Exploiting Context for Rumour Detection in Social Media

Arkaitz Zubiaga[1], Maria Liakata[1,2], and Rob Procter[1,2]

[1] University of Warwick, Coventry, UK
[2] Alan Turing Institute, London, UK

Abstract. Tools that are able to detect unverified information posted on social media during a news event can help to avoid the spread of rumours that turn out to be false. In this paper we compare a novel approach using Conditional Random Fields that learns from the sequential dynamics of social media posts with the current state-of-the-art rumour detection system, as well as other baselines. In contrast to existing work, our classifier does not need to observe tweets querying the stance of a post to deem it a rumour but, instead, exploits context learned during the event. Our classifier has improved precision and recall over the state-of-the-art classifier that relies on querying tweets, as well as outperforming our best baseline. Moreover, the results provide evidence for the generalisability of our classifier.

Keywords: social media, rumour detection, breaking news, journalism

1 Introduction

Social media platforms such as Twitter are increasingly being used by people to follow newsworthy events [25] and by journalists for news gathering [37]. However, the speed at which news unfolds on social media means that much of the information posted in the early stages of an event is unverified [22], which makes it more difficult for the public to distinguish verified information from rumours and covering the news becomes more challenging for journalists [29].

We set out to develop a rumour detection system that enables distinguishing between verified and unverified posts. This can be useful to limit the diffusion of information that might turn out subsequently to be false and so reduce the risk of harm to individuals, communities and society [32]. Research in rumour detection is scarce in the scientific literature, [35] being the only published work to date that addresses this issue. They introduced an approach that looks for 'enquiring' tweets, i.e., tweets that query or challenge the credibility of a previous posting to determine whether it is rumourous; a tweet is deemed to be enquiring if it matches one of a number of manually curated, regular expressions. While this is an ingenious approach, it has some important limitations: it is reliant on there being a human in the loop to regularly revise the list of regular expressions as these may not generalise well to new datasets; it assumes that enquiring posts will arise, though this may lead to low recall as not all rumourous posts will

necessarily provoke queries; and it takes no account of the context surrounding the post, which we believe can be exploited to gain insight into the way it emerges. Other work has dealt with "rumour detection" with what we argue is a questionable definition and which conflicts with definitions established in the literature [1,8]. These studies understand rumours as false pieces of information, and therefore misdefine the rumour detection task as consisting of distinguishing true and false stories. In our study we adhere to the established definition that understands a rumour as information circulating while its veracity is yet to be confirmed [1,8]. Consequently, we define the goal of the rumour detection task as that of identifying posts that are yet to be verified, distinguishing them from non-rumours [40].

To the best of our knowledge, our work is the first to attempt rumour detection without having to observe enquiring tweets. Instead, we introduce a sequential approach based on Linear-Chain Conditional Random Fields (CRF) to learn the dynamics of posts, which enables us to classify a post as a rumour or non-rumour while relying on the content of a tweet, in conjunction with context learnt from earlier posts associated with the same event, to determine if it is rumourous. We investigate the performance of CRF as a sequential classifier on five Twitter datasets associated with breaking news to detect tweets that constitute rumours. The performance of CRF is compared with its non-sequential equivalent, a Maximum Entropy classifier, as well as the state-of-the-art rumour detection approach by [35] and other baseline classifiers. Our experiments show substantial improvements and these improvements are consistent across the different events in our dataset.

2 Related Work

Despite increasing interest in rumours in social media [23,26,39,28,31,40], there has been very little work in automatic rumour detection [36]. Much of what work that has been done on rumour detection [24,10,11] has been limited to finding rumours known *a priori*. A classifier is fed with a set of predefined rumours (e.g., *Obama is muslim*), which then classifies new tweets containing a set of relevant keywords (e.g., *Obama* and *muslim*) as being related to one of the known rumours or not (e.g., *I think Obama is not muslim* would be about the rumour, while *Obama was talking to a group of Muslims* would not). An approach like this can be useful for long-standing rumours, where one wants to identify relevant tweets to track the rumours that have already been identified; one may also refer to this task as *rumour tracking* rather than *rumour detection*, given that the rumour is known *a priori*. However, this would not work for contexts such as breaking news, where previously unseen rumours emerge and *a priori* the specific keywords linked to the rumour are not yet known. In such cases, a classifier has to determine if each new update is yet to be verified and hence constitutes a rumour. To deal with such situations, a classifier would need to learn generalisable patterns that will help identify new rumours during breaking stories.

To the best of our knowledge, the only work to tackle the detection of new rumours is that by [35]. Their approach builds on the assumption that rumours will provoke tweets from skeptical users who question or enquire about their veracity; the fact that a post has a number of enquiring tweets associated with it would then imply it is rumourous. The authors created a manually curated list of five regular expressions (e.g., "is (that | this | it) true"), which are used to identify querying tweets. These enquiring tweets are then clustered by similarity, each cluster being ultimately deemed a candidate rumour. Their best approach achieved 52% and 28% precision for two datasets. While this work builds on a sensible hypothesis and presents a clever approach to the rumour detection task, there are three potential limitations: (1) being based on manually curated regular expressions the approach may not generalise well; (2) the hypothesis might not always apply and hence lead to low recall as, for example, certain rumours reported by reputable media are not always questioned by the general public [40]; (3) it takes no account of the context that precedes the rumour, which can give additional insights into what is going on and how a post can be rumourous in that context (e.g., the rumour that *a gunman is on the loose*, when the police have not yet confirmed it, is easier to be deemed a rumour if we put it into the context of preceding events, such as posts that the identity of the gunman is unknown).

While not strictly doing rumour detection, other researchers have worked on related tasks. For instance, there is an increasing body of work [24,15,10,11,17,34] looking into stance classification of tweets discussing rumours, categorising tweets as supporting, denying or questioning the rumour. The approach has been to train a classifier from a labelled set of tweets to categorise the stance observed in new tweets discussing rumours; however, these authors do not deal with non-rumours, assuming instead that the input to the classifier is already cleaned up to include only tweets related to rumours. There is also work on veracity classification both in the context of rumours and beyond [4,14,15,19,33,18,12]. Work on stance and veracity classification can be seen as complementary to our objectives; one could use the set of rumours detected by a rumour detection system as input to a classifier that determines stance of tweets in those rumours and/or veracity of those rumours [36].

3 Dataset

We collected a diverse set of stories that would not necessarily be known *a priori* and which would include both rumours and non-rumours. We did this by emulating the scenario in which a journalist is following reports associated with breaking news. Seeing a timeline of tweets about the breaking news, a user would then annotate each of the tweets as being a rumour or a non-rumour.

Tweets were collected from the Twitter streaming API relating to newsworthy events that could potentially prompt the initiation and propagation of rumours. As soon as our journalist collaborators informed us about a newsworthy event, we set up the data collection process, tracking the main hashtags and

keywords pertaining to the event as a whole. Note that while launching the collection slightly after the start of the event means that we may have missed the very early tweets, we kept collecting subsequent retweets of those early tweets, making it much more likely that we would retrieve the most retweeted tweets from the very first minutes. Once we had the collection of tweets for a newsworthy event, we sampled the timeline to enable manual annotation (signaled by highly retweeted tweets associated with newsworthy current events). Afterwards, the journalists read through the timeline to mark each of the tweets as being a rumour or not, making sure that the identification of rumours was in line with the established criteria [39]. A tweet was annotated as a rumour when there was no evidence or no authoritative source had confirmed it. Note that the annotation of a tweet as a rumour does not imply that the underlying story was later found to be true or false, but instead it reflects that the story was unconfirmed at the time of posting.

We followed the process above for five different newsworthy events, all of which attracted substantial interest in the media and were rife with rumours:

- Ferguson unrest: citizens of Ferguson in Missouri, USA, protested after the fatal shooting of an 18-year-old African American, Michael Brown, by a white police officer on August 9, 2014.
- Ottawa shooting: shootings occurred on Ottawas Parliament Hill, resulting in the death of a Canadian soldier on October 22, 2014.
- Sydney siege: a gunman held hostage ten customers and eight employees of a Lindt chocolate cafe located at Martin Place in Sydney on December 15, 2014.
- Charlie Hebdo shooting: two brothers forced their way into the offices of the French satirical weekly newspaper Charlie Hebdo, killing 11 people and wounding 11 more, on January 7, 2015.
- Germanwings plane crash: a passenger plane from Barcelona to Dsseldorf crashed in the French Alps on March 24, 2015, killing all passengers and crew. The plane was ultimately found to have been deliberately crashed by the co-pilot.

Given the large volume of tweets in the datasets, we sampled them by picking tweets that provoked a high number of retweets. The retweet threshold was set to 100, selected based on the size of the resulting dataset. For each of these tweets in the sampled subset, we also collected all tweets that replied to them. While Twitter does not provide an API endpoint to retrieve 'conversational threads' [30] provoked by tweets, it is possible to collect them by scraping tweets through the web client interface. We developed a script that enabled us to collect and store complete threads for all the rumourous source tweets[1]. We used replying tweets for two purposes: (1) for manual annotation work, where replies to each tweet can provide useful context for the annotator to decide if a tweet is a rumour where the tweet itself does not provide sufficient details; (2) we to reproduce one

[1] Collection script available at https://github.com/azubiaga/pheme-twitter-conversation-collection.

of our baselines classifiers, i.e. the classifier introduced by [35]. However, our approach ignores replying tweets, relying only on the source tweet itself.

The sampled subsets of tweets were visualised in a separate timeline per day and sorted by time. Using these timelines, the journalists were asked to use their knowledge of the events to identify rumours and non-rumours. Along with each tweet, journalists could optionally click on the bubble next to the tweet to visualise replying tweets. The annotation work led to the manual categorisation of each tweet as being a rumour or not. As the journalists progressed along the timeline, new tweets reporting repeated stories were assigned the same annotation as in the previous instance.

The final dataset comprised 5,802 annotated tweets, of which 1,972 were classified as rumours and 3,830 as non-rumours. These annotations are distributed differently across the five events, as shown in Table 3. While slightly over 50% of the tweets were rumours for the Germanwings Crash and the Ottawa Shooting, less than 25% were so for Charlie Hebdo and Ferguson. The Sydney Siege had an intermediate rumour ratio of (42.8%).

Event	Rumours	Non-rumours	Total
Charlie Hebdo	458 (22.0%)	1,621 (78.0%)	2,079
Ferguson	284 (24.8%)	859 (75.2%)	1,143
Germanwings Crash	238 (50.7%)	231 (49.3%)	469
Ottawa Shooting	470 (52.8%)	420 (47.2%)	890
Sydney Siege	522 (42.8%)	699 (57.2%)	1,221
Total	1,972 (34.0%)	3,830 (66.0%)	5,802

Table 1. Distribution of annotations of rumours and non-rumours for the five events.

4 The Rumour Detection Task

We define the rumour detection task as that in which, given a timeline of tweets, the system has to determine which tweets are reporting rumours and hence spreading information that is yet to be verified. Note that the fact that a tweet constitutes a rumour does not imply that it will later be deemed true or false, but that it is unverified at the time of posting. The identification of rumours within a timeline is ultimately meant to warn users that the information has not been confirmed and, while it may later be confirmed, it may also turn out to be false.

Formally, the task takes an evolving timeline of tweets $TL = \{t_1, ..., t_{|TL|}\}$ as input, and the classifier has to determine whether each of these tweets, t_i, is a rumour or a non-rumour by assigning a label from $Y = \{R, NR\}$. Hence, we formulate the task as a binary classification problem, whose performance

is evaluated by computing the precision, recall and F1 scores for the target category, i.e., rumours.

5 Exploiting Context for Rumour Detection

5.1 Hypothesis

In our dataset there were examples where the tweet alone provided sufficient evidence for classifying it as a rumour. For example, in '*the name of the police officer who fatally shot the kid would be reportedly announced by the police later in the day*" the use of "reportedly" expresses uncertainty and so we may conclude that the post is not confirmed. In contrast, posts such as "*the kid was involved in a robbery before being shot*" may not be as easily classified from the tweet alone. Hence, this argues for the need to leverage additional information in the form of context that may help the classifier distinguish between rumours and non-rumours.

One source of tweet context is how others react to it [35]. For example, the tweet "*the kid was involved in a robbery before being shot*" provoked the response *"is that true?"*. However, close examination of rumours in our dataset revealed that this cannot be relied upon. For example, "*the kid was shot 10 times by the police*" provoked no querying response, though it was subsequently revealed to be untrue. Hence, while reactions may be indicative of a posting being unverified, we conclude that relying on this will lead to a classifier with low recall and that the classifier needs to be aware of how the event is unfolding, drawing on the posts that constitute it before the current post. The tweet to be classified as rumour or non-rumour should therefore leverage earlier posts, both rumours and non-rumours, that make up a 'thread' in which th e current tweet fits. For example, a tweet reporting the rumour that "*the police officer who shot the kid has left the town*" may be easier to classify given previous reports related to the police officer and the killing. Based on this, we hypothesise that *aggregating rumourous and non-rumourous posts preceding the tweet being classified will improve performance of the rumour detection system.* We operationalise this by using a sequential classifier that learns from the dynamics of reports observed preceding the current tweet.

5.2 Classifiers

In order to test our hypothesis, we used Conditional Random Fields (CRF) as a sequential classifier that enables aggregation of tweets as a thread of individual posts. We used a Maximum Entropy classifier as the non-sequential equivalent of CRF to test the validity of the hypothesis and also use additional baseline classifiers for further comparison. Moreover, we also reproduced a baseline [35] to compare the performance of our approach with that of a state-of-the-art method.

Conditional Random Fields (CRF). We modeled the twitter thread as a linear chain or graph as a sequence of rumours and non-rumours. In contrast

to classifiers that choose a label for each input unit (e.g., a tweet), CRF also considers the neighbours of each unit, learning the probabilities of transitions of label pairs. The input for CRF is a graph $G = (V, E)$, where in our case each of the vertices V is a tweet and the edges E are relations of tweets, i.e., a link between a tweet and its preceding tweet in the event. Hence, having a data sequence X as input, CRF outputs a sequence of labels Y [13], where the output of each element y_i will not only depend on its features, but also on the probabilities of other labels surrounding it. The generalisable conditional distribution of CRF is shown in Equation 1 [27][2].

$$p(y|x) = \frac{1}{Z(x)} \prod_{a=1}^{A} \Psi_a(y_a, x_a) \tag{1}$$

where Z(x) is the normalisation constant and Ψ_a is the set of factors in the graph G.

Hence, for rumour detection, CRF exploits the sequence of rumours and non-rumours leading up to the current tweet to determine whether it is a rumour. It is important to note that with CRF the sequence of rumours and non-rumours preceding the tweet being classified will be based on the predictions of the classifier itself and will not use any ground truth annotations. Errors in early tweets in the sequence may then increase errors in subsequent tweets.

Maximum Entropy classifier (MaxEnt). As the non-sequential equivalent of CRF, we used a Maximum Entropy (or logistic regression) classifier, which operates at the tweet level. This enabled us to compare directly the extent to which treating the tweets posted during an event as a sequence can boost the performance of the classifier.

Enquiry-based approach [35]: We reproduced this approach, classifying a tweet as a rumour if at least one of the replying tweets matched one of the following regular expressions: (1) is (that | this | it) true, (2) wh[a]*t[?!][?1]*, (3) (real? | really ? | unconfirmed), (4) (rumor | debunk), (5) (that | this | it) is not true.

Additional baselines. We also compared three more non-sequential classifiers[3]: Naive Bayes (NB), Support Vector Machines (SVM) and Random Forests (RF).

We performed the experiments in a 5-fold cross-validation setting, having in each case four of the events for training and the remainder event for testing. This enabled us to simulate a realistic scenario where an event is completely unknown to the classifier and it has to identify rumours from the knowledge garnered from events in the training set. For evaluation purposes, we aggregated the output of all five runs as the micro-averaged evaluation across runs.

[2] We use the PyStruct package to implement CRF [21].
[3] We used the scikit-learn Python package for these baselines.

5.3 Features

We used two types of features: content-based features and social features, testing them individually as well as combined. These two types of features are intended to capture the role that both textual content and user behaviour play in the detection of rumours. Features are limited to those that can be obtained in a real-time scenario, hence we do not consider some features like number of retweets or number of favourites, which are zero at the very beginning and takes time for them to increase as people react.

Content-based Features We use seven different features extracted from the content of the tweets:

- **Word Vectors:** to create vectors representing the words in each tweet, we built word vector representations using Word2Vec [20]. We trained five different Word2Vec model with 300 dimensions, one for each of the five folds, training the model in each case from the collection of tweets pertaining to the four events in the training set, so that the event (and the vocabulary) in the test set was unknown.
- **Part-of-speech Tags:** we built a vector of part-of-speech (POS) tags with each feature in the vector representing the number of occurrences of a certain POS tag in the tweet. We used Twitie [3] to parse the tweets for POS tags, an information extraction package that is part of GATE [5].
- **Capital Ratio:** the ratio of capital letters among all alphabetic characters in the tweet. Use of capitalisation tends to reflect emphasis, among other attributes.
- **Word Count:** the number of words in the tweet, counted as the number of space-separated tokens.
- **Use of Question Mark:** a binary feature representing if the tweet had a question mark in it. Question marks may be indicative of uncertainty.
- **Use of Exclamation Mark:** a binary feature representing if the tweet had an exclamation mark in it. Exclamation marks may be indicative of emphasis or surprise.
- **Use of Period:** a binary feature representing if the tweet contained a period. Punctuation may be indicative of good writing and hence careful reporting.

Social Features We used five social features, all of which can be inferred from the metadata associated with the author of the tweet and which is embedded as part of a tweet object retrieved from the Twitter API. We defined a set of social features that are indicative of a user's experience and reputation:

- **Tweet Count:** we inferred this feature from the number of tweets a user had posted on Twitter. As numbers can vary substantially across users, we normalised them by rounding up the 10-base logarithm of the tweet count: $\lceil \log_{10}(\text{statusescount}) \rceil$.

- **Listed Count:** this feature was computed by normalising the number of lists a user belongs to, i.e., the number of times other users decided to add them to a list: $\lceil \log_{10}(\text{listedcount}) \rceil$.
- **Follow Ratio:** we looked at the reputation of a user as reflected by their number of followers. However, the number of followers might occasionally be rigged, e.g., by users who simply follow many others to attract more followers. To control for this, we defined the follow ratio as the logarithmically scaled ratio of followers over followees: $\lfloor \log_{10} (\#\text{followers}/\#\text{following}) \rceil$.
- **Age:** we computed the age of a user as the rounded number of years that the user has spent on Twitter, i.e., from the day the account was set up to the day of the current tweet.
- **Verified:** a binary feature representing if the user had been verified by Twitter or not, i.e., those whose identity Twitter has validated, and tend to be reputable people.

6 Results

6.1 Comparison of Classifiers

Table 2 shows the results for different classifiers using either or both content-based and social features, as well as the results for the state-of-the-art classifier [35]. Performance using content-based features suggests a remarkable improvement for CRF over the other classifiers. This is especially true when we look at precision, where CRF performs substantially better than the rest. Only the Naive Bayes classifier performs better in terms of recall, however, it performs poorly in terms of precision. CRF clearly balances precision and recall better, outperforming all the other classifiers in terms of the F1 score.

The results are not as clear when we look at social features. CRF still performs best in terms of precision, but recall performance drops, where most of the classifiers perform better than CRF, with SVM being the best. The F1 score shows that SVM best exploits social features, however, performance results using social features are significantly worse than those using content-based features, suggesting social features alone are not sufficient.

When both content-based features and social features are combined, we see that the results resemble that of the use of content-based features alone. CRF outperforms all the rest in terms of precision, while Naive Bayes is good only in terms of recall. The aggregation of features also leads to CRF being the best classifier in terms of F1 score, with CRF giving an improvement of 39.9% over Naive Bayes, the second best classifier. If we compare the results of CRF with the use of content-based features alone or combining both types of features, we notice that all F1 scores for combined features are superior to their counterparts using content-based features alone, among which CRF performs best.

Comparison with respect to the baseline approach [35] supports our conjecture that a manually curated list of regular expressions may lead to low recall. This approach gets a relatively good precision score but it performs substantially worse than CRF. Expanding and/or adapting the list of regular expressions to

	Content			Social			Cont + Social		
Classifier	P	R	F1	P	R	F1	P	R	F1
SVM	0.355	0.445	0.395	0.337	**0.524**	**0.410**	0.337	0.483	0.397
Random Forest	0.271	0.087	0.131	0.343	0.433	0.382	0.275	0.099	0.145
Naive Bayes	0.309	**0.723**	0.433	0.294	0.010	0.020	0.310	**0.723**	0.434
MaxEnt	0.329	0.425	0.371	0.336	0.476	0.394	0.338	0.442	0.383
CRF	**0.683**	0.545	**0.606**	**0.462**	0.268	0.339	**0.667**	0.556	**0.607**

State-of-the-art Baseline			
Classifier	P	R	F1
Zhao et al. [35]	0.410	0.065	0.113

Table 2. Classifier performance.

our specific set of events might improve performance but requires significant manual effort and may still not guarantee better performance in the general case.

6.2 Evaluation by Event

We now examine classifier performance broken down by event so that we can analyse the extent to which the CRF classifier performs well across datasets (see Table 3). The results are mostly consistent across events and in line with the overall performance scores. The Naive Bayes classifier performs best in terms of recall in most cases, however, this is due to it being skewed towards determining that tweets are rumours, as seen in the low precision scores. The CRF classifier achieves the highest precision scores consistently for all the datasets. Moreover, it also achieves the best balance of precision and recall. These results reaffirm the CRF classifier's superiority with respect to the range of classifiers under study, confirming also that exploiting context learned during the event as a sequential set of postings leads to substantially improved performance.

These results also show that while the baseline classifier [35] is among the best in terms of precision, and is often only outperformed by the CRF classifier, it nevertheless performs poorly in terms of recall.

7 Discussion

The aim of a social media rumour detection system is to identify posts whose content have yet to be verified. One application would be alerting users that a report is yet to be verified and so should be treated with caution. Another would

	Germanwings			**Charlie Hebdo**			**Ottawa Shooting**		
Classifier	P	R	F1	P	R	F1	P	R	F1
SVM	0.463	0.504	0.483	0.239	0.546	0.332	0.496	0.428	0.459
Random Forest	0.438	0.029	0.055	0.215	0.203	0.209	0.556	0.053	0.097
Naive Bayes	0.506	**0.882**	0.643	0.223	**0.961**	0.361	0.436	0.087	0.145
MaxEnt	0.475	0.441	0.458	0.239	0.535	0.330	0.512	0.409	0.454
Zhao et al. [35]	0.636	0.059	0.108	0.268	0.057	0.094	0.651	0.060	0.109
CRF	**0.743**	0.668	**0.704**	**0.545**	0.762	**0.636**	**0.841**	**0.585**	**0.690**

	Sydney Siege			**Ferguson**		
Classifier	P	R	F1	P	R	F1
SVM	0.435	0.485	0.458	0.240	0.451	0.313
Random Forest	0.466	0.065	0.114	0.254	0.127	0.169
Naive Bayes	0.426	**0.962**	**0.590**	0.248	**0.820**	0.381
MaxEnt	0.425	0.429	0.427	0.245	0.370	0.295
Zhao et al. [35]	0.429	0.075	0.127	0.355	0.077	0.127
CRF	**0.764**	0.385	0.512	**0.566**	0.394	**0.465**

Table 3. Classifier performance broken down by event.

be as input to classifiers that determine stance of tweets towards rumours [16,38] or classifiers that determine the veracity of rumours [9]. A rumour detection system can in fact be the first component of a system that deals with rumours [36]: (1) rumour detection; (2) rumour tracking; (3) rumour stance classification, and (4) rumour veracity classification.

Our rumour detection experiments on five datasets, each associated with a breaking news story, show that a classifier that sequentially exploits context from earlier tweets achieves significant improvements over non-sequential classifiers. Our CRF classifier substantially outperforms its non-sequential counterpart, a Maximum Entropy classifier, as well as other non-sequential classifiers. Moreover, our approach is better than the state-of-the-art baseline [35] that uses regular expressions to classify as rumours. The latter fails to achieve a competitive recall score, which we believe is for two main reasons: (1) rumours will not always provoke enquiring reactions; and (2) regular expressions may have limited generalisability and require regular manual updates. In contrast, our automated sequential classifiers can classify a tweet as a rumour or non-rumour from its own content and context from earlier tweets, without having to wait for any reactions.

While we are confident that our approach covers a diverse range of rumours and non-rumours, one caveat is that our experiments have been limited to tweets retweeted at least 100 times. While this is consistent with one of the key char-

acteristics of rumours, i.e., that they have to attract a substantial interest to be deemed rumours, it is necessary to wait until a tweet gets retweeted a number of times before it can be considered a candidate for input to the classifier. The development of a classifier that identifies these highly retweeted tweets promptly would enable earlier detection of rumours. Likewise, experimentation with a dataset that includes tweets annotated as rumour or non-rumour which has not been filtered by retweet count would be useful to extend our work and validate with an entire timeline of tweets. The latter has not been possible in our case owing to the cost associated with such large-scale annotation of tweets.

8 Conclusion

We have introduced a novel approach to rumour detection in social media by leveraging the context preceding a tweet with a sequential classifier. Experimenting over five news datasets collected from Twitter and annotated for rumours and non-rumours by journalists, we have shown that this can substantially boost rumour detection performance. Our approach has also proven to outperform the state-of-the-art rumour detection system [35] that relies on finding querying posts that match a set of manually curated list of regular expressions. Their approach performs well in terms of precision but fails in terms of recall, suggesting that regular manual input is needed to revise the regular expressions. Our fully automated approach instead achieves superior performance that is better balanced for both precision and recall.

Social media and user-generated content (UGC) are increasingly important in a number of different ways for the work of not only journalists but also government agencies such as the police and civil protection agencies [22]. However, their use presents major challenges, not least because information posted on social media is not always reliable and its veracity needs to be checked before it can be considered as fit for use in the reporting of news, or decision-making in the case of responses to civil emergencies [22] or natural disasters [2]. Hence, it is vital that tools be developed that can aid: a) the detection of rumours; b) determination of their likely veracity. In the Pheme project [7], we have been developing tools that address the need for the latter [40,16,6]. However, for tools for rumour veracity determination to be effective, they need to be applied in combination with the former and progress so far has been limited. In this paper, we present a novel approach whose performance suggests it has the potential to address this problem.

Finally, we have made the annotated datasets publicly available to promote further research.[4]

[4] `https://figshare.com/articles/PHEME_dataset_of_rumours_and_non-rumours/4010619`

9 Acknowledgments

This work has been supported by the PHEME FP7 project (grant No. 611233). Maria Liakata and Rob Procter were also supported by the Alan Turing Institute. We would also like to thank Queen Mary University of London for the use of its MidPlus computational facilities, which was supported by QMUL Research-IT and funded by EPSRC grant EP/K000128/1.

References

1. Allport, G.W., Postman, L.: An analysis of rumor. Public Opinion Quarterly 10(4), 501–517 (1946)
2. Bazerli, G., Bean, T., Crandall, A., Coutin, M., Kasindi, L., Procter, R.N., Rodger, S., Saber, D., Slachmuijlder, L., Trewinnard, T.: Humanitarianism 2.0. Global Policy Journal (2015)
3. Bontcheva, K., Derczynski, L., Funk, A., Greenwood, M.A., Maynard, D., Aswani, N.: TwitIE: An open-source information extraction pipeline for microblog text. In: Proceedings of the International Conference on Recent Advances in Natural Language Processing. Association for Computational Linguistics (2013)
4. Cai, G., Wu, H., Lv, R.: Rumors detection in chinese via crowd responses. In: IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). pp. 912–917. IEEE (2014)
5. Cunningham, H., Maynard, D., Bontcheva, K.: Text processing with gate. Gateway Press CA (2011)
6. Derczynski, L., Bontcheva, K., Liakata, M., Procter, R., Wong Sak Hoi, G., Zubiaga, A.: Semeval-2017 task 8: Rumoureval: Determining rumour veracity and support for rumours. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017). pp. 69–76. Association for Computational Linguistics, Vancouver, Canada (August 2017)
7. Derczynski, L., Bontcheva, K., Lukasik, M., Declerck, T., Scharl, A., Georgiev, G., Osenova, P., Lobo, T.P., Kolliakou, A., Stewart, R., et al.: Pheme: Computing veracity – the fourth challenge of big social data. In: Proceedings of the Extended Semantic Web Conference EU Project Networking session (ESCW-PN) (2015)
8. DiFonzo, N., Bordia, P.: Rumor, gossip and urban legends. Diogenes 54(1), 19–35 (2007)
9. Giasemidis, G., Singleton, C., Agrafiotis, I., Nurse, J.R., Pilgrim, A., Willis, C., Greetham, D.: Determining the veracity of rumours on twitter. In: International Conference on Social Informatics. pp. 185–205. Springer (2016)
10. Hamidian, S., Diab, M.T.: Rumor detection and classification for twitter data. In: Proceedings of the Fifth International Conference on Social Media Technologies, Communication, and Informatics (SOTICS). pp. 71–77 (2015)
11. Hamidian, S., Diab, M.T.: Rumor identification and belief investigation on twitter. In: Proceedings of NAACL-HLT. pp. 3–8 (2016)
12. Jin, Z., Cao, J., Zhang, Y., Luo, J.: News verification by exploiting conflicting social viewpoints in microblogs. In: Thirtieth AAAI Conference on Artificial Intelligence. pp. 2972–2978 (2016)
13. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proceedings of the eighteenth international conference on machine learning, ICML. vol. 1, pp. 282–289 (2001)

14. Liang, G., He, W., Xu, C., Chen, L., Zeng, J.: Rumor identification in microblogging systems based on users behavior. IEEE Transactions on Computational Social Systems 2(3), 99–108 (2015)
15. Liu, X., Nourbakhsh, A., Li, Q., Fang, R., Shah, S.: Real-time rumor debunking on twitter. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. pp. 1867–1870. ACM (2015)
16. Lukasik, M., Bontcheva, K., Cohn, T., Zubiaga, A., Liakata, M., Procter, R.: Using gaussian processes for rumour stance classification in social media. arXiv preprint arXiv:1609.01962 (2016)
17. Lukasik, M., Cohn, T., Bontcheva, K.: Classifying tweet level judgements of rumours in social media. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 2590–2595 (2015)
18. Ma, J., Gao, W., Mitra, P., Kwon, S., Jansen, B.J., Wong, K.F., Cha, M.: Detecting rumors from microblogs with recurrent neural networks. In: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI). pp. 3818–3824 (2016)
19. Ma, J., Gao, W., Wei, Z., Lu, Y., Wong, K.F.: Detect rumors using time series of social context information on microblogging websites. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. pp. 1751–1754. ACM (2015)
20. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp. 3111–3119 (2013)
21. Müller, A.C., Behnke, S.: Pystruct: learning structured prediction in python. The Journal of Machine Learning Research 15(1), 2055–2060 (2014)
22. Procter, R., Crump, J., Karstedt, S., Voss, A., Cantijoch, M.: Reading the riots: What were the police doing on twitter? Policing and society 23(4), 413–436 (2013)
23. Procter, R., Vis, F., Voss, A.: Reading the riots on twitter: methodological innovation for the analysis of big data. International journal of social research methodology 16(3), 197–214 (2013)
24. Qazvinian, V., Rosengren, E., Radev, D.R., Mei, Q.: Rumor has it: Identifying misinformation in microblogs. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1589–1599 (2011)
25. Sankaranarayanan, J., Samet, H., Teitler, B.E., Lieberman, M.D., Sperling, J.: Twitterstand: news in tweets. In: Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems. pp. 42–51. ACM (2009)
26. Starbird, K., Maddock, J., Orand, M., Achterman, P., Mason, R.M.: Rumors, false flags, and digital vigilantes: Misinformation on twitter after the 2013 boston marathon bombing. Proceedings of iConference 2014 (2014)
27. Sutton, C., McCallum, A.: An introduction to conditional random fields. Machine Learning 4(4), 267–373 (2011)
28. Takayasu, M., Sato, K., Sano, Y., Yamada, K., Miura, W., Takayasu, H.: Rumor diffusion and convergence during the 3.11 earthquake: a twitter case study. PLoS one 10(4), e0121443 (2015)
29. Tolmie, P., Procter, R., Randall, D.W., Rouncefield, M., Burger, C., Wong Sak Hoi, G., Zubiaga, A., Liakata, M.: Supporting the use of user generated content in journalistic practice. In: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems. pp. 3632–3644. ACM (2017)
30. Tolmie, P., Procter, R., Rouncefield, M., Liakata, M., Zubiaga, A.: Microblog analysis as a programme of work. arXiv preprint arXiv:1511.03193 (2015)

31. Tolosi, L., Tagarev, A., Georgiev, G.: An analysis of event-agnostic features for rumour classification in twitter. In: ICWSM Workshop on Social Media in the Newsroom. pp. 151–158 (2016)
32. Webb, H., Burnap, P., Procter, R., Rana, O., Stahl, B., Williams, M., Housley, W., Edwards, A., Jirotka, M.: Digital wildfires: Propagation, verification, regulation, and responsible innovation. ACM Transactions on Information Systems 34(3), 15:1–15:23 (2016)
33. Wu, K., Yang, S., Zhu, K.Q.: False rumors detection on sina weibo by propagation structures. In: 2015 IEEE 31st International Conference on Data Engineering. pp. 651–662. IEEE (2015)
34. Zeng, L., Starbird, K., Spiro, E.S.: # unconfirmed: Classifying rumor stance in crisis-related social media messages. In: Tenth International AAAI Conference on Web and Social Media. pp. 747–750 (2016)
35. Zhao, Z., Resnick, P., Mei, Q.: 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 (2015)
36. Zubiaga, A., Aker, A., Bontcheva, K., Liakata, M., Procter, R.: Detection and resolution of rumours in social media: A survey. arXiv preprint arXiv:1704.00656 (2017)
37. Zubiaga, A., Ji, H., Knight, K.: Curating and contextualizing twitter stories to assist with social newsgathering. In: Proceedings of the 2013 international conference on Intelligent user interfaces. pp. 213–224. ACM (2013)
38. Zubiaga, A., Kochkina, E., Liakata, M., Procter, R., Lukasik, M.: Stance classification in rumours as a sequential task exploiting the tree structure of social media conversations. In: Proceedings of the International Conference on Computational Linguistics (COLING). Association for Natural Language Processing (ANLP) (2016)
39. Zubiaga, A., Liakata, M., Procter, R., Bontcheva, K., Tolmie, P.: Crowdsourcing the annotation of rumourous conversations in social media. In: Proceedings of the 24th International Conference on World Wide Web Companion. pp. 347–353. International World Wide Web Conferences Steering Committee (2015)
40. Zubiaga, A., Liakata, M., Procter, R., Wong Sak Hoi, G., Tolmie, P.: Analysing how people orient to and spread rumours in social media by looking at conversational threads. PLoS ONE 11(3), 1–29 (03 2016), http://dx.doi.org/10.1371%2Fjournal.pone.0150989