Birds of a Feather Check Together: Leveraging Homophily for Sequential Rumour Detection

Shital Lathiya[a], J S Dhobi[a], Arkaitz Zubiaga[b], Maria Liakata[b,c,d], Rob Procter[c,d]

[a]*Government Engineering College, Gandhinagar, India*
[b]*Queen Mary University of London, London, UK*
[c]*Alan Turing Institute, London, UK*
[d]*University of Warwick, Coventry, UK*

Abstract

As breaking news unfolds, social media has become the go-to platform to learn about the latest updates from journalists and eyewitnesses on the ground. The fact that anybody can post content in social media during these breaking news leads to posting and diffusion of unverified rumours, which in turn produces uncertainty and increases anxiety. Given the scale of social media contents, automation is key for effective detection of these rumours. In this paper we introduce a novel approach to rumour detection that learns from the sequential dynamics of reporting during breaking news in social media to detect rumours in new stories. Using five Twitter datasets collected during breaking news stories, we experiment with Conditional Random Fields as a sequential classifier that leverages context learnt during an event for rumour detection, which we compare with the 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 a piece of information to deem it a rumour, but instead we detect rumours from the tweet alone by exploiting context learnt during the event. Further, we experiment with homophily as a predictive feature for detecting rumours, i.e. setting forth the hypothesis that a user will be more likely to post a rumour if they follow users who posted or spread rumours in the past. Our classifier achieves state-of-the-art performance, beating

*Shital Lathiya

Email addresses: shitallathiya1@gmail.com (Shital Lathiya), a.zubiaga@qmul.ac.uk (Arkaitz Zubiaga), m.liakata@qmul.ac.uk (Maria Liakata), rob.procter@warwick.ac.uk (Rob Procter)

competitive baselines as well as outperforming our best baseline with nearly 40% improvement in terms of F1 score. Our research proves the effectiveness of the consideration of the sequential nature of social media data and the usefulness of homophily as a feature for rumour detection.

Keywords: rumour detection, social media, text classification

1. Introduction

The use of social media to follow news stories has become commonplace in recent years. Well-known platforms such as Twitter are increasingly being used by people to learn about the latest developments [1], as well as by journalists for news gathering [2, 3]. This is possible thanks to the way in which they enable users to post and share updates from anywhere and at any time, hence making it possible to get reports from users on the ground who happen to witness a newsworthy event or from users that, for some reason, appear to have access to exclusive information. However, the speed at which breaking news unfolds on social media during fast-paced events, such as terrorist attacks or riots, inevitably means that much of the information posted in the early stages of news reporting is unverified [4]. The presence of such rumours in the stream of tweets makes it more difficult for users to distinguish verified information from rumours, and coverage of the news becomes more challenging for news practitioners.

In this work we set out to develop a rumour detection system that enables flagging of unverified posts, so that one can easily distinguish information that is unsubstantiated. A rumour detection system would ultimately warn users of the unverified status of a post, letting them know that it might later be proven false; this can be useful both 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 [5]. Research in rumour detection is scarce in the scientific literature, Zhao et al.'s [6] being the only published work to date that addresses this issue. They introduced an approach that looks for 'enquiry 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 querying if it matches one of a number of manually curated, regular expressions. As we show in our experiments where we use their approach as a baseline, a manually curated list of regular expressions is limited in terms of generalisability, and cannot deal with unseen types of responses or cases where no such responses are triggered. Other work has dealt with "rumour detection" with what we argue is a questionable definition

and which conflicts with definitions established in the scientific literature [7, 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 in the scientific literature that understands a rumour as the information that is being circulated while its veracity is yet to be confirmed [7, 8]. Consequently, we define the goal of the rumour detection task as that of identifying pieces of information that are yet to be verified, distinguishing them from non-rumours. Our work makes the following contributions within the scope of this definition of the rumour detection task:

- We describe a novel methodology for the collection and annotation of five Twitter datasets containing a diverse range of rumours and non-rumours. Our methodology, developed in close collaboration with journalists, consists in a bottom-up approach that enables going through a timeline of tweets associated with a breaking news story to annotate rumours that were not necessarily known *a priori*. Previous work has largely focused on top-down approaches that first list a set of rumourous stories known to have been circulating, and then go through the tweets to find them, which does not make possible discovery of new rumours that have not yet been listed. The dataset produced by following this methodology is publicly available, which includes 5,802 annotated tweets (1,972 rumours and 3,830 non-rumours).

- To the best of our knowledge, our work is the first to perform the rumour detection task without having to observe querying tweets to identify that a piece of information is rumourous. Instead, we introduce a sequential approach based on Linear-Chain Conditional Random Fields (CRF) to learn the dynamics of information during breaking news, which enables us to classify a piece of information as a rumour or non-rumour by leveraging the context learnt as the event unfolds, and relying only on the content of a tweet to determine if it is rumourous. Hence, our approach does not require a tweet to trigger querying posts to determine if it is rumourous.

- We set forth the hypothesis that users will exhibit the property of homophily in their rumour sharing behaviour, meaning that users will be more likely to post a rumour if they follow others who posted or spread rumours in the past, and vice versa. Our experiments validate this hypothesis by showing a relative increase of 4.4% in F1 score when we add homophily as a feature. Given that the ability to capture homophily is largely dependent on the size of the training data (i.e. the more rumour sharing behaviour we observe the

more precisely we can model homophily), we anticipate that this improvement can be significantly boosted with larger training datasets. The relative improvement of 4.4% reported here is solely dependent on rumour sharing behaviour observed in past events and is therefore directly applicable to new events.

- We investigate the performance of CRF as a sequential classifier on five Twitter datasets associated with breaking news to detect the 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 [6] and additional baseline classifiers. Our experiments show substantial improvements with CRF's use of the sequential dynamics of reporting learnt during an event as context that enriches the content of the tweet itself. These improvements are consistent across the different events in our dataset, as well as over different phases of event reporting, including in the early stages where the sequence to be exploited is more limited.

2. Background: Definition of Rumour

Rumours have been studied and analysed from a range of perspectives, and within and across different disciplines [9]. Definitions given by major dictionaries consistently define rumours as unverified pieces of information, such as the Oxford English Dictionary defining a rumour as "a currently circulating story or report of uncertain or doubtful truth", as well as the Merriam-Webster dictionary defining it as "information or a story that is passed from person to person but has not been proven to be true". Irrespective of the underlying story being ultimately proven true or false, or remaining unsubstantiated, a rumour circulates while it is yet to be verified. A number of researchers have extended the definition of rumour. For instance, [8] define rumours as "unverified and instrumentally relevant information statements in circulation that arise in contexts of ambiguity, danger or potential threat, and that function to help people make sense and manage risk". Moreover, [7] posit that one of the main reasons why rumours circulate is that "the topic has importance for the individual who hears and spreads the story". In [10], the authors also emphasise that "newsworthy events are likely to breed rumors" and that "the amount of rumor in circulation will vary with the importance of the subject to the individuals involved times the ambiguity of the evidence pertaining to the topic at issue".

Consistent with these definitions, we adhere here to a definition adapted to the context of breaking news, which we introduced in previous work [11]: a rumour is a "circulating story of questionable veracity, which is apparently credible but hard to verify, and produces sufficient skepticism and/or anxiety so as to motivate finding out the actual truth". In the context of journalism and breaking news, a rumour can be understood as a piece of information that has not yet been verified, and hence its truth value remains unresolved. In journalism, spreading rumours can have harmful consequences for the reputation of a news organisation if they are used in reporting and later proven false, and hence being able with confidence to quickly assess whether information has not yet been verified as breaking news unfolds is crucial. Likewise, it is important for end users to know when a piece of information constitutes a rumour, so that they are aware of it before spreading the information that is yet to be verified.

3. Related Work

3.1. Rumour Detection

Despite increasing interest in analysing rumours in social media [12, 4, 13, 14, 15, 16, 11, 17, 18] and the building of tools to deal with rumours that had been previously identified [19, 20], there has been very little work in automatic detection of newly-emerging rumours [21, 22], i.e. rumours that were not observed in the training data. Some of the work in rumour detection [23, 24, 25] 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* and one can collect tweets filtered by relevant keywords, rather than having to detect those rumours. However, this would not work for fast-paced contexts such as breaking news, where new, previously unseen rumours emerge, and one does not know *a priori* the specific keywords linked to the rumour, which is yet to be detected. During breaking news, while new reports are released piecemeal, a classifier has to determine if each new update is yet to be verified and hence constitutes a rumour; in these situations, new rumours emerge that had not been observed before. 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 that has tackled the detection of new rumours is that by Zhao et al. [6]. 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 piece of information has a number of querying tweets associated with it would then imply that the information 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 querying tweets are then clustered by similarity, each cluster being ultimately deemed a candidate rumour. It was not viable for the authors to evaluate by recall, but 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 tackling the rumour detection task, we foresee 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 [11], and (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 piece of information can be rumourous in that context (e.g., the rumour that *a gunman is on the loose*, when the police has not confirmed it yet, is easier to be deemed a rumour if we put it into the context of the preceding events, such as additional reports that the identity of the gunman is unknown and the reasons that motivated the shooting have not been found out). In this work, we introduce a context-aware rumour detection system that uses a sequential classifier to examine the reporting dynamics during breaking news to determine if a new piece of information constitutes a rumour. This paper expands previous work described in [26], exploring the utility of homophily as a feature for rumour detection, as well as providing an extended analysis on the tweets that the system can best identify, e.g. looking at early and late stages of an event.

Recent work has also further looked at the rumour detection task along the lines of the above definition, however with significant alterations that do not allow comparison with the present work. These include (1) a substantially altered version of our dataset to make it balanced [27], without further detail of the resulting dataset, and (2) as well as amplifying our dataset to make it orders of magnitude larger [28], whose details are missing.

3.2. Claim Detection for Fact-checking

While rumour detection is the task of flagging unverified information in circulation, claim detection is a related task that is part of the fact-checking pipeline. Claim detection, or sometimes also referred to us check-worthiness detection, consists in selecting or prioritising statements of interest to fact-checking organisations and needing fact-checking [29]. Similarities between these two tasks include the need to detect that need attention, either for verification (rumours) or fact-checking (claims). The major differences between both tasks is that rumour detection consists in identifying every statement of unverified nature, with the aim of asking journalists and the public at large to hold off sharing that piece of information while it is not yet verified. Claim detection, instead, has the objective of finding statements that are both worthy of checking and checkable. For example, a statement like "*Ten people have died and the police are investigating if an eleventh person may have died*" is a rumour that has not yet been verified; it is, however, not of interest to fact-checking organisations provided that it is not checkable. Conversely, "*Investment in education has gone up by 10%*" would not typically be deemed a rumour, whereas it is a typical claim of interest for fact-checkers. Still, both tasks have significant similarities and one could consider using similar methods for tackling both tasks.

Two of the best-known publicly available and published systems for claim detection include (i) ClaimBuster [30], which combines several features such as TF-IDF, POS tags and NER on an SVM to produce importance scores for each claim in a collection, and (ii) ClaimRank [31], which uses a large set of features both from individuals sentences and from surrounding context for ranking check-worthy claims. Another recent method described in [32] uses embedding-based methods –in this case, InferSent [33]– for detecting claims by leveraging contextual features within sentences, showing that, among others, the use of numeric expressions can be of significant help for detecting claims, such as the education investment example above.

3.3. Related Tasks

While not strictly doing rumour detection, other researchers have worked on related tasks. For instance, there is an increasing body of work [23, 34, 24, 25, 35, 36] 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 [37, 38, 39, 34, 40, 41, 42, 43, 44, 45, 46]. 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. However, this previous step of distinguishing between rumours and non-rumours is largely unexplored, and most work deals directly with subsequent steps.

4. Dataset

One of our main objectives when planning to put together a dataset of rumours and non-rumours was to develop a means to collect a diverse set of stories, which 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 user 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. To make sure that our users had the expertise to perform this annotation, we enlisted the help of a team of journalists who are partners of our research project on social media rumours. Our data collection approach differs substantially from that of previous work [23, 12, 13], who first identified the rumours of interest and then collected tweets associated with those by filtering using relevant keywords. By following the latter approach of gathering rumours known *a priori*, one can, for instance, search for tweets with specific keywords, e.g., for tweets posted during the 2011 England Riots, one can search for 'London Eye fire', to retrieve tweets associated with the rumour that the London Eye had been set on fire. However, this requires the rumour in question to be known *a priori*, and will fail to identify rumours associated with events for which specific keywords have not been previously defined. This approach will miss some of the rumours, a problem that we overcome here by having journalists sift through the timeline of tweets to identify rumours. This annotation is done soon after the events occurred so that they can make use of their recent experience covering the story as well as their recent discussions in the newsroom to determine the reports that were yet to be verified. While we only collected a single annotation per tweet as rumour or non-rumour, this decision was taken after collective discussions in the newsroom and based on earlier discussions while the story was unfolding. Well established verification practices, such as those written by [47, 48], have been used to inform the annotation work. While manual annotation of the whole stream of tweets as-

sociated with breaking news is not viable, we alleviate the task by sampling the tweets that provoked a large number of retweets and hence are likely to be of interest for reporting. This is also consistent with one of the main characteristics of rumours, which tend to generate significant levels of interest.

4.1. Data Collection

The data collection consisted in retrieving tweets from the Twitter streaming API relating to newsworthy events that could potentially prompt the initiation and propagation of rumours. Collection through the streaming API was launched straight after the journalists identified a newsworthy event likely to give rise to rumours, 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 in place, we sampled the timeline of tweets to enable manual annotation (signaled by highly retweeted tweets associated with newsworthy current events). Afterwards, 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 [14]. Each tweet was examined to deem it a rumour or non-rumour based on whether it was verified (non-rumour) or not (rumour). They annotated a tweet as a rumour when there was no evidence to confirm it and 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: (1) 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; (2) Ottawa shooting: shootings occurred on Ottawa's Parliament Hill in Canada, resulting in the death of a Canadian soldier on October 22, 2014; (3) Sydney siege: a gunman held hostage ten customers and eight employees of a Lindt chocolate caf é located at Martin Place in Sydney, Australia, on December 15, 2014; (4) Charlie Hebdo shooting: two brothers forced their way into the offices of the French satirical weekly newspaper Charlie Hebdo in Paris, killing 11 people and wounding 11 more, on January 7, 2015; (5) Germanwings plane crash: a passenger plane from Barcelona to Düsseldorf 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 of the plane.

4.2. Data Sampling

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 collect all the tweets that reply to them. While Twitter does not provide an API endpoint to retrieve conversations 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 conversations for all the rumourous source tweets[1]. We use the replying tweets for two purposes: (1) for the 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, and (2) we use them to reproduce one of our baselines classifiers, i.e. the classifier introduced by [6]. However, our approach ignores replying tweets, relying only on the source tweet itself.

4.3. Annotation of Rumours and Non-Rumours

The sampled subsets of tweets were visualised in a separate timeline per day and sorted by time (see Figure 1). Using these timelines, journalists were asked to identify rumours and non-rumours. Along with each tweet, journalists could optionally click on the bubble next to the tweet to visualise tweets replying to the tweet; the conversation provoked by the tweet could assist them by providing context, albeit the annotation was independent of this context and based on their experience. The fact that this annotation work was performed by journalists was convenient as they had continually tracked the five events while they were unfolding, and so they were knowledgeable about the stories as they had to cover them and had discussions in the newsroom. The annotation, however, was conducted *a posteriori*, once the reporting about the event had come to an end. This encouraged careful annotation that encompassed a broad set of rumours; the journalists could go through the whole timeline of tweets as we presented them and perform the annotations. The annotation work led to the manual categorisation of each tweet as being a rumour or not. The methodology they followed to perform these

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

annotations in a more manageable and scalable way was to go through the timeline by analysing carefully those tweets that reported new stories that they had not seen before; for those cases, they investigated the story further on social media and the Web when the origin and nature of the story was not known to them. As they progressed in the timeline, new tweets reporting repeated stories where assigned the same annotation as in the previous instance. This made their job easier as they only had to investigate carefully stories that they had not seen.

Annotation of tweet timelines led to cases where a rumour had been verified at a certain point. While such a story is not strictly a rumour anymore, we opted for labelling those cases as rumours, with an indication that it was verified from a certain point onwards. The rationale for this is that we intended to design our rumour detection system in a way that all tweets related to the same rumourous story would be identified, with a subsequent step in the system pipeline detecting this verification timepoint. In this paper, we deal with the first step, detection of rumours, with the subsequent verification step left for future work.

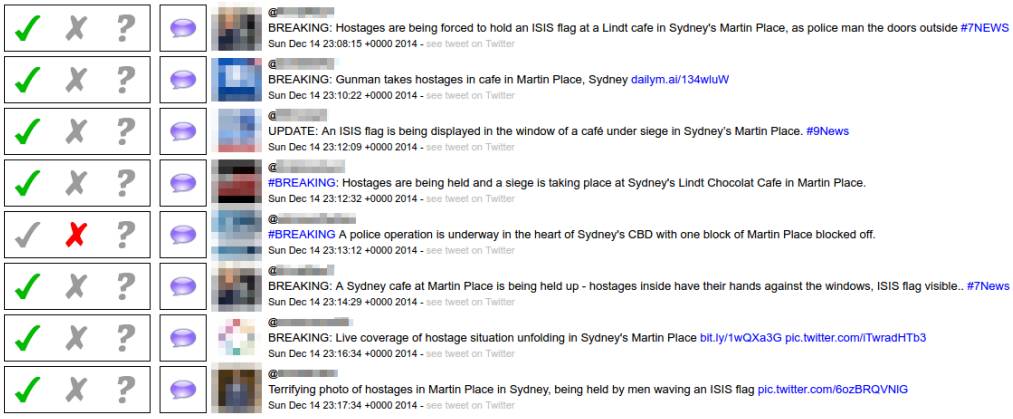

Figure 1: Screenshot of the tool that the journalists utilised to annotate tweets in a timeline as being rumours or non-rumours. Each tweet can be annotated as a rumour (green tick) or a non-rumour (red cross). The question mark was solely used to leave a tweet for later. Additionally, the bubble next to each tweet enables to visualise the tweets replying to that tweet, used occasionally for context.

4.4. Resulting Dataset

The annotation of tweets sampled for all five events led to a collection of 5,802 annotated tweets, of which 1,972 were deemed rumours and 3,830 were deemed

non-rumours.[2] These annotations are distributed differently across the five events, as shown in Table 1. 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 ratio of rumours (42.8%). While the global figures of rumours vary substantially across events, we dug further into these distributions to understand how rumours and non-rumours are distributed during events, e.g., to look at whether rumours occur especially at the beginning of the event, along with the very early reports. To do this, we broke down the timeline of tweets for each event into deciles (10% percentiles) and look at the ratio of rumours in each of these deciles. Figure 2 shows the ratios of rumours for each of the deciles for the five events in our dataset. Contrary to what we initially expected, there is no common pattern across events. One can see events with uniformly distributed ratios of rumours, such as with the Ottawa Shooting, events where the ratio of rumours fades at least eventually, such as Charlie Hebdo, Germanwings crash and Sydney Siege, or events where the majority of the rumours emerge in later stages of the reporting, such as Ferguson. These varying distributions of rumours across different events makes the rumour detection task even more challenging, as one may not be able to rely on the earliness of a report to determine if it is more likely to be a rumour.

Table 1: Distribution of annotations of rumours and non-rumours for the five breaking news datasets.

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

4.5. Collection of Following Users

In addition, we also collected complete lists of all users followed by all the users in our datasets. This allows us to establish if there is a follow relationship

[2]The dataset can be downloaded at `https://figshare.com/articles/dataset/PHEME_dataset_of_rumours_and_non-rumours/4010619`

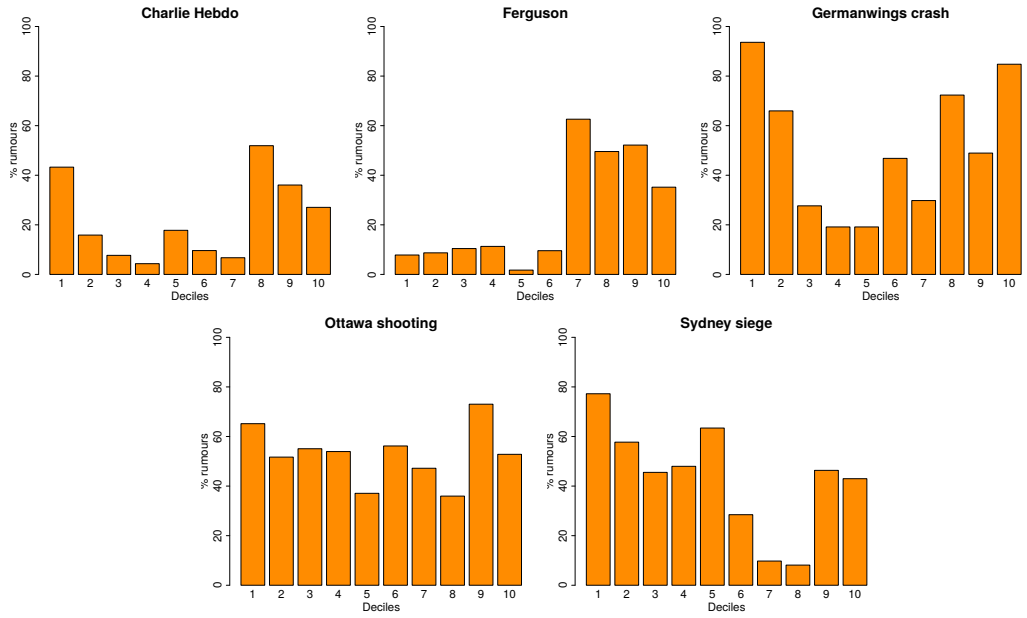

Figure 2: Rumour ratios for deciles within the timeline of each event, showing how ratios of rumours evolve as the event unfolds.

between all pairs of users in the datasets and hence to compute the homophily feature, as we describe later in the experiments.

5. Rumour Detection Task

We define the rumour detection task as that in which, given a timeline of tweets, the system has to determine which of the tweets are reporting rumours, and hence are 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 instead 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. This can be operationalised by flagging those tweets that are identified as rumours, warning users to think twice before spreading the information. 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.

When it comes to the applications of a rumour detection system, one could see it either as a stand alone application or as a part of a bigger rumour classification system. A stand along system could be one that flags information as being unverified and therefore warns users before they decide to spread it. As part of a larger system, a rumour classification tool could be composed of (1) a rumour detection system that identifies newly emerging rumours, (2) a rumour tracking system that collects new posts related to the previously identified rumour, (3) a rumour stance classifier that classifies each of the posts as support, denying, questioning or commenting on the rumour, and (4) a veracity classifier that ultimately determines if the rumour is true or false, or alternatively cannot yet be verified for not having sufficient evidence. In this work we focus on the first component dealing with rumour detection, i.e. identifying unverified pieces of information that need to be tracked and eventually verified or debunked.

6. Learning Sequential Dynamics for Rumour Detection

6.1. Hypothesis

Our work builds on two key hypotheses:

Hypothesis 1. Context is Crucial in Rumour Detection.

We hypothesise that a single headline or tweet may not always be indicative of a piece of information being a rumour. There are, indeed, cases where a single tweet uses hedging words or provides little or no evidence so as to be deemed corroborated information, and hence those cases can be deemed rumours from the tweet alone. This is the case, for instance, of tweets reporting during the Ferguson riots that "*the name of the police officer who fatally shot the kid would be reportedly announced by the police later in the day*". If the tweet itself expresses uncertainty, as is the case here with the use of "reportedly", one can consider that the underlying information is not confirmed. However, reports confidently reporting that "*the kid was involved in a robbery before being shot*" may not be as easily identified by an automated classifier from the tweet alone, despite being an unverified piece of information and hence a rumour. The dearth of sufficient evidence as occurs in many tweets encourages us to further leverage context that could help the classifier distinguish rumours and non-rumours.

One possibility to extend a tweet with context is to look at how others react to it, as [6] proposed in their work that querying or enquiring tweets provoked by

a posting may indicate it is a rumour. This means for instance that for the tweet "*the kid was involved in a robbery before being shot*", somebody will respond with a question *"is that true?"*. However, we believe the public will not always question the veracity of rumours, given that average users may not always notice that a piece of information is not confirmed. This is the case of a number of tweets during the Ferguson riots reporting that "*the kid was shot 10 times by the police*". While this information was not queried by the public, the media treated the information as not being verified; instead, they waited for the autopsy to be carried out, in this case resulting in the rumour being false as he was found to be shot 6 times. Hence, while reactions may be indicative of a piece of information being unverified, we believe that it may lead to low recall, missing other cases that are not rebutted.

To better harness the context surrounding a tweet, we believe that the classifier needs to be aware of how the whole event is unfolding, analysing the different announcements that build a story before the current tweet is posted. The tweet that is being classified as rumour or non-rumour should therefore leverage earlier happenings within that event, both rumours and non-rumours, that make up a story in which the current tweet fits. For instance, a tweet reporting the rumour that "*the police officer who shot the kid has left the town*" may be easier to classify being aware of all the previous reports related to the police officer and the killing. Based on this, we set forth the hypothesis that *aggregation of all the rumourous and non-rumourous reports leading up to the tweet being classified will provide key context to boost performance of the rumour detection system*. We operationalise this hypothesis by using a sequential classifier that mines the discourse leading to a tweet, learning from the dynamics of reports observed throughout the event.

Hypothesis 2. Users will exhibit homophily, following others with similar rumour-sharing behaviours.

Previous research has shown evidence that users in social networks tend to form ties with others with similar characteristics, be it in terms of "genders, races, ethnicities, ages, class backgrounds, educational attainment, etc." [49]. Not only does this apply to social media, but also to the offline world, with evidence showing that friendship in school is largely determined by one's gender and race [50]. A number of recent studies in social media have shown that homophily is a strong feature to predict friendship [51], political ideology [52] and retweeting or sharing behaviour [53], among others.

We hypothesise that homophily will also have an impact on rumour sharing, suggesting that users with similar rumour sharing behaviours will be connected to each other. Reversely, we propose to leverage existing connections (i.e. follow

relationships) between users to determine that if a user has shared rumours in the past, other users connected to them will be more likely to engage in rumour sharing or posting in the future. Conversely, users will be less likely to share or post rumours if they are connected to users who have not posted or shared rumours in the past.

6.2. Classifiers

In order to test our hypothesis, we use Conditional Random Fields (CRF) as a sequential classifier that enables aggregation of tweets as a chain of reports. We use 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 reproduce a baseline based on the approach introduced by [6] to compare the performance of our approach with that of a state-of-the-art approach.

Conditional Random Fields (CRF). We use CRF as a structured classifier to model sequences of tweets as observed in the timelines of Twitter breaking news. With CRF, we can model the timeline as a linear chain or graph that will be treated as a sequence of rumours and non-rumours. In contrast to classifiers traditionally used for this task, which choose a label for each input unit (e.g., a tweet), CRF also consider the neighbours of each unit, learning the probabilities of transitions of label pairs to be followed by each other. 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 [54], 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 [55][3].

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

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

Therefore, in our specific case of rumour detection, CRF will exploit the sequence of rumours and non-rumours leading up to the current tweet to determine

[3]We use the PyStruct package to implement Conditional Random Fields [56].

if it is a rumour or not. 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 augment errors in subsequent tweets. For each tweet to be classified, we solely feed the preceding tweets to the classifier to simulate a realistic scenario where subsequent tweets are not yet posted and early decisions need to be made on each tweet.

Logistic Regression or Maximum Entropy classifier (MaxEnt). As the non-sequential equivalent of CRF, we use a Maximum Entropy (or logistic regression) classifier, which is also a conditional classifier but which will operate at the tweet level, ignoring the sequence and hence the preceding tweets. This enables us to compare directly the extent to which treating the tweets posted during an event as a sequence instead of having each tweet as a separate unit can boost the performance of the classifier.

Enquiry-based approach by [6]: As a state-of-the-art baseline for rumour detection, and the only approach that so far has tackled rumour detection in social media, we reproduce the approach by Zhao et al., which uses regular expressions to look for enquiry posts. We use the set of replies responding to each tweet to look for enquiry posts. Following the approach described by the authors, we consider that a tweet is a rumour if at least one of the replying tweets matches with one of the regular expressions that the authors curated. The list of regular expressions defined by the authors is shown in Table 2.

Table 2: List of regular expressions utilised by Zhao et al., which we reimplemented to reproduce their approach as a baseline. Regular expressions for both enquires and corrections are combined, and a tweet that matches any of them will be deemed an enquiry tweet.

Pattern Regular Expression	Type
is (that | this | it) true	Verification
wh[a]*t[?!][?1]*	Verification
(real? | really ? | unconfirmed)	Verification
(rumor | debunk)	Correction
(that | this | it) is not true	Correction

Additional baselines. We also compare three more non-sequential classifiers[4]: Naive Bayes (NB), Support Vector Machines (SVM), and Random Forests

[4]We use their implementation in the scikit-learn Python package for Maximum Entropy, Naive

(RF), as well as a baseline classifier predicting everything as a rumour (Rum). A classifier predicting everything as non-rumour is not tested, as an all-non-rumour classifier is considered useless when the task consists in rumour detection.

We perform 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 enables 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 aggregate the output of all five runs as the micro-averaged evaluation across runs.

6.3. Features

We use two types of features with the classifiers: content-based features and social features. We test them separately as well as combined. The features that fall in each of these categories are as follows:

6.3.1. RumourRatio and Homophily as Features

We aim to capture the impact of homophilic connections between users. To capture that, we consider RumourRatio as a metric that measures the extent to which others followed by user have engaged in rumour sharing in the past. Incorporating our hypothesis of homophily in rumour sharing, we then expect that a user will be more likely to post rumours in the future if their network connections exhibited a high RumourRatio in the past. We therefore expect that a user's connections having shared multiple rumours in the past indicates that the user itself will share rumours in the future.

To compute the RumourRatio, we consider the number of actual rumour or non-rumour posts posted by a user in the past. We calculate RumourRatio i.e., number of rumours spread by a user over the total number of rumours and non-rumours posted by that user as:

$$RumourRatio = \frac{\#rumours}{\#rumours + \#nonrumours} \tag{2}$$

where #rumours is total number of rumours posted by a user and #nonrumours is total number of non-rumours posted by user.

This is however a limited metric when one has small training data and has not observed any prior interventions of a user. Hence, we use an alternative calculation for the users who don't have past history. In those cases, we resort the idea

Bayes, Support Vector Machines and Random Forests.

of homophily, i.e. given that we don't know anything about the user itself, let's assume that their behaviour will resemble that of the connections in their network. We hypothesise that users are more likely to follow others like them, i.e. if they follow many rumour-sharers, then they are more likely to share rumours too. In this case, we calculate RumourRatio of a user who hadn't posted any rumour yet as the average RumourRatio of all the users followed by that particular user as per below equation:

$$RumourRatio = \frac{\sum_{n=1}^{N} FollowingRumourRatio}{N} \tag{3}$$

where N is total number of users followed by a particular user and FollowingRumourRatio is RumourRatio of users followed by a particular user.

For each of the five events in our dataset, we calculate RumourRatio of a user and its followings from the other four events. This allows us to hold out one event for testing while only the rest are used for the calculations of RumourRatios.

6.3.2. 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 build word vector representations using Word2Vec [57]. We train a different Word2Vec model with 300 dimensions 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 is unknown. As a result, we get five different Word2Vec models, each used in a separate fold.

- **Part-of-speech Tags:** we build 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 use Twitie [58] to parse the tweets for POS tags, an information extraction package that is part of GATE [59].

- **Capital Ratio:** the ratio of capital letters among all alphabetic characters in the tweet. Use of capitalisation tends to represent emphasis, among others.

- **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 has at least a question mark in it. Question marks may be indicative of uncertainty.

- **Use of Exclamation Mark:** a binary feature representing if the tweet has at least an exclamation mark in it. Exclamation marks may be indicative of emphasis or surprise.

- **Use of Period:** a binary feature representing if the tweet has at least a period in it. Punctuation may be indicative of good writing and hence potentially of slow reporting.

6.3.3. Social Features

We use 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 define a set of social features that are indicative of a user's experience and reputation:

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

- **Listed Count:** this feature is 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:** in this feature we look 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 effect, we define the follow ratio as the logarithmically scaled ratio of followers over followees: $\lfloor \log_{10}\ (\#\text{followers}/\#\text{following}) \rceil$.

- **Age:** we compute 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 has been verified by Twitter or not. Verified users are those whose identity Twitter has validated, and tend to be reputable people.

7. Results

7.1. Comparison of Classifiers

Table 3 shows the results for different classifiers using either or both of the content-based and social features, as well as the results for the state-of-the-art classifier by [6]. Performance results of the classifiers using content-based features suggests a remarkable improvement for CRF over the rest of the classifiers, implying that CRF benefits from the use of the sequence of tweets preceding each tweet as context to enrich the input to the classifier. 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. As a result, CRF balances precision and recall in a clearly better way, outperforming all the other classifiers in terms of the F1 score.

Results are not as clear when we look at those using social features. CRF still performs best in terms of precision, but performance drops if we look at the recall. In fact, most of the classifiers perform better than CRF in terms of recall, with SVM as the best performing classifier. Combining both precision and recall in an F1 score shows that SVM is the classifier that best exploits social features. However, performance results using social features are significantly worse than those using content-based features, which suggests that social features alone are not sufficient.

When both content-based features and social features are combined as an input to the classifier, 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. As a result, the aggregation of features also leads to CRF being the best classifier in terms of F1 score. In fact, CRF leads to an improvement of 39.9% over the second best classifier in terms of F1, Naive Bayes. If we compare the results of CRF with the use of content-based features alone or combining both types of features, we notice that the improvement comes especially for recall, which is balanced out with a slight drop of precision. As a result, we get an F1 score that is slightly better when using both features together. In fact, 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 enquiry-based baseline approach introduced by Zhao et al. buttresses our conjecture that a manually curated list of regular expressions may lead to low recall, which is as low as 0.065 in this case. This approach gets a relatively good precision score, which beats all of our baselines, although it performs substantially worse than CRF. However, 59% of false pos-

itives as can be inferred from the precision of 0.41 indicates that the regular expressions also match non-rumours. One could also opt to expand the list of regular expressions and/or adapt them to our specific scenario and events; however, this may involve substantial manual work and would not guarantee generalisable performance.

Content			
Classifier	P	R	F1
Rum	0.340	0.340	0.340
SVM	0.355	0.445	0.395
RF	0.271	0.087	0.131
NB	0.309	**0.723**	0.433
LogReg	0.329	0.425	0.371
CRF	**0.683**	0.545	0.606

Social			
Classifier	P	R	F1
Rum	0.340	0.340	0.340
SVM	0.337	0.524	0.410
RF	0.343	0.433	0.382
NB	0.294	0.010	0.020
LogReg	0.336	0.476	0.394
CRF	0.462	0.268	0.339

Content + Social			
Classifier	P	R	F1
Rum	0.340	0.340	0.340
SVM	0.337	0.483	0.397
RF	0.275	0.099	0.145
NB	0.310	**0.723**	0.434
LogReg	0.338	0.442	0.383
CRF	**0.667**	0.556	0.607

State-of-the-art Baseline			
Classifier	P	R	F1
[6]	0.410	0.065	0.113

Table 3: Results for the different classifiers using content and/or social features.

7.2. Using RumourRatio and Homophily as Features

Next we assess the effectiveness of the proposed RumourRatio and Homophily based features. Table 4 shows results for the best performing classifier, CRF, as well as its performance difference when RumourRatio (RR) and Homophily (HP) are leveraged as features. Results show clear improvements in terms of recall and F1 scores with the use of both RR and HP as features. This improvement is incremental; while the use of RR leads to improvement, the addition of the concept of homophily to expand the feature based on following user accounts leads to further improvement with the use of HP as a feature.

The improvement in terms of recall and F1 scores comes at the expense of precision, which generally drops with the use of either RR or HP features; the exception is when this is applied to social features, which however performs worse than the rest. We consider, however, that improvement in terms of recall is desired in the context of rumour detection, where one wants to make sure to flag as many rumours as possible before they slip through as non-rumours. Hence, we positively value an improvement in recall which is also reflected in increased F1 scores.

Content			
Classifier	P	R	F1
CRF	**0.683**	0.545	0.606
CRF + RR	0.663	0.580	0.619
CRF + HP	0.646	**0.616**	**0.630**

Social			
Classifier	P	R	F1
CRF	0.462	0.268	0.339
CRF + RR	**0.534**	0.460	0.494
CRF + HP	0.423	**0.794**	**0.552**

Content + Social			
Classifier	P	R	F1
CRF	**0.667**	0.556	0.607
CRF + RR	0.654	0.593	0.622
CRF + HP	0.633	0.635	**0.634**

Table 4: Results comparing CRF with the original features, as well as incorporating RR (Rumour-Ratio) and HP (Homophily) as features.

7.3. Evaluation by Event

Further delving into the use of CRF as a classifier and the impact of using RR and HP as features, we now look into their performance broken down by event, so that we can analyse the extent to which these features are helpful across events. Table 5 shows the results broken down by event, for the five datasets under study. Results are mostly consistent across events, and in line with the overall performance scores reported in the previous section.

These results again show a consistent improvement in terms of recall across all events when we use RR and HP features. Again, the RR feature leads to improve over the base CRF classifier, whereas the HP feature leads to a further improvement, in terms of recall. Results in terms of F1 scores are however not

as consistent. We observe improvements using the HP feature over the base CRF for three of the events (Germanwings crash, Ottawa shooting, Sydney siege), with performance drops in terms of F1 scores for the other two events (Charlie Hebdo, Ferguson). Hence, despite consistent improvement in terms of recall, and overall improvement in terms of F1 score when we aggregate all five events, performance in F1 score can have varying degrees of success across events. Using HP as a feature is still the safest bet when one wants to achieve high recall in rumour detection.

Germanwings crash			
Classifier	P	R	F1
CRF	**0.743**	0.668	0.704
CRF + RR	0.694	0.706	0.700
CRF + HP	0.686	**0.752**	**0.717**

Charlie Hebdo			
Classifier	P	R	F1
CRF	**0.545**	0.762	**0.636**
CRF + RR	0.522	0.762	0.620
CRF + HP	0.488	**0.806**	0.608

Ottawa Shooting			
Classifier	P	R	F1
CRF	**0.841**	0.585	0.690
CRF + RR	0.835	0.594	0.694
CRF + HP	0.819	**0.647**	**0.723**

Sydney Siege			
Classifier	P	R	F1
CRF	**0.764**	0.385	0.512
CRF + RR	0.754	0.510	0.608
CRF + HP	0.744	**0.550**	**0.632**

Ferguson			
Classifier	P	R	F1
CRF	**0.566**	0.394	**0.465**
CRF + RR	0.565	0.380	0.455
CRF + HP	0.554	**0.398**	0.463

Table 5: Results broken down by event, comparing the use of CRF with social+content features, as well as addition of RumourRatio (RR) and Homophily (HP) as features.

7.4. Consistency of the Sequential Classifier's Performance

Even though CRF as a sequential classifier has proven to perform better than the rest of the non-sequential classifiers overall, we are interested in seeing whether the high performance of CRF is consistent over time. Given that CRF depends on

the sequence as context to enhance classification of a tweet, the first few tweets in each event lack the context that later tweets have. To analyse performance over time, we look at the F1 scores for each decile and for each event separately. Figure 3 shows these results, broken down by event and decile; thick, orange bars represent the F1 score of the CRF + HP classifier in each decile, while the thinner, grey bars represent the highest F1 score across all non-sequential classifiers in each decile. We make some interesting observations from these results:

- CRF generally outperforms the best of the non-sequential classifiers, also when breaking down the results by decile.
- CRF does suffer from a cold start problem at the beginning of each event, which is especially noticeable with Charlie Hebdo, Ottawa shooting and Sydney siege. In the five events under study, CRF performs better in the second decile than in the first, and it tends to perform better in later deciles than in the first. This may indicate, as we conjectured, that CRF can only make use of a short sequence, providing little context, when classifying the very first tweets. Nevertheless, CRF also generally shows better performance for the initial deciles than the best of the non-sequential classifiers, except for the Sydney siege, where non-sequential classifiers achieve better performance.
- Not all the events have the same characteristics when it comes to rumour spawning, and for some events it even becomes challenging to detect rumours in later stages. This occurs, for instance, with Ferguson, where performance drops in the 10th decile, or with Germanwings crash, where performance drops in the 7th decile and then it progressively recovers again.

8. Analysis of Users

In this section, we investigate the distributions of users across events in our dataset. This helps us determine the extent to which we are actually dealing with new users in each event, as well as the extent to which the use of homophily is having an impact on our experiments.

Table 6 shows the overlaps of unique users across the five events in our dataset. These figures highlight the very low overlap across events, all overlap values comparing two different events being below 10%. This indicates that the users in the dataset are sparse, and that what we can learn in terms of user behaviour can be

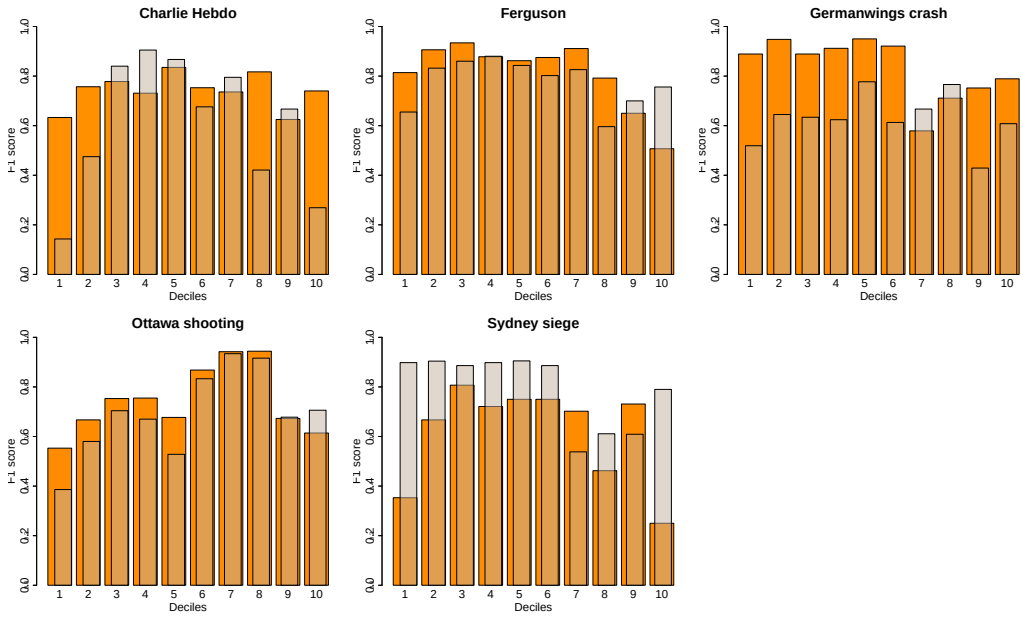

Figure 3: F1 scores by event and decile using CRF + HP (orange, thick bars) vs the best of the non-sequential classifiers in each decile (grey, thin bars).

limited when applied to new events, unless we leverage more sophisticated features.

Further, in Table 7 we look at the availability of the RumourRatio feature we use in our study. In this case, we present two statistics in separate columns:

- *Availability of RumourRatio:* the RumourRatio feature (how many of a user's previous posts were rumours) is dependent on having observed the user in the training data. This means a user in the test set must have been observed in one of the four events in the training data in order to have a RumourRatio value available. The figures on availabililty of RumourRatio indicate that, with some variability, we are only observing users in the test set in 12.6% to 41% of the cases. While these figures are not as bad for some of the events, we still miss RumourRatio values for more than half of the users in all cases.

- *Availability of FollowingRumourRatio:* when we use homophily as a feature we are trying to overcome the limited availability of RumourRatio values. When we looked at the availability of FollowingRumourRatio (the extended

	CH	F	SS	GC	OS
Charlie Hebdo	1049/1049 (100%)	– –	– –	– –	– –
Ferguson	54/1542 (3.5%)	547/547 (100%)	– –	– –	– –
Sydney Siege	142/1486 (9.6%)	31/1095 (2.8%)	579/579 (100%)	– –	– –
Germanwings Crash	76/1207 (6.3%)	22/759 (2.9%)	62/751 (8.3%)	234/234 (100%)	– –
Ottawa Shooting	56/1436 (3.9%)	21/969 (2.2%)	56/966 (5.8%)	32/645 (5.0%)	443/443 (100%)

Table 6: Overlap of unique users across pairs of events, represented as the intersection with respect to the union of users in both events.

Event	Availability of RumourRatio	Availability of FollowingRumourRatio
Charlie Hebdo	215/1049 (20.5%)	913/1049 (87.0%)
Ferguson	69/547 (12.6%)	499/547 (91.2%)
Sydney Siege	178/579 (30.7%)	476/579 (82.2%)
Germanwings Crash	96/234 (41.0%)	211/234 (90.2%)
Ottawa Shooting	77/443 (17.4%)	339/443 (76.5%)

Table 7: Users observed in training data made of other events. These represent the ratio of users with a value of RumourRatio available, as well as a FollowingRumourRatio available, which is obtained thanks to homophily.

feature that leverages RumourRatio scores from users followed by a user), we can observe that numbers increase drastically. Availability of FollowingRumourRatio, again with some variability, is substantially higher, ranging from 76.5% to 91.2%.

From always missing at least half of the users' information by using RumourRatio, the use of FollowingRumourRatio can increase the availability to having at 3/4 of the users' information in the worst case in our dataset. This reinforces the fact that the use of homophily for computing the RumourRatio score not only boosts performance, but does this by increasing the coverage of users having this feature available.

9. Discussion

In contrast with previous work in related tasks dealing with rumours, our work here has covered a wide range of rumours and non-rumours that were not necessarily known *a priori*. This was possible thanks to having as annotators a team of journalists who had followed the events closely and for the way the annotation work was performed, i.e., showing them a timeline of tweets that enabled them discovering rumours and non-rumours that one may have initially missed. The annotation has been performed in a single newsroom, and we are aware that the annotations may have slight variations across newsrooms, depending on the policies for determining when a piece of information is verified. To minimise any potential biases due to this, they carried out continual discussions and collaboration in the newsroom to come up with an agreed annotation methodology. Further studying differences across newsrooms in determining what constitutes a rumours would be ideal for future work, however it is beyond the scope of this paper. While we are confident that this approach covers a diverse range of rumours and non-rumours, one caveat that is important to note is that it is restricted to a subset of highly retweeted tweets. Consequently, our experiments have been limited to tweets being retweeted at least 100 times. This is consistent with one of the key characteristics of rumours, i.e., that they have to attract a substantial interest to be deemed rumours. While this is sensible for the task of rumour detection and the objectives of our work, 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 early detection of rumours by not having to wait for the tweets to reach a certain threshold of retweets.

Our study aims to make a focused contribution by looking at the potential of sequential classifiers for rumour detection and the assumption of homophily in rumour-sharing behaviour. The features studied in this paper are not necessarily exhaustive, and the exploration of additional features, such as user location, for further boosting performance, is left for future work. Another way of attempting to boost performance would be to look at the pairwise similarities between tweets in a timeline; while this is in part done by using sequential classifiers, use of additional similarity-based features could indeed lead to improved performance. This work also has the objective of achieving early detection of rumours, hence attempting to make these predictions before observing any replies; for an analytical look at the replies triggered by rumourous tweets, we refer the reader to [11].

Some of design decisions for our experimentation and evaluation are done in

line with the rumour resolution pipeline documented in [21]: (1) rumour detection, as the task aiming to identify unverified information and described in this paper, (2) tracking of relevant content, which can consist in clustering relevant content, (3) stance classification, which aims to determine if each of the tweets relevant to a rumour supports, denies, questions or comments on the rumour, and (4) veracity classification, which would incrementally classify the rumour-related tweets as true, false or still unverified. Given that the entire system is expected to run in real-time as new tweets come, we choose to evaluate the performance of our system at the tweet level, rather than a macro-evaluation looking at entire rumour stories, which would be done in later steps of the pipeline. The implementation of this pipeline also means that the system would flag as rumours stories that have just been verified; while a story that is just verified is no longer a rumour, the system would flag it as such, and later steps in the pipeline would then aim to predict that the rumour is now verified, with a veracity value of either true or false.

10. 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 with over five breaking news datasets collected from Twitter and annotated for rumours and non-rumours by journalists, we show that the preceding context exploited as a sequence can substantially boost the rumour detector's performance. Likewise, we have shown that we have model a user's probability for posting rumours based on their past history, which can be further enhanced by considering the rumour sharing behaviour of others in their networks; we do this by relying on the concept of homophily, i.e. users in the same network will exhibit similar patterns, which we have demonstrated for the first time that also holds true in the context of rumour sharing behaviours. While chances of having observed a user in the training data so as to determine their rumour-sharing behaviour are low, we show that the use of homophily can substantially boost our knowledge of users' rumour-sharing behaviour through the use of homophily, by improving both coverage and performance. Our approach has also proven to outperform the state-of-the-art rumour detection system introduced by [6] that, instead, 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 features 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 [4]. However, their use present 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 criminal activity [4] or disaster response [60]. Hence, it is vital that tools be developed that can aid a) the detection of rumours and b) determining their likely veracity. In the Pheme project [61], we have been developing tools that address the need for the latter [11, 62]. 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 the former problem.

The dataset employed in this research is also publicly available to enable and encourage further research in the task [63].

Acknowledgments

This work has been supported by the PHEME FP7 project (grant No. 611233). This research utilised Queen Mary's Apocrita HPC facility, supported by QMUL Research-IT. http://doi.org/10.5281/zenodo.438045

References

[1] J. Sankaranarayanan, H. Samet, B. E. Teitler, M. D. Lieberman, J. Sperling, Twitterstand: news in tweets, in: Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems, ACM, 2009, pp. 42–51.

[2] A. Zubiaga, H. Ji, K. Knight, Curating and contextualizing twitter stories to assist with social newsgathering, in: Proceedings of the 2013 international conference on Intelligent user interfaces, ACM, 2013, pp. 213–224.

[3] A. Zubiaga, Mining social media for newsgathering: A review, Online Social Networks and Media 13 (2019) 100049.

[4] R. Procter, J. Crump, S. Karstedt, A. Voss, M. Cantijoch, Reading the riots: What were the police doing on twitter?, Policing and society 23 (4) (2013) 413–436.

[5] H. Webb, P. Burnap, R. Procter, O. Rana, B. Stahl, M. Williams, W. Housley, A. Edwards, M. Jirotka, Digital wildfires: Propagation, verification, regulation, and responsible innovation, ACM Transactions on Information Systems 34 (3).

[6] Z. Zhao, P. Resnick, Q. Mei, Enquiring minds: Early detection of rumors in social media from enquiry posts, in: Proceedings of the 24th International Conference on World Wide Web, ACM, 2015, pp. 1395–1405.

[7] G. W. Allport, L. Postman, An analysis of rumor, Public Opinion Quarterly 10 (4) (1946) 501–517.

[8] N. DiFonzo, P. Bordia, Rumor, gossip and urban legends, Diogenes 54 (1) (2007) 19–35.

[9] P. Donovan, How idle is idle talk? one hundred years of rumor research, Diogenes 54 (1) (2007) 59–82.

[10] G. W. Allport, L. Postman, The psychology of rumor., Henry Holt, 1947.

[11] A. Zubiaga, M. Liakata, R. Procter, G. Wong Sak Hoi, P. Tolmie, , PLoS ONE 11 (3) (2016) 1–29. doi:10.1371/journal.pone.0150989.
URL http://dx.doi.org/10.1371%2Fjournal.pone.0150989

[12] R. Procter, F. Vis, A. Voss, Reading the riots on twitter: methodological innovation for the analysis of big data, International journal of social research methodology 16 (3) (2013) 197–214.

[13] K. Starbird, J. Maddock, M. Orand, P. Achterman, R. M. Mason, Rumors, false flags, and digital vigilantes: Misinformation on twitter after the 2013 boston marathon bombing, iConference 2014 Proceedings.

[14] A. Zubiaga, M. Liakata, R. Procter, K. Bontcheva, P. Tolmie, Crowdsourcing the annotation of rumourous conversations in social media, in: Proceedings of the 24th International Conference on World Wide Web Companion, International World Wide Web Conferences Steering Committee, 2015, pp. 347–353.

[15] M. Takayasu, K. Sato, Y. Sano, K. Yamada, W. Miura, H. Takayasu, Rumor diffusion and convergence during the 3.11 earthquake: a twitter case study, PLoS one 10 (4) (2015) e0121443.

[16] L. Tolosi, A. Tagarev, G. Georgiev, An Analysis of Event-Agnostic Features for Rumour Classification in Twitter, in: ICWSM Workshop on Social Media in the Newsroom, 2016, pp. 151–158.

[17] L. Wu, J. Li, X. Hu, H. Liu, Gleaning wisdom from the past: Early detection of emerging rumors in social media, in: Proceedings of the 2017 SIAM International Conference on Data Mining, SIAM, 2017, pp. 99–107.

[18] H. J. Oh, H. Lee, When do people verify and share health rumors on social media? the effects of message importance, health anxiety, and health literacy, Journal of health communication 24 (11) (2019) 837–847.

[19] E. Seo, P. Mohapatra, T. Abdelzaher, Identifying rumors and their sources in social networks, in: SPIE defense, security, and sensing, International Society for Optics and Photonics, 2012, pp. 83891I–83891I.

[20] T. Takahashi, N. Igata, Rumor detection on twitter, in: Soft Computing and Intelligent Systems (SCIS) and 13th International Symposium on Advanced Intelligent Systems (ISIS), 2012 Joint 6th International Conference on, IEEE, 2012, pp. 452–457.

[21] A. Zubiaga, A. Aker, K. Bontcheva, M. Liakata, R. Procter, Detection and resolution of rumours in social media: A survey, ACM Computing Surveys (CSUR) 51 (2) (2018) 32.

[22] S. Lathiya, M. Chaudhari, Rumour detection from social media: a review, International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT) 3 (7).

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

[24] S. Hamidian, M. T. Diab, Rumor detection and classification for twitter data, in: Proceedings of the Fifth International Conference on Social Media Technologies, Communication, and Informatics (SOTICS), 2015, pp. 71–77.

[25] S. Hamidian, M. T. Diab, Rumor identification and belief investigation on twitter, in: Proceedings of NAACL-HLT, 2016, pp. 3–8.

[26] A. Zubiaga, M. Liakata, R. Procter, Exploiting context for rumour detection in social media, in: International Conference on Social Informatics, Springer, 2017, pp. 109–123.

[27] J. Ma, W. Gao, K.-F. Wong, Detect rumors on twitter by promoting information campaigns with generative adversarial learning, in: The World Wide Web Conference, 2019, pp. 3049–3055.

[28] M. Abulaish, N. Kumari, M. Fazil, B. Singh, A graph-theoretic embedding-based approach for rumor detection in twitter, in: IEEE/WIC/ACM International Conference on Web Intelligence, 2019, pp. 466–470.

[29] M. Babakar, W. Moy, The State of Automated Factchecking, Tech. rep., Full Fact, London, UK (2016).

[30] N. Hassan, G. Zhang, F. Arslan, J. Caraballo, D. Jimenez, S. Gawsane, S. Hasan, M. Joseph, A. Kulkarni, A. K. Nayak, et al., ClaimBuster: the first-ever end-to-end fact-checking system, Proceedings of the VLDB Endowment 10 (12) (2017) 1945–1948.

[31] P. Gencheva, P. Nakov, L. Màrquez, A. Barrón-Cedeño, I. Koychev, A context-aware approach for detecting worth-checking claims in political debates, in: Proceedings of the International Conference Recent Advances in Natural Language Processing, INCOMA Ltd., 2017, pp. 267–276.

[32] L. Konstantinovskiy, O. Price, M. Babakar, A. Zubiaga, Towards automated factchecking: Developing an annotation schema and benchmark for consistent automated claim detection, CoRR abs/1809.08193. arXiv:1809.08193. URL http://arxiv.org/abs/1809.08193

[33] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, A. Bordes, Supervised learning of universal sentence representations from natural language inference data, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 670–680.

[34] X. Liu, A. Nourbakhsh, Q. Li, R. Fang, S. Shah, Real-time rumor debunking on twitter, in: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, ACM, 2015, pp. 1867–1870.

[35] M. Lukasik, T. Cohn, 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, 2015, pp. 2590–2595.

[36] L. Zeng, K. Starbird, E. S. Spiro, # unconfirmed: Classifying rumor stance in crisis-related social media messages, in: Tenth International AAAI Conference on Web and Social Media, 2016, pp. 747–750.

[37] S. Sun, H. Liu, J. He, X. Du, Detecting event rumors on sina weibo automatically, in: Asia-Pacific Web Conference, Springer, 2013, pp. 120–131.

[38] G. Cai, H. Wu, R. Lv, Rumors detection in chinese via crowd responses, in: Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on, IEEE, 2014, pp. 912–917.

[39] G. Liang, W. He, C. Xu, L. Chen, J. Zeng, Rumor identification in microblogging systems based on users' behavior, IEEE Transactions on Computational Social Systems 2 (3) (2015) 99–108.

[40] J. Ma, W. Gao, Z. Wei, Y. Lu, K.-F. Wong, 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, ACM, 2015, pp. 1751–1754.

[41] K. Wu, S. Yang, K. Q. Zhu, False rumors detection on sina weibo by propagation structures, in: 2015 IEEE 31st International Conference on Data Engineering, IEEE, 2015, pp. 651–662.

[42] J. Ma, W. Gao, P. Mitra, S. Kwon, B. J. Jansen, K.-F. Wong, M. Cha, Detecting rumors from microblogs with recurrent neural networks, in: 25th International Joint Conference on Artificial Intelligence, IJCAI 2016, International Joint Conferences on Artificial Intelligence, 2016, pp. 3818–3824.

[43] Z. Jin, J. Cao, Y. Zhang, J. Luo, News verification by exploiting conflicting social viewpoints in microblogs, in: Thirtieth AAAI Conference on Artificial Intelligence, 2016, pp. 2972–2978.

[44] L. Derczynski, K. Bontcheva, M. Liakata, R. Procter, G. W. S. Hoi, A. Zubiaga, SemEval-2017 Task 8: RumourEval: Determining rumour veracity and

support for rumours, in: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), 2017, pp. 69–76.

[45] E. Kochkina, M. Liakata, A. Zubiaga, All-in-one: Multi-task learning for rumour verification, in: Proceedings of the 27th International Conference on Computational Linguistics, 2018, pp. 3402–3413.

[46] G. Gorrell, E. Kochkina, M. Liakata, A. Aker, A. Zubiaga, K. Bontcheva, L. Derczynski, Semeval-2019 task 7: Rumoureval, determining rumour veracity and support for rumours, in: Proceedings of the 13th International Workshop on Semantic Evaluation, 2019, pp. 845–854.

[47] C. Silverman, Verification handbook, The European Journalism Centre (EJC);, 2013.

[48] C. Silverman, Lies, damn lies, and viral content: How news websites spread (and debunk) online rumors, unverified claims and misinformation, Tech. rep., Tow Center for Digital Journalism (2015).

[49] M. McPherson, L. Smith-Lovin, J. M. Cook, Birds of a feather: Homophily in social networks, Annual review of sociology 27 (1) (2001) 415–444.

[50] W. Shrum, N. H. Cheek Jr, S. MacD, Friendship in school: Gender and racial homophily, Sociology of Education (1988) 227–239.

[51] L. M. Aiello, A. Barrat, R. Schifanella, C. Cattuto, B. Markines, F. Menczer, Friendship prediction and homophily in social media, ACM Transactions on the Web (TWEB) 6 (2) (2012) 1–33.

[52] E. Colleoni, A. Rozza, A. Arvidsson, Echo chamber or public sphere? predicting political orientation and measuring political homophily in twitter using big data, Journal of communication 64 (2) (2014) 317–332.

[53] S. A. Macskassy, M. Michelson, Why do people retweet? anti-homophily wins the day!, in: Fifth International AAAI Conference on Weblogs and Social Media, 2011, pp. 209–216.

[54] J. Lafferty, A. McCallum, F. Pereira, Conditional random fields: Probabilistic models for segmenting and labeling sequence data, in: Proceedings of the eighteenth international conference on machine learning, ICML, Vol. 1, 2001, pp. 282–289.

[55] C. Sutton, A. McCallum, An introduction to conditional random fields, Machine Learning 4 (4) (2011) 267–373.

[56] A. C. Müller, S. Behnke, Pystruct: learning structured prediction in python, The Journal of Machine Learning Research 15 (1) (2014) 2055–2060.

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

[58] K. Bontcheva, L. Derczynski, A. Funk, M. A. Greenwood, D. Maynard, N. Aswani, TwitIE: An open-source information extraction pipeline for microblog text, in: Proceedings of the International Conference on Recent Advances in Natural Language Processing, 2013, pp. 83–90.

[59] H. Cunningham, D. Maynard, K. Bontcheva, Text processing with gate, Gateway Press CA, 2011.

[60] G. Bazerli, T. Bean, A. Crandall, M. Coutin, L. Kasindi, R. N. Procter, S. Rodger, D. Saber, L. Slachmuijlder, T. Trewinnard, Humanitarianism 2.0, Global Policy Journal.

[61] L. Derczynski, K. Bontcheva, M. Lukasik, T. Declerck, A. Scharl, G. Georgiev, P. Osenova, T. P. Lobo, A. Kolliakou, R. Stewart, 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.

[62] M. Lukasik, K. Bontcheva, T. Cohn, A. Zubiaga, M. Liakata, R. Procter, Using gaussian processes for rumour stance classification in social media (2016).

[63] A. Zubiaga, G. W. S. Hoi, M. Liakata, R. Procter, Pheme dataset of rumours and non-rumoursdoi:10.6084/m9.figshare.4010619.v1.
URL https://figshare.com/articles/PHEME_dataset_of_rumours_and_non-rumours/4010619