Simple Open Stance Classification for Rumour Analysis

Ahmet Aker[a,b] and **Leon Derczynski**[a] and **Kalina Bontcheva**[a]
Department of Computer Science, University of Sheffield[a]
Department of Information Engineering, University of Duisburg-Essen[b]
a.aker@is.inf.uni-due.de, leon.derczynski@sheffield.ac.uk
K.Bontcheva@sheffield.ac.uk

Abstract

Stance classification determines the attitude, or stance, in a (typically short) text. The task has powerful applications, such as the detection of fake news or the automatic extraction of attitudes toward entities or events in the media. This paper describes a surprisingly simple and efficient classification approach to open stance classification in Twitter, for rumour and veracity classification. The approach profits from a novel set of automatically identifiable problem-specific features, which significantly boost classifier accuracy and achieve above state-of-the-art results on recent benchmark datasets. This calls into question the value of using complex sophisticated models for stance classification without first doing informed feature extraction.

1 Introduction

Stance detection is the problem of classifying the attitude taken by an author in a short piece of text. Typical stances include showing support, denying, commenting on or querying an existing claim or fact. Knowing the stance that authors hold in response to claims, e.g. in online commentary, gives useful insights. It can reveal rumours and fake news claims as the discourse around them is monitored (Procter et al., 2013). Stance reflects how certain authors are of a claim's veracity (Biber, 2006), which enables the effective detection of potential false rumours (Lukasik et al., 2015). Stance also reveals how online populations react to business and political news.

This paper addresses the general-purpose, or *open* stance classification task. This is distinct from *target-specific* stance classification, as in Augenstein et al. (2016) and Mohammad et al. (2016), which focus on stances towards known, pre-determined targets. In the latter task, the target has already been extracted, from e.g. conversational cues. Target-specific stance classification is suited to situations where the target is already known, such as analyses of a specific product or political actor. In contrast, the open stance classification task is appropriate in emerging news or novel contexts, such as working with online media or streaming news analysis.

Open stance classification is often applied in rumour resolution. Since attitudes in discourse around a claim are indicative not only of the controversiality of the claim, but also can act as a proxy for its veracity, it is reasonable to consider the application of open stance detection for rumour analysis. Indeed, many approaches to rumour and fake news analysis rely on this signal (Derczynski et al., 2017)[1]. In veracity analysis, the claim is already known, and the goal is to gather observations and analyse crowd reaction in order to resolve the claim. Instead of being concerned with specific targets, we apply non-targeted – *open* – stance analysis to messages replying to a claim, where the target may vary but the high-level rumour topic rumour remains the same.

Our simple approach to open stance classification implements common features used in stance classification reported by related work (e.g. bag-of-words, named entities, user activity information, URL presence). We extend this with problem-specific features (which we refer to as the AF features) designed to capture how users react to tweets and express confidence in them. Our results show adding these features gives significantly higher performance on benchmark datasets, compared to recent state-of-the-art systems.

[1] http://approximatelycorrect.com/2017/01/23/is-fake-news-a-machine-learning-problem/

The outline of the paper is as follows. First we describe related work (Section 2) and then introduce our method along with the classification techniques used and features extracted (Section 3). Next, Section 4 describes our experimental setups, followed by results in Section 5. We report on feature analysis in Section 6, prior to concluding the paper (Section 7).

2 Related Work

The first study that tackles automatic stance classification is that of Qazvinian et al. (2011). 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 (2015) perform rumour stance classification by applying supervised machine learning using the dataset created by Qazvinian et al. (2011). However, instead of Bayesian classifiers, the authors use J48 decision tree implemented within the Weka platform (Hall et al., 2009). The features from Qazvinian et al. (2011) are adopted and extended with time-related information and the hastags themselves, instead of the content of the hashtag as used by Qazvinian et al. (2011). In addition to the feature categories introduced above, Hamidian and Diab (2015) 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, rumour, and not determined), followed by a 4 class classification task (stance classification). The two step approach achieves better performance with 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 worst performing ones – the network and Twitter specific features. In their most recent paper, Hamidian and Diab (2016) introduce the Tweet Latent Vector (TLV) approach that is obtained by applying the Semantic Textual Similarity model proposed by Guo and Diab (2012). The authors compare the TLV approach to their own earlier system, as well as to the original features of Qazvinian et al. (2011) and show that the TLV approach outperforms both baselines.

Liu et al. (2015) use a rule-based method and show that it outperforms the approach reported by Qazvinian et al. (2011). Zeng et al. (2016) enrich the feature sets investigated by earlier studies by features derived from the Linguistic Inquiry and Word Count (LIWC) dictionaries (Tausczik and Pennebaker, 2010). Lukasik et al. (2016) investigate Gaussian Processes as rumour stance classifier. For the first time the authors also use Brown Clusters to extract the features for each tweet. Unlike researchers above, Lukasik et al. evalute on the rumour data released by Zubiaga et al. (2016b), where they report an accuracy of 67.7%. This result is achieved when the classifier is trained on $n-1$ rumours and tested on the n^{th} rumour. However, the authors achieve substantially better results when a small proportion of the in-domain data (data from the n^{th} rumour) is also included in the training (68.6% accuracy). Performance scores differ substantially from those in the studies described above, given that Lukasik et al. (2016) tackled classification of stance in new rumours that differ from those in the training set.

Subsequent work has also tackled stance classification for new, unseen rumours. Zubiaga et al. (Zubiaga et al., 2016a) moved away from the classification of tweets in isolation, focusing instead on Twitter 'conversations' (Tolmie et al., 2015) initiated by rumours, as part of the PHEME project (Derczynski and Bontcheva, 2014). They looked at tree-structured conversations initiated by a rumour and followed by tweets responding to it by supporting, denying, querying or commenting on the rumour.

Rumour stance classification for tree structured conversations has also been studied in the RumourEval shared task at SemEval 2017 (Derczynski et al., 2017). Subtask A there consisted of stance classification of individual tweets discussing a rumour within a conversational thread as one of *support*, *deny*, *query*, or *comment*. Eight participanting teams submitted results to this task. Most of the systems viewed this task as a 4-way single tweet classification task, with the exception of the best performing system by Kochkina et al. (2017), as well as the systems by Wang et al. (2017) and Singh et al. (2017). The winning system addressed the task as a sequential classification problem, where the stance of each tweet

takes into consideration the features and labels of the preceding tweets. The system by Singh et al. (2017) takes as input pairs of source and reply tweets, whereas Wang et al. (2017) addressed class imbalance by decomposing the problem into a two step classification task – first distinguishing between comments and non-comments and then classifying non-comment tweets as one of support, deny or query. Half of the systems employed ensemble classifiers, where classification was obtained through majority voting (Wang et al., 2017; García Lozano et al., 2017; Bahuleyan and Vechtomova, 2017; Srivastava et al., 2017). In some cases the ensembles were hybrid, consisting both of machine learning classifiers and manually created rules, with differential weighting of classifiers for different class labels (Wang et al., 2017; García Lozano et al., 2017; Srivastava et al., 2017). Three systems used deep learning, with Kochkina et al. (2017) employing LSTMs for sequential classification, Chen et al. (2017) using convolutional neural networks (CNN) for obtaining the representation of each tweet, assigned a probability for a class by a softmax classifier and García Lozano et al. (2017) using CNN as one of the classifiers in their hybrid conglomeration. The remaining two systems by Enayet and El-Beltagy (2017) and Singh et al. (2017) used support vector machines with a linear and polynomial kernel respectively.

3 Method

3.1 Data

In our experiments we used two different data sets: RumourEval dataset (Derczynski et al., 2017) and the PHEME dataset (Zubiaga et al., 2016b). In the PHEME dataset 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 stance annotations: "supporting", "questioning", "denying" or "commenting". A summary of the data is given in Table 1.

The RumourEval dataset is derived from the PHEME dataset, however, for the purpose of the RumourEval shared Task A the data has a given split into training and testing. This provides an established basis for evaluation. The training data draws from stories in 2014–2016, from the earlier PHEME dataset. The evaluation split covers two new stories, both from 2016: first, the disappearance of Marina Joyce, a British Youtube personality, who was rumoured to have been abducted in July 2016. There was significant speculation in social media, and the case was brought to a concrete resolution as the police investigated and posted an open public response. The second story was that Hillary Clinton had pneumonia during mid-September 2016. The prevalence and spread of this story could be tracked easily, and it emerged in a short space of time, though among background noise of speculative, unsubstantiated claims about her and her opponent's health. More details about this dataset can be obtained from the SemEval website[2].

Dataset	Rumours	S	D	Q	C
Ottawa shooting	58	161	76	64	481
Ferguson riots	46	192	83	94	685
Charlie Hebdo	74	236	56	51	710
Sydney siege	71	89	4	99	713

Table 1: PHEME Data: Counts of tweets with supporting (S), denying (D), questioning (Q) and commenting (C) labels in each event collection.

In keeping with prior work (Zeng et al., 2016; Lukasik et al., 2016; Zubiaga et al., 2016a), our experiments assume that incoming tweets already belong to a particular rumour, e.g. a user is tracking tweets related to a certain rumour. For each new tweet, features are extracted into a feature vector, which is then used to assign each tweet its stance towards the rumour.

3.2 Classifiers

We experiment with three different, well known machine learning classifiers: (1) a decision tree, J48; (2) Random Forests (Breiman, 2001); and (3) an Instance Based classifier (K-NN). For the Random Forest we use 50 trees (*-I 50*). Pruning is enabled for J48. Finally we run the Instance Based classifier with *-I -K 10* settings.

3.3 Features

Prior work on stance classification investigated various features which can be categorized into linguistic, message-based, and topic-based cate-

[2]http://alt.qcri.org/semeval2017/task8/

gories (Mendoza et al., 2010; Qazvinian et al., 2011; Hamidian and Diab, 2015; Liu et al., 2015; Zeng et al., 2016; Lukasik et al., 2016; Zubiaga et al., 2016a). The following list summarizes the features adopted in this work.

- **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 (Liang, 2005). In previous work, it has been shown that Brown clusters yield better performance than directly using the BOW features (Lukasik et al., 2015). 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.[3] However, instead of using just single POS tags, 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(Socher et al., 2013) 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 present, "1" is added, or a "0" otherwise.
- **Reply:** This is a binary feature, which is assigned "1" if the tweet is a reply to a previous one, or a "0" otherwise. Tweet reply information is extracted from the tweet metadata. Again this feature is domain independent.
- **Emoticon:** We created a dictionary of emoticons using Wikipedia[4]. In Wikipedia those emoticons are grouped by categories, which we use as a 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, or "0" otherwise.
- **Mood:** Mood detection analyses textual content using different view points or angles. Mood detection is performed using the tool from (Celli et al., 2016), which analyses tweets from five different angles: amused, disappointed, indignant, satisfied and worried. For each of this angles it returns a value from -1 to +1. We use the different angles as the mood features and the returned values as the feature value.
- **Originality score**: This is the count of tweets the user has produced, i.e. "statuses count" in the Twitter API.
- **isUserVerified(0-1)**: Whether the user is verified or not.
- **NumberOfFollowers**: Number of followers the user has.

[3] 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 classification accuracy, so it has been omitted from the feature set.

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

- **Role score**: This is the ratio between the number of followers and followees (i.e. NumberOfFollowers/NumberOfFollowees).
- **Engagement score**: 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 (Chen and Manning, 2014) 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[5] 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.[6]
- **hasAcronyms(0-1)**: The tweet is checked for presence of acronyms using a acronym dictionary.[7]
- **averageWordLength**: Average length of words (sum of word character counts divided by number of words in each tweet).
- **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.

This work extends the features above, with new additional problem-specific features (**AF features**). AF features score the level of confidence in a tweet. We compute scores for surprise (*surpriseScore (SS)*), doubt (*doubtScore (DS)*), certainty (*noDoubtScore (NDS)*) and support (*supportScore (SPS)*) towards rumourous tweets. For each of these features a list of typical words is collected. We use this list to compute a cumulative vector using word2Vec (Mikolov et al., 2013). 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 each of the scores. Our word embeddings comprise the vectors published by Baroni et al. (2014). The full list of tweet confidence AF features is as follows:

- **surpriseScore (SS)**: cosine between embedding of tweet content and the list of surprise words, e.g. "surprise", "wonder", etc.
- **doubtScore (DS)**: cosine between embedding of tweet content and the list of doubt words, e.g. "doubt", "uncertain", etc.
- **noDoubtScore (NDS)**: cosine between embedding of tweet content and the list of certainty words, e.g."surely", "sure", etc.
- **supportScore (SPS)**: cosine between embedding tweet content and the list of support words, e.g. "support", "confirm", etc.

Furthermore, the following two AF features are included:

- **initialTweetSim (ITS)** captures tweets that tend to support rumours. Every rumour is initiated by a tweet. We compute the cosine similarity based on word2Vec of the tweet being classified to the first tweet in the rumour thread. If the tweet is just a simple retweet of the initial tweet, this is taken as an evidence that the tweet is supportive of that tweet.
- **isQuestion (IQ)** indicates whether a tweet starts with an interrogative. The feature is binary and aims to capture questioning tweets.

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

Classifier	All features	w.o. AF
Decision tree	74.16	72.25
Random Forest	**79.02**	76.54
IBk	75.59	73.02
Baseline-Turing	78.4	–

Table 2: Accuracy scores of different stance classifiers for the RumourEval dataset. The baseline is the best performing system in the SemEval evaluation **Turing**.

4 Experimental Setup

4.1 Baselines

On the RumourEval dataset we run different classifiers (see Section 3.2). We compare the performance of these classifiers against the best-performing system from the RumourEval challenge, namely **Turing** (Kochkina et al., 2017).

We also run all the classifiers from the RumourEval dataset on the PHEME dataset. The results are compared against the following baseline systems reported on the PHEME dataset:

- **Gaussian Processes (GP)** reported by Lukasik et al. (2015).
- **Hawkes Processes (HP)** reported by Lukasik et al. (2016). HPs make use of both temporal and textual information of tweets.

4.2 Training-Testing Settings

We have two different settings. In the first setting we use the SemEval training data to train the models and apply on the testing data. In the second setting we perform training and testing on the PHEME dataset. For the PHEME dataset, we follow the leave one out (LOO) strategy taken by Lukasik et al. (2016) to construct the training and testing data. In LOO n-1 rumours (all tweets within these rumours) are used for training and the resulting model is tested on the n^{th} rumour. Finally, results are macro-averaged.

5 Results

As shown in Table 2 (column two of the table) the best performing learner on the RumourEval dataset is the Random Forest classifier. It achieves the accuracy of *79.02*, higher than any participating system in the RumourEval Task A.[8]

The results on the PHEME dataset are shown in Table 3. Overall the best performing classifier is the J48 decision tree learner. The difference in accuracy scores between the classfiers is tested for significance using paired t-test ($p<0.001$). J48 is only significantly better than IBk and J48 for the *Ottawa shooting* event type. In the remaining event types, J48 performs better, but not significantly better than IBk and Random Forest.

All classifiers J48, IBk and Random Forest, however, outperform the **GP** and **HP** baselines on all event types [9].

What these results demonstrate is that simpler classifiers, such as J48 and Random Forest can outperform significantly more sophisticated machine learning methods (GPs and HPs in this case, and LSTMs in the RumourEval case), thanks to the additional knowledge captured in the rich feature set. In contrast, for example, the GP and HP models relied primarily on BOW and Brown clustering features.

6 Feature Analysis

The results described in Section 5 are based on features reported by related work, enhanced by us with AF features (Section 3.3). We repeat the experiments with AF features removed from the feature set, in order to quantify the extent of their contribution.

For the RumourEval dataset the results are shown in column 3 of Table 2. The omission of the AF features leads to a performance decrease for all classifiers. The accuracy scores also fall below that of the SemEval winner **Turing** – the state-of-the-art system on the RumourEval dataset.

The results on the PHEME dataset are shown in Table 4. The exclusion of the AF features leads to an overall drop in performance when compared to the same classifiers in Table 3. However, these differences are not significant and the classifiers with AF features removed still perform at least as well as the GP and HP baselines (for the event type *Ferguson riots*), or outperform the baselines (for all other event types).

Table 5 shows the accuracy scores of the Random Forest stance classifier, the best performing system on the RumourEval dataset when each AF feature is removed in turn. The results indicate that each AF feature contributes to the accuracy boost in stance classification. The highest accu-

[8] The results are reported in `http://alt.qcri.org/semeval2017/task8/index.php?id=results`

[9] Although the significance test could not be run for the baselines, as the single data point values are not available, the proportion of difference in accuracy and the fact that it is the same data sets let us assume that the classifiers significantly outperform the baselines.

classifier	Ottawa shooting	Ferguson riots	Charlie Hebdo	Sydney siege	macro mean
IBk	70.31*	72.35	**78.33** (ref)	75.44	74.10
Decision tree	**76.28** (ref)	**75.20** (ref)	78.21	**80.01** (ref)	**77.42**
Random Forest	69.39*	69.16	74.57	74.49	71.90
Baseline - GP	62.28	64.31	70.66	65.04	65.57
Baseline - HP	67.77	68.44	72.93	68.59	69.43

Table 3: Accuracy scores for different stance classifiers on the PHEME dataset. * indicates a significant difference to ("ref") scores for each column of the table respectively as indicated by the paired t-test with $p < 0.001$.

classifier	Ottawa shooting	Ferguson riots	Charlie Hebdo	Sydney siege	macro mean
IBk / AF	69.26	69.54	77.09	73.28	72.29
J48 / AF	75.62	74.85	77.05	79.21	76.68
Random Forest / AF	67.87	68.31	75.40	72.57	71.03

Table 4: Accuracy scores of different stance classifiers on the PHEME dataset with AF features removed.

Features	Accuracy
All features	**79.02**
All without AF	76.54
All without ITS	78.55
All without SS	77.59
All without SPS	78.16
All without DS	78.36
All without NDS	77.59
All without IQ	78.64

Table 5: Contribution of each AF feature. Accuracy scores are for the Random Forest classifier on RumourEval data set with each feature removed in turn.

racy loss results from removing the surprise (SS) and certainty (NDS) scores and the least – when the *isQuestion* (IQ) feature is removed. None of the AF feature removals cause a significant drop in accuracy. However, the loss is significant ($p < 0.0001$) when all AF features are removed.

Both RumourEval and PHEME dataset evaluations show that the AF features play an important role in terms of achieving higher accuracy for tweet-based stance classification. They also show the importance of task or problem-specific feature engineering and point out that it is possible with some feature engineering effort to outperform state-of-the-art techniques that are typically considered more powerful and sophisticated than traditional learning methods.

7 Conclusion

This paper tackled the problem of stance classification of tweets towards rumours. In our approach we use a simple classification approach, combining common features reported by related studies with our novel AF features, to boost overall accuracy. Our results show that this approach leads to significantly better results on both RumourEval and PHEME datasets compared to current state-of-the-art systems. Furthermore, our results show that the omission of the AF features proposed in this work leads to significantly lower performance. Adding AF to the feature set causes our approach to outperform the best performing system on the RumourEval dataset. These results show the importance of task- or problem-oriented feature engineering.

The proposed features are content based and work on text level. In our future work we plan to investigate features that are able to capture communication behaviours between users. We also plan to apply stance information as a feature in rumour veracity classification.

Acknowledgments

This work was partially supported by the European Union funded COMRADES (grant agreement No. 687847) and PHEME projects (grant agreement No. 611223), as well as an EPSRC career acceleration fellowship (EP/I004327/1).

References

Isabelle Augenstein, Tim Rocktäschel, Andreas Vlachos, and Kalina Bontcheva. 2016. Stance detection with bidirectional conditional encoding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, pages 876–885. https://aclweb.org/anthology/D16-1084.

Hareesh Bahuleyan and Olga Vechtomova. 2017. UWaterloo at SemEval-2017 Task 8: Detecting

Stance towards Rumours with Topic Independent Features. In *Proceedings of SemEval*. ACL.

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

Douglas Biber. 2006. Stance in spoken and written university registers. *Journal of English for Academic Purposes* 5(2):97–116.

Leo Breiman. 2001. Random forests. *Machine learning* 45(1):5–32.

Fabio Celli, Arindam Ghosh, Firoj Alam, and Giuseppe Riccardi. 2016. In the mood for sharing contents: Emotions, personality and interaction styles in the diffusion of news. *Information Processing & Management* 52(1):93–98.

Danqi Chen and Christopher D Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of EMNLP*. pages 740–750.

Yi-Chin Chen, Zhao-Yand Liu, and Hung-Yu Kao. 2017. IKM at SemEval-2017 Task 8: Convolutional Neural Networks for Stance Detection and Rumor Verification. In *Proceedings of SemEval*. ACL.

Leon Derczynski and Kalina Bontcheva. 2014. PHEME: Veracity in Digital Social Networks. In *Proceedings of the UMAP Workshops*.

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.

Omar Enayet and Samhaa R. El-Beltagy. 2017. NileTMRG at SemEval-2017 Task 8: Determining Rumour and Veracity Support for Rumours on Twitter. In *Proceedings of SemEval*. ACL.

Marianela García Lozano, Hanna Lilja, Edward Tjörnhammar, and Maja Maja Karasalo. 2017. Mama Edha at SemEval-2017 Task 8: Stance Classification with CNN and Rules. In *Proceedings of SemEval*. ACL.

Weiwei Guo and Mona Diab. 2012. Modeling sentences in the latent space. In *Proceedings of ACL*. Association for Computational Linguistics, pages 864–872.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. 2009. The weka data mining software: an update. *ACM SIGKDD explorations newsletter* 11(1):10–18.

Sardar Hamidian and Mona T Diab. 2015. Rumor Detection and Classification for Twitter Data. In *Proceedings of SOTICS*.

Sardar Hamidian and Mona T Diab. 2016. Rumor identification and belief investigation on twitter. In *Proceedings of NAACL-HLT*. pages 3–8.

Elena Kochkina, Maria Liakata, and Isabelle Augenstein. 2017. Turing at SemEval-2017 Task 8: Sequential Approach to Rumour Stance Classification with Branch-LSTM. In *Proceedings of SemEval*.

Percy Liang. 2005. *Semi-supervised learning for natural language*. Ph.D. thesis, Massachusetts Institute of Technology.

Xiaomo Liu, Armineh Nourbakhsh, Quanzhi Li, Rui Fang, and Sameena Shah. 2015. Real-time rumor debunking on twitter. In *Proceedings of CIKM*. ACM, pages 1867–1870.

Michal Lukasik, Trevor Cohn, and Kalina Bontcheva. 2015. Classifying tweet level judgements of rumours in social media. *arXiv preprint arXiv:1506.00468* .

Michal Lukasik, P. K. 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 the 54th Meeting of the Association for Computational Linguistics*. Association for Computer Linguistics, pages 393–398.

Marcelo Mendoza, Barbara Poblete, and Carlos Castillo. 2010. Twitter under crisis: can we trust what we rt? In *Proceedings of the workshop on social media analytics*. ACM, pages 71–79.

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.

Saif M Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016. Semeval-2016 task 6: Detecting stance in tweets. *Proceedings of SemEval* 16.

Rob Procter, Jeremy Crump, Susanne Karstedt, Alex Voss, and Marta Cantijoch. 2013. Reading the riots: What were the police doing on twitter? *Policing and society* 23(4):413–436.

Vahed Qazvinian, Emily Rosengren, Dragomir R Radev, and Qiaozhu Mei. 2011. Rumor has it: Identifying misinformation in microblogs. In *Proceedings of EMNLP*. pages 1589–1599.

Vikram Singh, Sunny Narayan, Md Shad Akhtar, Asif Ekbal, and Pushpak Bhattacharya. 2017. IITP at SemEval-2017 Task 8: A Supervised Approach for Rumour Evaluation. In *Proceedings of SemEval*.

Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng,

and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of EMNLP*. volume 1631, page 1642.

Ankit Srivastava, Rehm Rehm, and Julian Moreno Schneider. 2017. DFKI-DKT at SemEval-2017 Task 8: Rumour Detection and Classification using Cascading Heuristics. In *Proceedings of SemEval*. ACL.

Yla R Tausczik and James W Pennebaker. 2010. The psychological meaning of words: Liwc and computerized text analysis methods. *Journal of language and social psychology* 29(1):24–54.

Peter Tolmie, Rob Procter, Mark Rouncefield, Maria Liakata, and Arkaitz Zubiaga. 2015. Microblog analysis as a programme of work. *arXiv preprint arXiv:1511.03193* .

Feixiang Wang, Man Lan, and Yuanbin Wu. 2017. ECNU at SemEval-2017 Task 8: Rumour Evaluation Using Effective Features and Supervised Ensemble Models. In *Proceedings of SemEval*. ACL.

Li Zeng, Kate Starbird, and Emma S Spiro. 2016. #unconfirmed: Classifying rumor stance in crisis-related social media messages. In *Proceedings of ICWSM*.

Arkaitz Zubiaga, Elena Kochkina, Maria Liakata, Rob Procter, and Michal Lukasik. 2016a. Stance classification in rumours as a sequential task exploiting the tree structure of social media conversations. In *Proceedings of COLING*.

Arkaitz Zubiaga, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Peter Tolmie. 2016b. Analysing how people orient to and spread rumours in social media by looking at conversational threads. *PLoS ONE* 11(3):1–29. https://doi.org/10.1371/journal.pone.0150989.