Lukasik, PK 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 54th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 393–398.

Vahed Qazvinian, Emily Rosengren, Dragomir R Radev, and Qiaozhu Mei. 2011. Rumor has it: Identifying misinformation in microblogs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 1589–1599.

Soroush Vosoughi. 2015. *Automatic detection and verification of rumors on Twitter*. Ph.D. thesis, Massachusetts Institute of Technology.

Janyce Wiebe. 2000. Learning subjective adjectives from corpora. In *AAAI/IAAI*. pages 735–740.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*. Association for Computational Linguistics, pages 347–354.

Dejin Zhao and Mary Beth Rosson. 2009. How and why people twitter: the role that micro-blogging plays in informal communication at work. In *Proceedings of the ACM 2009 international conference on Supporting group work*. ACM, pages 243–252.

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. *arXiv preprint arXiv:1609.09028* .