Stance Classification in Out-of-Domain Rumours: A Case Study around Mental Health Disorders

Ahmet Aker[1,2], Arkaitz Zubiaga[3], Kalina Bontcheva[1], Anna Kolliakou[4], Rob Procter[3,5], Maria Liakata[3,5]

[1] University of Sheffield, UK
[2] University of Duisburg-Essen, Germany
[3] University of Warwick, UK
[4] King's College London, UK
[5] Alan Turing Institute, UK

Abstract. Social media being a prolific source of rumours, stance classification of individual posts towards rumours has gained attention in the past few years. Classification of stance in individual posts can then be useful to determine the veracity of a rumour. Research in this direction has looked at rumours in different domains, such as politics, natural disasters or terrorist attacks. However, work has been limited to in-domain experiments, i.e. training and testing data belong to the same domain. This presents the caveat that when one wants to deal with rumours in domains that are more obscure, training data tends to be scarce. This is the case of mental health disorders, which we explore here. Having annotated collections of tweets around rumours emerged in the context of breaking news, we study the performance stability when switching to the new domain of mental health disorders. Our study confirms that performance drops when we apply our trained model on a new domain, emphasising the differences in rumours across domains. We overcome this issue by using a little portion of the target domain data for training, which leads to a substantial boost in performance. We also release the new dataset with mental health rumours annotated for stance.

Keywords: social media, stance classification, veracity, rumours, mental health

1 Introduction

Social media are known to be rife with rumours, where along with the circulation of valuable information and fresh news, users also post about and spread information that is yet to be verified [22]. Twitter has become one of the main online platforms to access information that is updated in real time. However, the fact that it is not moderated and anyone can post and share tweets gives rise to rumours [18]. An approach that is increasingly being used to alleviate the effect of rumours is stance classification, which aims to determine the stance

of individual posts discussing a rumour, defined as a classification problem that classifies each post as supporting, denying, querying or commenting on the rumour. While stance classification has been increasingly studied in recent years, previous work assumes that sufficient training data is available in the target domain, and therefore have trained and tested in the same domain.

Previous research on rumour stance classification for tweets has mostly focused on rumours about politics, natural disasters or terrorist attacks [16, 18, 7, 13, 21, 15, 23]. The fact that it is relatively easy to retrieve large amounts of data for these domains has enabled them to use in-domain data for training. However, one may not be able to retrieve as much training data for more obscure domains for which collection of data is harder. Here we document our work on performing rumour stance classification in the domain of mental health disorders, where the dearth of sufficient training data required us to look into the use of out-of-domain data for training. Leveraging out-of-domain rumour data within the context of breaking news, available from previous work, we study different classifiers to determine the stance of tweets discussing rumours in the context of mental health disorders, looking particularly at a rumoured case of depression that led to a pre-meditated plane crash. Our study contributes with analyses about how classifiers trained on out-of-domain data perform on mental health rumours where the shortage of training data makes it more difficult to build a model for the classification. We also investigate alternative ways of boosting the performance by adding a proportion of the testing data into the training process. Our results show that the domain switch from breaking news to mental health is bound with a performance loss when it comes to rumours. However, the addition of a small proportion of the mental health data to the training process leads to remarkable improvements.

2 Related Work

One of the pioneering studies in this task is reported by Mendoza et al. [16]. In this study they have manually looked into rumours with established veracity levels to understand the stance Twitter users take with respect to true and false rumours. They analysed seven rumours which were later proven true and seven rumours which had been debunked. They manually labelled the tweets with the stance categories "affirms" (supports), "denies" and "questions". They showed encouraging results correlating stance and veracity, with 95% of the tweets associated with true rumours labelled as "affirming", and 38% of the tweets associated with false rumours labelled as "denying".

The first study that tackles the stance classification automatically is reported by Qazvinian et al. [18]. With a dataset containing 10K tweets and using a Bayesian classifier and three types of features categorised as "content", "network" and "Twitter specific memes", the authors achieved an accuracy of 93.5%. Similar to them, Hamidian and Diab [7] perform rumour stance classification by applying supervised machine learning using the data set reported by Qazvinian et al. [18]. However, instead of Bayesian classifiers the authors use J48 decision

tree implemented within the Weka platform [6]. The features from Qazvinian et al. [18] are adopted and extended with time related information and hastag itself instead of the content of the hashtag as used by [18]. In addition to the feature categories introduced above Hamidian and Diab [8] introduce another feature category namely "pragramatic". The pragmatic features include named entity, event, sentiment and emoticons. The evaluation of the performance is casted as either 1-step problem containing a 6 class classification task (not rumour, 4 classes of stance and not determined by the annotator) or 2-step problem containing first a 3 class classification task (non-rumour or rumour, not determined) and then 4 class classification task (stance classification). Better performances are achieved using the 2-step approach leading to 82.9% F-1 measure compared to 74% with the 1-step approach. The authors also report that the best performing features were the content based features and the least performing ones the network and twitter specific features. In their recent paper Hamidian and Diab [9] introduce the Tweet Latent Vector (TLV) approach that is obtained by applying the Semantic Textual Similarity model proposed by Guo and Diab [5]. The authors compare the TLV approach to their own earlier system as well as to original features of Qazvinian et al. [18] and show that the TVL approach outperforms both baselines.

Liu et al. [13] follow the resulting investigations about stances in rumours made by Mendoza et al. [16] and use stance as additional feature to those reported by related work to tackle the veracity classification problem. On the stance classification the authors adopt the approach of Qazvinian et al. [18] and compare it with a rule-based method briefly outlined by the authors. They claim that their rule-based approach performed better than the one adopted from related work and thus use the rule-based stance classification as additional component on the veracity problem . The experiments were performed on the data set reported by Qazvinian et al. [18]. Unfortunately the authors do not provide detailed analysis about the performance of their rule-based stance classification.

More recently, Zeng et al. [21] enriches the feature sets investigated by earlier studies by features determined through the Linguistic Inquiry and Word Count (LIWC) dictionaries [20]. They investigate supervised approaches using Logistic Regression, naïve Bayes and Random Forest classification. The authors use their own manually annotated data to classify them by stance. However, unlike previous studies Zeng et al. consider only two classes: affirm and deny. Best results are reported with Random Forest leading to 87% precision, 96.9% recall, 91.7% F1-measure and 88.4% accuracy.

Unlike related work we test all reported machine learning techniques on the same data set. This helps to compare their performance better. In addition, we evaluate the best performing model using out-of-domain data. This gives reliable indication about how portable a model is when used in an unseen environment.

3 Rumour Data

We use two types of datasets, both related and unrelated to mental health:

3.1 Mental health data

For our scenario studying mental health related rumours, we collected a dataset from Twitter during the Germanwings plane crash in March 2015. Following the approach described in Zubiaga et al. [24], we sampled tweets related to the rumour that the co-pilot had been diagnosed with depression, and randomly selected a subset of 31 tweet conversations (tweets discussing a rumour and replies to those) to annotate for stance, amounting to a total of 401 tweets. More details about the different stance distributions are shown in Table 1.

Owing to the small size of this dataset, we opted for obtaining out-of-domain data that would expand the data available for training.

Dataset	Rumours	S	D	Q	C
Health data					
Depression	1	85	67	14	235
Non-health data					
Ottawa shooting	58	161	76	64	481
Ferguson riots	46	192	83	94	685
Germanwings crash	68	177	12	28	169
Charlie Hebdo	74	236	56	51	710
Sydney siege	71	89	4	99	713

Table 1: Counts of tweets with supporting, denying or questioning labels in each event collection from our 6 datasets. S: supporting, D: denying, Q: querying, C: commenting.

3.2 Out-of-domain data

The out-of-domain data is reported by Zubiaga et al. [24], who made it publicly available. The authors identify rumours associated with events, collect conversations sparked by those rumours in the form of replies and annotate each of the tweets in the conversations for stance. These data consist of tweets from 5 different events: Ottawa shooting, Ferguson riots, Germanwings crash, Charlie Hebdo and Sydney siege. Each dataset has a different number of rumours where each rumour contains tweets marked with annotations for stance. These 5 datasets contain a total of 2,758 tweets and each post is annotated as one of "supporting", "questioning", "denying" or "commenting". Different from the mental health data, these 5 datasets are collected in the early stages of breaking news, where rumours are related to the reporting of the event and unrelated to mental health disorders. Examples of rumours in the out-of-domain-data include stories such as "*Suspected shooter has been killed/is dead*" or "*There were three separate shooting incidents*". A summary of the data is given in Table 1.

4 Experimental Setup

In keeping with prior work, our experiments assume that incoming tweets already belong to a particular rumour, e.g. a user is tracking tweets related to a certain rumour.

Using the out-of-domain data we follow two scenarios during training and testing: (1) training and testing are performed on isolated data, i.e. we train our models on *n-1* non-health rumours and test them on the n^{th} non-health rumour, and (2) introducing a proportion of the n^{th} rumour in the training data. In (1) the classifier is trained on all rumours except the one that is used for testing. In (2) the training data is enriched with first 10%, 20%, 30%, 40%, 50% and 60% tweets from the rumour set that builds the testing data i.e. from the n^{th} rumour data. Note that in setting (2) we exclude from the testing data whatever is included in the training data. We use setting (1) to determine the best performing classifier. We use this best classifier and run it using scenario (2) on the non-health rumours.

For the mental health related rumours we use all the non-health rumors to train the classifiers and test them on the health data. However, similar to the above setting (2), we also introduce 10-60% tweets from the health rumours in the non-health training data. Again like above the tweets included from the testing set to the training one are excluded from the testing data. We report performance results in accuracy. However, in some cases accuracy can be biased if there is an unbalanced number of class instances. Therefore we also report results in macroaveraged F1 scores – the harmonic mean between precision and recall, computed first for each class and then averaged across all classes; this enables a complementary evaluation for an imbalanced problem like this.

Classifiers. We experiment with five different classifiers: (1) Support Vector Machines (SVMs) using the RBF kernel [2], (2) the J48 Tree, (3) Random Forests, (4) Naïve Bayes, and (5) an Instance Based classifier.

Features. Prior work on stance classification investigated various features varying from syntactical, semantical, indicator, user-specific, message-specific, etc. types [16, 18, 7, 13, 21, 15, 23]. This paper adopts the features from these papers, coupled with experiments with a wide range of machine learning classifiers. All in all, we use a range of 33 different features, which we describe in detail in Appendix A.

5 Results

On the non-health data we run each of the classifiers using the setting (1). The results are shown in Table 2.

From the results in Table 2 we can see that the worst performing classifier is the SVM and the best the J48. This is the case both in terms of the accuracy and F1 metrics. We think SVM does not perform well because our training data

classifier	accuracy	F1
SVM	64.59	52.13
Random Forest	70.07	62.99
IBk	71.82	72.95
Bayes	73.14	68.83
J48	**75.84**	**73.37**

Table 2: Different classifier performances on setting (1). IBk is the Instance Based Learning classifier. The F1 figures are weighted over the 4 classes (support, deny, question and comment).

is imbalanced in terms of class instances. As shown in Table 1 there are far more commenting instances than the other 3 classes. The J48 Tree is not affected to the same extent by this as it can handle imbalanced data. In the following we use J48 to report detailed results in both non-health and health related data. The results for the non-health rumours for the best performing classifier – J48 Tree – are shown in Table 3.

in domain tweets	accuracy	F1
0%	75.84	73.37
10%	75.66	71.42
20%	76.64	72.67
30%	76.86	73.25
40%	77.5	73.74
50%	78.9	75.64
60%	80.1	76.86

Table 3: Classification results in accuracy and F1 obtained using the J48 Tree.

The row with 0% shown in Table 3 represents the set-up scenario (1) discussed above. From the results we can see inclusion of testing data (instances from the rumour that is under test) in the training improves the results in both accuracy and F1 cases. Furthermore, we can see that the more testing data is included the better is the overall performance (except from 0% to 10%).

The results shown in Table 4 are obtained by the classifiers trained using the entire non-health data without inclusion of any health-rumours. This simulates the scenario of applying a classifier to out-of-domain data. From these results we can see that there is a performance drop of all classifiers when applied on a different domain. In terms of accuracy we can see that the largest drop in performance happens with the Random Forest classifier. The smallest drop can be observed with the SVM and J48 classifiers. In terms of F1 score, again Random Forest is affected with the largest drop. The least affected classifier for the F1 score is the instance based classifier. However, the overall picture is that again

classifier	accuracy	F1
SVM	59.27	46.18
Random Forest	57.89	29.8
IBk	65.22	60.42
Bayes	63.22	58.31
J48	**69.45**	**65.45**

Table 4: Stance classification results for the health rumours in accuracy and weighted F1 over the 4 classes.

J48 is the best performing classifier based on both accuracy and F1 metrics. The worst performing one is the Random Forest classifier.

in domain tweets	accuracy	F1
0%	69.45	65.45
10%	70.77	65.83
20%	72.41	66.79
30%	73.12	67.69
40%	74.67	70.06
50%	72.89	70.08
60%	75.87	73.54

Table 5: Classification results in accuracy and F1 obtained using the J48 Tree.

Using the best performing classifier, the J48 tree, we replicated the (2) scenario experiment with the health data, i.e. 10-60% of health rumours were injected into the entire non-health data for training purposes. The results of this experiment are shown in Table 5. From this table we can see that the inclusion of in-domain data to the training process increases the results gradually (except for 50% for the accuracy case) both for accuracy and F1 measures and so demolishes the performance loss. In fact, these results suggest that inclusion of in-domain data in this scenario with little annotated data is much more crucial than in the non-health scenario; we see for instance that in the non-health scenario the use of 30% in-domain data leads to 1% improvement, while the same amount of in-domain data leads to nearly 4% improvement in the new domain.

Finally, to test the real impact of the out-domain data on the health data we trained and tested our classifier only on the health data. Because of the size of the data we performed 50% till 80% reservations for training purposes and the remaining for testing. Results are shown in Table 6.

From Table 6 we can see that, by performing training and testing on the in-domain data, we achieve significantly lower results than when the training process is augmented with out-domain data. This also shows that the out-domain data has substantial contribution on boosting the results.

in domain tweets	accuracy	F1
50%	44.55	37.45
60%	46.15	41.64
70%	51.02	48.57
80%	59.95	59.27

Table 6: Classification results in accuracy and F1 obtained using the J48 Tree. The results are obtained using only in domain data for training.

6 Conclusions

We have tackled the rumour stance classification task leveraging out-of-domain data for training for the first time. While previous research utilised in-domain data for training in scenarios with large datasets available, such as breaking news, here we studied the classification in more obscure scenarios, which is the case of mental health disorders. We experimented with various classifiers and reported the performance of different classifiers on the same dataset. We also performed experiments on domain transfer, applying models trained from the non-health domain on the health-related rumours. We showed that the best performing classifier is the J48 decision tree. It outperformed all other classifiers on the non-health rumours and also achieved the best results after domain transfer. We also observed that the domain transfer is in general bound with a loss in performance. For instance, J48 dropped from an accuracy of 75% to 69% when switching from the non-health to the health domain. Our results showed that the Random Forest classifier has undergone the worst performance loss among all other methods. However, we also reported that inclusion of some proportion of the in-domain data to the training process helps boost the performance. Finally, we reported training and testing on the in-domain data only and showed that the results are substantially lower compared to the case when the training data is augmented with the out-domain data.

Our rumour stance classifier applied to new, obscure domains with shortage of training data has numerous applications that we aim to explore in the near future, such as rumours around bullying and suicide [10], or disputed perceptions around psychoactive substances [11]. Further improving our classifier, in future work we also aim to perform a more detailed analysis underpinning the reasons for the performance drop. For instance, we plan to investigate the stability of features during a domain transfer.

Acknowledgements

This work was partially supported by the European Union under grant agreement No. 654024 SoBigData, PHEME project under the grant agreement No. 611223 and by the Deutsche Forschungsgemeinschaft (DFG) under grant No. GRK 2167, Research Training Group “User-Centred Social Media”. Rob Procter and Maria Liakata were supported by the Alan Turing Institute.

Bibliography

[1] M. Baroni, G. Dinu, and G. Kruszewski. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of ACL*, pages 238–247, 2014.

[2] M. D. Buhmann. Radial basis functions: theory and implementations. *Cambridge Monographs on Applied and Computational Mathematics*, 12:147–165, 2003.

[3] F. Celli, A. Ghosh, F. Alam, and G. Riccardi. In the mood for sharing contents: Emotions, personality and interaction styles in the diffusion of news. *Information Processing & Management*, 52(1):93–98, 2016.

[4] D. Chen and C. D. Manning. A fast and accurate dependency parser using neural networks. In *Proceedings of EMNLP*, pages 740–750, 2014.

[5] W. Guo and M. Diab. Modeling sentences in the latent space. In *Proceedings of ACL*, pages 864–872. Association for Computational Linguistics, 2012.

[6] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

[7] S. Hamidian and M. Diab. Rumor detection and classification for twitter data. In *Proceedings of SOTICS*, pages 71–77, 2015.

[8] S. Hamidian and M. T. Diab. Rumor detection and classification for twitter data. In *Proceedings of SOTICS*, 2015.

[9] S. Hamidian and M. T. Diab. Rumor identification and belief investigation on twitter. In *Proceedings of NAACL-HLT*, pages 3–8, 2016.

[10] S. Hinduja and J. W. Patchin. Bullying, cyberbullying, and suicide. *Archives of suicide research*, 14(3):206–221, 2010.

[11] A. Kolliakou, M. Ball, L. Derczynski, D. Chandran, G. Gkotsis, P. Deluca, R. Jackson, H. Shetty, and R. Stewart. Novel psychoactive substances: An investigation of temporal trends in social media and electronic health records. *European Psychiatry*, 38:15–21, 2016.

[12] P. Liang. *Semi-supervised learning for natural language*. PhD thesis, Massachusetts Institute of Technology, 2005.

[13] X. Liu, A. Nourbakhsh, Q. Li, R. Fang, and S. Shah. Real-time rumor debunking on twitter. In *Proceedings of CIKM*, pages 1867–1870. ACM, 2015.

[14] M. Lukasik, T. Cohn, and K. Bontcheva. Classifying tweet level judgements of rumours in social media. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015*, pages 2590–2595, 2015.

[15] M. Lukasik, P. K. Srijith, D. Vu, K. Bontcheva, A. Zubiaga, and T. Cohn. Hawkes processes for continuous time sequence classification: an applica-

tion to rumour stance classification in twitter. In *Proceedings of the 54th Meeting of the Association for Computational Linguistics*, pages 393–398. Association for Computer Linguistics, 2016.

[16] M. Mendoza, B. Poblete, and C. Castillo. Twitter under crisis: can we trust what we rt? In *Proceedings of the workshop on social media analytics*, pages 71–79. ACM, 2010.

[17] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[18] V. Qazvinian, E. Rosengren, D. R. Radev, and Q. Mei. Rumor has it: Identifying misinformation in microblogs. In *Proceedings of EMNLP*, pages 1589–1599, 2011.

[19] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of EMNLP*, volume 1631, page 1642. Citeseer, 2013.

[20] Y. R. Tausczik and J. W. Pennebaker. The psychological meaning of words: Liwc and computerized text analysis methods. *Journal of language and social psychology*, 29(1):24–54, 2010.

[21] L. Zeng, K. Starbird, and E. S. Spiro. # unconfirmed: Classifying rumor stance in crisis-related social media messages. In *Proceedings of ICWSM*, 2016.

[22] A. Zubiaga, A. Aker, K. Bontcheva, M. Liakata, and R. Procter. Detection and resolution of rumours in social media: A survey. *arXiv preprint arXiv:1704.00656*, 2017.

[23] A. Zubiaga, E. Kochkina, M. Liakata, R. Procter, and M. Lukasik. Stance classification in rumours as a sequential task exploiting the tree structure of social media conversations. In *Proceedings of COLING*, 2016.

[24] A. Zubiaga, M. Liakata, R. Procter, G. Wong Sak Hoi, and P. Tolmie. Analysing how people orient to and spread rumours in social media by looking at conversational threads. *PLoS ONE*, 11(3):1–29, 03 2016.

A Complete set of features

- **BOW (Bag of words):** For this feature we first create a dictionary from all the tweets in the out-of-domain dataset. Next each tweet is assigned the words in the dictionary as features. For words occurring in the tweet the feature values are set to the number of times they occur in the tweet. For all other words "0" is used.
- **Brown Cluster:** Brown clustering is a hard hierarchical clustering method and we use it to cluster words in hierarchies. It clusters words based on maximising the probability of the words under the bigram language model, where words are generated based on their clusters [12]. In previous work it

has been shown that Brown clusters yield better performance than directly using the BOW features [14]. Brown clusters are obtained from a bigger tweet corpus that entails assignments of words to brown cluster ids. We used 1000 clusters, i.e. there are 1000 cluster ids. All 1000 ids are used as features however only, ids that cover words in the tweet are assigned a feature value "1". All other cluster id feature values are set to "0".

- **POS tag:** The BOW feature captures the actual words and is domain dependent. To create a feature that is not domain dependent we added Part of Speech (POS) tags as additional feature. Similar to the BOW feature we created a dictionary of POS tags from the entire corpus (excluding the health data) and used this dictionary to label each tweet with it – binary, i.e. whether a POS tag is present.[1] However, instead of using just single POS tag we created sequences containing bi-gram, tri-gram and 4-gram POS tags. Feature values are the frequencies of POS tag sequences occurring in the tweet.
- **Sentiment:** This is another domain independent feature. Sentiment analysis reveals the sentimental polarity of the tweet such as whether it is positive or negative. We used the Stanford sentiment[19] tool to create this feature. The tool returns a range from 0 to 4 with 0 indicating "very negative" and 4 "very positive". First, we used this as a categorical feature but turning it to a numeric feature gave us better performance. Thus each tweet is assigned a sentiment feature whose value varies from 0 to 4.
- **NE:** Named entity (NE) is also domain independent. We check for each tweet whether it contains *Person, Organization, Date, Location* and *Money* tags and for each tag in case of presence we add "1" otherwise "0".
- **Reply:** This feature is a binary feature and assigns "1" if the tweet is a reply to a previous one or not and otherwise "0". The reply information is extracted from the tweet metadata. Again this feature is domain independent.
- **Emoticon:** We created a dictionary of emoticons using Wikipedia[2]. In Wikipedia those emoticons are grouped by categories. We use the categories as the feature. If any emoticon from a category occurs in the tweet we assign for that category feature the value "1" otherwise "0". Again similar to the previous features this feature is domain independent.
- **URL:** This is again domain independent. We assign the tweet "1" if it contains any URL otherwise "0".
- **Mood:** Mood detection analyses a textual content using different view points or angles. We use the tool described by [3] to perform the mood detection. This tools looks from 5 different angles to each tweet: amused, disappointed, indignant, satisfied and worried. For each of this angles it returns a value

[1] We also experimented with frequencies of POS tags, i.e. counting how many times a particular POS tag occurs in the tweet. The counts then have been normalized using mean and standard deviation. However, the frequency based POS feature negatively affected the classification accuracy so that we omitted it from the feature set.

[2] https://en.wikipedia.org/wiki/List_of_emoticons

from -1 to +1. We use the different angles as the mood features and the returned values as the feature value.
- **Originality score**: Is the count of tweets the user has produced, i.e. the "statuses count" in the Twitter API.
- **isUserVerified(0-1)**: Whether the user is verified or not.
- **NumberOfFollowers**: Number of followers the user have.
- **Role score**: Is the ratio between the number of followers and followees (i.e. NumberOfFollowers/NumberOfFollowees).
- **Engagement score**: Is the number of tweets divided by the number of days the user has been active (number of days since the user account creation till today).
- **Favourites score**: The "favourites count" divided by the number of days the user has been active.
- **HasGeoEnabled(0-1)**: User has enabled geo-location or not.
- **HasDescription(0-1)**: User has description or not.
- **LenghtOfDescription in words**: The number of words in the user description.
- **averageNegation**: We determine using the Stanford parser [4] the dependency parse tree of the tweet, count the number of negation relation ("neg") that appears between two terms and divide this by the number of total relations.
- **hasNegation(0-1)**: Tweet has negation relationship or not.
- **hasSlangOrCurseWord(0-1)**: A dictionary of key words[3] is used to determine the presence of slang or curse words in the tweet.
- **hasGoogleBadWord(0-1)**: Same as above but the dictionary of slang words is obtained from Google.[4]
- **hasAcronyms(0-1)**: The tweet is checked for presence of acronyms using a acronym dictionary.[5]
- **averageWordLength**: Average length of words (sum of word character counts divided by number of words in each tweet).
- **surpriseScore**: We collected a list of surprise words such as "amazed", "surprised", etc. We use this list to compute a cumulative vector using word2Vec [17] – for each word in the list we obtain its word2Vec representation, add them together and finally divide the resulting vector by the number of words to obtain the cumulative vector. Similarly a cumulative vector is computed for the words in the tweet – excluding acronyms, named entities and URLs. We use cosine to compute the angle between those two cumulative vectors to determine the surprise score. Our word embeddings comprise the vectors published by Baroni et al. [1].
- **doubtScore**: Similar to the *surpriseScore* but use instead a list of doubt words such as "doubt", "uncertain", etc.

[3] www.noswearing.com/dictionary
[4] http://fffff.at/googles-official-list-of-bad-words
[5] www.netlingo.com/category/acronyms.php

- **noDoubtScore**:As in *doubtScore* but use instead words which stand for certainty such as “surely”, “sure”, “certain”, etc.
- **hasQuestionMark(0-1)**: The tweet has “?” or not.
- **hasExclamationMark(0-1)**: The tweet has “!” or not.
- **hasDotDotDot(0-1)**: Whether the tweet has “...” or not.
- **numberOfQuestionMark**: Count of “?” in the tweet.
- **NumberOfExclamationMark**: Count of “!” in the tweet.
- **numberOfDotDotDot**: Count of “...” in the tweet.
- **Binary regular expressions applied on each tweet**: .*(rumor?—debunk?).*, .*is (that—this—it) true.*, etc. In total there are 10 features covering regular expressions.