NileTMRG at SemEval-2017 Task 8: Determining Rumour and Veracity Support for Rumours on Twitter

Omar Enayet and Samhaa R. El-Beltagy
Center for informatics sciences
Nile University
Giza, Egypt
omar.enayet@gmail.com, samhaa@computer.org

Abstract

This paper presents the results and conclusions of our participation in SemEval-2017 task 8: Determining rumour veracity and support for rumours. We have participated in 2 subtasks: SDQC (Subtask A) which deals with tracking how tweets orient to the accuracy of a rumourous story, and Veracity Prediction (Subtask B) which deals with the goal of predicting the veracity of a given rumour. Our participation was in the closed task variant, in which the prediction is made solely from the tweet itself. For subtask A, linear support vector classification was applied to a model of bag of words, and the help of a naïve Bayes classifier was used for semantic feature extraction. For subtask B, a similar approach was used. Many features were used during the experimentation process but only a few proved to be useful with the data set provided. Our system achieved 71% accuracy and ranked 5th among 8 systems for subtask A and achieved 53% accuracy with the lowest RMSE value of 0.672 ranking at the first place among 5 systems for subtask B.

1 Introduction

Over the past 15 years, and in a gradual manner, social media has started to become a main source of news. However, social media has also become a ripe ground for rumours, spreading them in a matter of a few minutes. A rumour is defined as a claim that could be true or false. False rumours may greatly affect the social, economic and political stability of any society around the world, hence the need for tools to help people, especially journalists, analyze the spread of rumours and their effect on the society as well as determine their veracity.

Twitter is a famous social media platform capable of spreading breaking news, thus most of rumour related research uses Twitter feed as a basis for research.

SemEval (Semantic Evaluation) is an ongoing series of evaluations of computational semantic analysis systems. Task 8 (RumourEval) (Derczynski, et al. (2017)) is one of 12 tasks presented in SemEval 2017. This paper describes the system that we have used to participate in this task. The task consists of 2 subtasks: SDQC (Subtask A) which has the objective of tracking how other tweets orient to the accuracy of a rumourous story, and Veracity Prediction (Subtask B) for which has the goal to predict the veracity of a given rumour. Task B has two variants: an open variant and a closed one. We have only participated in the closed variant, in which the prediction should be made solely from the tweet itself.

Scientific literature related to rumours on social media has started to emerge over the past 7 years. It can be categorized into 4 main categories: 1) the detection of the spreading of a rumour, 2) the determination of the veracity of a rumour, 3) the analysis of the rumour propagation through a social network and 4) speech act analysis of different online replies to the rumour. Subtask A belongs to the 4th category, while subtask B belongs to the 2nd category.

The rest of the paper is organized as follows: section 2 briefly overviews related work, section 3 provides task description details, section 4 provides a detailed system description covering preprocessing, feature extraction and selection, learning model and evaluation done for both subtasks A and B. In the end a conclusion is given with the future work needed.

Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017), pages 470–474,
Vancouver, Canada, August 3 - 4, 2017.

2 Related Work

Zubiaga et al. (2016), presented a methodology that enabled them to collect, identify and annotate a big data set of rumours associated with multiple newsworthy events, and analyzed how people orient to and spread rumours in social media. This data set was used for task 8 of SemEval 2017: RumourEval. Qazvinian et al. (2011), addressed the problem of automatic rumour detection in microblogs as well as identifying users that support or deny or question the rumour. They achieved this by exploring the effectiveness of 3 categories of features: content-based, network-based and micro-blog specific memes. Vosoughi et al. (2015), addressed the problem of rumour detection via a speech act classifier that detects assertions using different semantic and syntactic features in addition to a clustering algorithm to cluster groups of rumourous tweets talking about the same topic together. Hamidiain et al. (2015), Castillo et al. (2011), Vosoughi et al. (2015) and Giasemidis et al. (2016) addressed the issue of detecting the veracity of rumours using manually selected and annotated rumours on Twitter using linguistic, user, rumour, pragmatic, content, twitter-specific and propagation features and the latter developed a software demonstration that provides a visual user interface to allow the user to examine the analysis. Chua et al. (2016), concentrated on linguistic features such as comprehensibility, sentiment and writing style to predict rumour veracity, ignoring all non-linguistic features. Galitsky et al. (2015), also concentrated on linguistic features to detect disinformation by comparing the text to search results using the significant sentences in that text. Liu et al. (2015), proposed the first real time rumour debunking algorithm for Twitter while Zhao et al. (2015), concentrated on identifying a trending rumour as early as possible without trying to assess its veracity.

3 Task Description

Below is a brief overview of each subtask. For more details, please refer to RumourEval (Derczynski, et al. (2017))

Subtask A: The input to this task is a set of tweets each replying to a rumourous tweet, which we name the rumour source tweet. The training data is composed of the tweet content and its speech act class. A tweet can be classified to be a support, deny, query or a comment.

Subtask B: The input to this task is a set of tweets each representing a source of a rumour. The training data is composed of the tweet content and its veracity. A tweet's veracity can be either true, false or unknown. Also, a confidence value which is a float from 0 to 1 is required for each tweet.

4 System Overview

The systems used for both subtasks A and B were very similar, except that each focused on a different set of features. Python libraries scikit-learn (Buitinck et al. (2013)) and NLTK (Bird et al., 2009) were mainly used to implement this work. Below are the general system specifications. All classifiers were adjusted to use their default parameters.

4.1 Preprocessing and feature extraction

The system depends on performing some preprocessing on the tweets' texts, extracting simple bag of words features from them, and then extracting additional higher level features from them as well as from the entire twitter feed provided. These steps were carried out with the aid of the NLTK Tweet Tokenizer (2015).

Preprocessing also included the removal of stop words, punctuation characters and twitter specific words such as 'rt' and 'via'.

No further pre-processing was performed. Below are some notes in this context:

- The case of the words could be useful in showing the sentiment and the context of the word, thus all words were kept in their original case.
- Performing stemming or lemmatization caused worse performance as keeping the word in its original form proved to be useful.
- Removing URLs from text yielded worse performance, as tweets using the same URL usually shared the same speech act, so the URL word token acted as an important feature.
- Using bi-grams resulted in noise being added to the training data, causing the classifier's performance to degrade.

4.2 Features Selection

The following feature selection and dimensionality reduction methods were used on the basic bag of words features, before adding higher level features:

- Chi-Squared Feature selection. (2010)
- Variance Threshold Feature selection. (2010)
- Truncated SVD dimensionality reduction. (2010)

None of the above algorithms were used, as they all yielded worse results when the model was cross-validated. We attribute this to the fact that the number of features were not big enough.

Additional features were manually selected by measuring the performance of the classifier on training data when adding/removing each additional feature. Features with big numerical values were scaled down to the range between 0 and 1.

4.3 Additional features extraction

Several features were extracted though not all of them proved useful in the classification process.

Below is the complete list of features which apply for **both subtask A and B**.

- **Question Existence:** The relationship between a question and a query is tight. A question is often a query and a query is often a question. Also, if a tweet is a question, then it is highly unlikely that it is a normal comment; it is more likely it is a support or a denial, if not a query. Thus, being a question is an important feature to consider. A question detection module was built for this purpose. Below are details for this module:
  - An assumption was made that any sentence containing a question mark is considered a question.
  - In case the question mark was absent, any sentence classified as a question should contain at least one of the following keywords used in WH-questions: "what, why, how, when, where" or in Yes/No questions: "did, do, does, have, has, am, is, are, can, could, may, would, will" as well as their negations. It is highly unlikely that a question does not have one of these words.
  - A utility classifier was used for further detection of questions; we performed speech act recognition using a Naive Bayes classifier on NLTK corpus 'nps_chat' (2015). On cross validating that classifier, we got an accuracy of 67%. If this utility classifier marks the tweet as a Yes-No question or a wh-question, the tweet is considered to be a question.
- **Denial term detection:** We found that explicitly specifying the existence of a denial word within a tweet, to be a useful feature for generalizing over the data. The list of words used are: 'not true', 'don't agree', 'impossible', 'false', 'shut'.
- **Support words detection:** Like denial words, we included another feature for signaling the existence of a support term. These were detected based on the following list of common support words: 'true', 'exactly', 'yes', 'indeed', 'omg', 'know'
- **Hashtag Existence**
- **URL Existence**
- **Tweet is reply:** This feature specifies whether the tweet was a reply to another tweet or whether it is a source tweet. Source tweets are rarely queries, and not often a denial or support. Most of them are normal comments.
- **Tweet's words' sentiments:** Simple sentiment prediction was performed on each tweet's text though counting the number of positive and negative sentiment in the tweet using the NLTK opinion lexicon (2015). If the positive words exceeded negative words, the feature got a value of 1, otherwise, it got a value of 0. If there were no sentiment words o or if the positive and negative words were equal, this feature value was set to a 0.5.
- **Tweet sentiment:** A utility classifier was used for further detection of sentiment. For setting this feature, a naïve Bayes classifier was trained using the NLTK movie reviews corpus (2015) for sentiment analysis. It would be better of course to train this classifier using tagged tweets, which is what we intend to do in future work.
- **Is User verified**
- **Number of followers**
- **Number of user's past tweets**
- **Number of user's friends**
- **Retweet Ratio:** This feature represents the ratio between the numbers of retweets of the

target tweet over the number of retweets of the rumour source tweet.

- **Photo Existence**
- **Days since user creation:** This feature represents the number of days since user account was created on Twitter. Older accounts may have more credibility than new ones.
- **Source tweet user is verified:** This feature represents whether the tweeter of the rumour source has a verified account or not.

The following list of features applies to **subtask A only**:

- **User 'replied to' is verified**
- **Cosine similarity with root rumourous tweet and the 'replied to' tweet:** Using the same words may imply more that the tweet is a support.

Finally, the following features applies to **subtask B only**:

- **Percentage of replying tweets classified as queries, denies or support:** These 3 features represent the percentage of tweets classified as different classes via the system implemented for Task A, for this rumour's source tweet.

5 Evaluation

Several scikit-learn classifiers were used during experimentation before deciding on the final model.

For subtask A, the linear support vector machine classifier (Linear SVC) proved to be the most accurate during cross validation, however, logistic regression generalized the best on test data. During cross-validation the macro-averaged F1 measure was used to evaluate the classifiers and choose the best amongst them, as the distribution of categories was clearly skewed towards comments.

For subtask B, Linear SVC proved to be the best in terms of accuracy and the confidence root mean square error (RMSE).

Table 1 shows the features used for each subtask along with its type. Type 'Content' refers to the features determined from the tweet's text, 'user' refers to the features determined from the user who tweeted and his behavior, 'twitter' refers to twitter specific features used. Table 2 compares the accuracy of different classifiers for each subtask.

Feature	Type	A	B
Question Existence	**content**	**Y**	N
Denial term detection	**content**	**Y**	N
Support words detection	**content**	**Y**	N
Hashtag existence	**twitter**	N	**Y**
URL existence	**content**	**Y**	**Y**
Tweet is reply	**twitter**	**Y**	-
Tweets' words' sentiments	**content**	**Y**	N
Tweet sentiment	**content**	N	N
Is User Verified	**user**	N	N
Number of followers	**user**	**Y**	N
Number of user's past tweets	**user**	N	N
Number of user's friends	**user**	N	N
Retweet Ratio	**twitter**	N	N
Photo Existence	**content**	N	N
Days since user creation	**user**	N	N
Source tweet user is verified	**user**	**Y**	N
User 'replied to' is verified	**user**	**Y**	-
Cosine similarity with root rumourous tweet	**content**	**Y**	-
Cosine similarity with the 'replied to' tweet	**content**	N	-
Percentage of replying tweets classified as queries, denies or support	**content**	-	**Y**

Table 1 – Features found useful for each subtask and used in final evaluation.

Classifier	A	B	B(RMSE)
Linear SVC	0.71	**0.53**	**0.67**
Random Forest	0.75	0.39	0.77
Linear SVM with SGD learning	0.72	0.5	0.73
Logistic Regression	**0.76**	0.53	0.71
Decision Tree	0.71	0.46	0.73

Table 2 – The resultant accuracy and confidence RMSE for subtasks A and B

6 Conclusion

In this paper, we have performed a quick analysis of using different pre-processing, features extraction and selection, learning classifiers which achieved good results in the RumourEval task. For subtask A, a combination of different types of content, twitter and user specific features were used. For subtask B, it was clear that only content and twitter features were useful. User based features didn't enhance the performance for the latter subtask, thus we conclude that the identity and behavior of the user didn't affect much the credibility of the rumour he/she is spreading, at least for the data set provided.

7 Future Work

Additional features could be extracted that can play a better role in classifying each tweet or rumour. On the tweet text level, better linguistic features could be extracted. A better sentiment analysis model could be employed. On the rumour level, network-based features maybe extracted such as the work done by Vosoughi, et. al. (2015). Time-based analysis could be performed to detect certain patterns in the change of reactions to the rumour.

8 References

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

Zubiaga, A., Liakata, M., Procter, R., Hoi, G.W.S. and Tolmie, P., 2016. Analysing how people orient to and spread rumours in social media by looking at conversational threads. PloS one, 11(3), p.e0150989.

Vosoughi, S., 2015. Automatic detection and verification of rumors on Twitter (Doctoral dissertation, Massachusetts Institute of Technology).

Qazvinian, V., Rosengren, E., Radev, D.R. and Mei, Q., 2011, July. 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.

Giasemidis, G., Singleton, C., Agrafiotis, I., Nurse, J.R., Pilgrim, A., Willis, C. and Greetham, D.V., 2016, November. Determining the veracity of rumours on Twitter. In International Conference on Social Informatics (pp. 185-205). Springer International Publishing.

Castillo, C., Mendoza, M. and Poblete, B., 2011, March. Information credibility on twitter. In Proceedings of the 20th international conference on World wide web (pp. 675-684). ACM.

Hamidiain, S. and Diab, M., 2015. Rumor detection and classification for twitter data. In The Fifth International Conference on Social Media Technologies, Communication, and Informatics, SOTICS, IARIA (pp. 71-77).

Liu, X., Nourbakhsh, A., Li, Q., Fang, R. and Shah, S., 2015, October. Real-time rumor debunking on twitter. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (pp. 1867-1870). ACM.

Zhao, Z., Resnick, P. and Mei, Q., 2015, May. Enquiring minds: Early detection of rumors in social media from enquiry posts. In Proceedings of the 24th International Conference on World Wide Web (pp. 1395-1405). ACM.

Galitsky, B., 2015, March. Detecting Rumor and Disinformation by Web Mining. In 2015 AAAI Spring Symposium Series.

Bird, Steven, Edward Loper and Ewan Klein (2009), Natural Language Processing with Python. O'Reilly Media Inc.

NLTK Tweet Tokenizer (2015). Retrieved April 1, 2017, from http://www.nltk.org/api/nltk.tokenize.html

Scikit-learn Chi-Squared Feature Selection, 2010. Retrieved April 1, 2017, from http://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.chi2.html

Scikit-learn Variance Threshold Feature Selection, 2010. Retrieved April 1, 2017, from http://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.VarianceThreshold.html

Scikit-learn Truncated SVD, 2010. Retrieved April 1, 2017, from http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.TruncatedSVD.html

NLTK NPS Chat Corpus Reader, 2015. Retrieved April 1, 2017, from http://www.nltk.org/_modules/nltk/corpus/reader/nps_chat.html

NLTK Opinion Lexicon Corpus Reader, 2015. Retrieved April 1, 2017, from http://www.nltk.org/_modules/nltk/corpus/reader/opinion_lexicon.html

NLTK Categorized Sentences Corpus Reader. Retrieved April 1, 2017, from http://www.nltk.org/_modules/nltk/corpus/reader/categorized_sents.html

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J. and Layton, R., 2013. API design for machine learning software: experiences from the scikit-learn project. arXiv preprint arXiv:1309.0238.

Chua, A.Y. and Banerjee, S., 2016. Linguistic Predictors of Rumor Veracity on the Internet. In Proceedings of the International MultiConference of Engineers and Computer Scientists (Vol. 1).