Context-Aware Message-Level Rumour Detection with Weak Supervision

Sooji Han

The University of Sheffield

Submitted to the Department of Computer Science
of the University of Sheffield
in partial fulfillment of the requirements
for the degree of *Doctor of Philosophy*

July 2020

Sooji Han: *Context-Aware Message-Level Rumour Detection with Weak Supervision,*
July 2020

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere gratitude to my PhD supervisor Fabio Ciravegna for his support and guidance throughout my PhD study, valuable and useful feedback on this thesis, and funding my research during the writing-up phase. The enthusiasm and passion he has for his research has been motivational me especially during tough times of my PhD. I also appreciate his giving me an opportunity to continue working as a Research Associate in the OAK group after submitting my thesis. My thanks also go to my dissertation examiners Dr Nikos Aletras and Professor Harith Alani for their constructive and professional feedback and advice. I would also like to thank the Korean government for funding my research for the first three years of my PhD.

I am grateful to Jie Gao for sharing his expertise on natural language processing and deep learning, collaborations, and inspiring me to come up with better research plans. I also thank Dr Vitaveska Lanfranchi and Dr Andreas Vlachos for their useful feedback and suggestions on my research during the initial stages of my PhD.

Special thanks go to Dr Bartosz Reguła for sharing emotional support, motivating and inspiring me to progress my career and giving me constructive advice and feedback.

My thanks also go to the members of the OAK group for warm friendships and constructive advice and discussions. I am grateful for Neil Ireson for personal and professional conversations we have had. I would also like to thank Dr Lubos Vaci for bringing fun and joy to the group and inspirational discussions on common research interests.

Last but not the least, I would like to express my warmest thanks to my family and friends for their love, support and encouragement.

ABSTRACT

Social media has become the main source of all sorts of information beyond a communication medium. Its intrinsic nature can allow a continuous and massive flow of misinformation to make a severe impact worldwide. In particular, *rumours* emerge unexpectedly and spread quickly. It is challenging to track down their origins and stop their propagation. One of the most ideal solutions to this is to identify rumour-mongering messages as early as possible, which is commonly referred to as "*Early Rumour Detection (ERD)*". This dissertation focuses on researching ERD on social media by exploiting weak supervision and contextual information. *Weak supervision* is a branch of Machine Learning (ML) where noisy and less precise sources (e.g. data patterns) are leveraged to learn limited high-quality labelled data (Ratner et al., 2017). This is intended to reduce the cost and increase the efficiency of the hand-labelling of large-scale data. This thesis aims to study whether identifying rumours before they go viral is possible and develop an architecture for ERD at *individual post level*. To this end, it first explores major bottlenecks of current ERD. It also uncovers a research gap between system design and its applications in the real world, which have received less attention from the research community of ERD. One bottleneck is limited labelled data. Weakly supervised methods to augment limited labelled training data for ERD are introduced. The other bottleneck is enormous amounts of noisy data. A framework unifying burst detection based on temporal signals and burst summarisation is investigated to identify potential rumours (i.e. input to rumour detection models) by filtering out uninformative messages. Finally, a novel method which jointly learns rumour sources and their contexts (i.e. conversational threads) for ERD is proposed. An extensive evaluation setting for ERD systems is also introduced.

PUBLICATIONS

The research documented in this thesis has been published in conference proceedings and is entirely my own.

- **Chapter** 4 is based on publications at the workshop on Learning from Limited Labelled Data (ICLR 2019) at the 7th International Conference on Learning Representations (Han et al., 2019a) and in the proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (Han et al., 2019b).

- **Chapter** 5 is based on a publication in the proceedings of the 16th International Conference on Information Systems for Crisis Response And Management (Han and Ciravegna, 2019).

- **Chapter** 6 is based on a publication in the proceedings of the 12th Language Resources and Evaluation Conference (Gao et al., 2019). I share co-first authorship with Jie Gao and we equally contributed the work on this paper. The two authors' names are listed in alphabetical order in Gao et al. (2019).

I am the first author of all the four publications. I designed research; collected and processed data; developed solutions; carried out experiments and evaluation; and performed all of the analyses. I was supported by the co-authors for discussions and feedback on writing.

REFERENCES

Gao, Jie, Sooji Han, Xingyi Song, and Fabio Ciravegna (May 2020). "RP-DNN: A Tweet Level Propagation Context Based Deep Neural Networks for Early Rumor Detection in Social Media." In: *Proceedings of The 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, pp. 6094–6105. URL: https://www.aclweb.org/anthology/2020.lrec-1.748.

Han, Sooji and Fabio Ciravegna (May 2019). "Rumour Detection on Social Media for Crisis Management." In: *Proceedings of the 16th International Conference on Information Systems for Crisis Response And Management*. ISCRAM, pp. 660–673. URL: https://www.researchgate.net/publication/332513623_Rumour_Detection_on_Social_Media_for_Crisis_Management.

Han, Sooji, Jie Gao, and Fabio Ciravegna (May 2019a). "Data augmentation for rumor detection using context-sensitive neural language model with large-scale credibility corpus." In: *Proceedings of the 7th International Conference on Learning Representations. Learning from Limited Labeled Data: ICLR 2019 Workshop*. OpenReview. URL: http://eprints.whiterose.ac.uk/145668/.

Han, Sooji, Jie Gao, and Fabio Ciravegna (2019b). “Neural language model based training data augmentation for weakly supervised early rumor detection.” In: *Proceedings of 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. IEEE. URL: https://arxiv.org/abs/1907.07033.

CONTENTS

IV APPENDIX

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

SOTA	State-Of-The-Art
ML	Machine Learning
NLP	Natural Language Processing
DNN	Deep Neural Network
NLM	Neural Language Model
NN	Neural Network
CNNs	Convolutional Neural Networks
RNNs	Recurrent Neural Networks
LSTM	Long shor-term memory
LOOCV	Leave-One-Out-Cross-Vaildation
BiLM	Bidirectional Language Model
REGEX	Regular Expression
SVM	Support Vector Machine
ERD	Early Rumour Detection
LM	Language Model
ELMo	Embeddings from Language Models
CV	Cross Validation
SC	source tweet contents
CC	context contents
CM	context metadata

Part I

1 INTRODUCTION

1.1 PROBLEM STATEMENT

Social media platforms are the main sources of a variety of information with rapidly growing rates of user engagement and global mobile social media usage. According to the Global Digital Report 2019 [1], 3.48 billion users (i.e. 45% of the world's population) are using social media. People share their own opinions, emotions, and beliefs on social media where they also find a variety of information and opinions regarding an event of interest. In particular, *Twitter*[2] is one of the most representative social media platforms. Users post *tweets* to report on real-world events on Twitter. Social media allow users to diffuse information firsthand. Many users use social media as the main source of information, effectively replacing traditional mass media sources (Ingram, 2016; Ries et al., 2018). As this phenomenon emerges in diverse areas of our lives such as journalism, marketing, politics, and economy (Derczynski and Bontcheva, 2014; Zheltukhina et al., 2016; Jin et al., 2017c), decision-makers and citizens use social media to better understand unfolding events in real life. For example, the role of social media during emergencies and crises has become prominent (Andrews et al., 2016; Arif et al., 2017; Castillo, 2016; Rudra et al., 2018; Rudra et al., 2016; Imran et al., 2015; Rudra et al., 2015). Emergency services can remotely identify areas affected by crises situations based on social media users' posts reporting what they are seeing and hearing (Yin et al., 2012) or find victims seeking help (Zubiaga et al., 2018a). Emergency responders can then make adequate decisions such as the allocation of resources and police.

At the same time, however, concerns about the adverse impact of online *rumours* have been raised worldwide. An early study in social psychology defines a rumour as "a proposition for the belief of topical reference disseminated without official verification (Knapp, 1944)". Social media are origins of rumours and where they spread among a large number of people (Ma et al., 2018b). Rumours tend to spread very quickly and unexpectedly throughout social media (Doerr et al., 2012; Stewart et al., 2016). The emergence and propagation of rumours, particularly false or misleading viral reports, can endanger the well-being of individuals, society, and the economy (Spiegel et al., 2010; Matthews, 2013). In the case of breaking news events such as terrorist attacks, for example, major rumours are related to situational awareness such as victims, suspects, and locations of events (Zubiaga et al., 2016a).

Research on rumours on social media has become increasingly popular to understand their emergence and development and to prevent and resolve problems posed by them. A typical resolution process can include four sub-tasks: rumour detection, tracking, stance classification, and verification (Zubiaga et al., 2018a). First of all, *rumour detection* aims to identify whether a piece of information is a rumour or non-rumour. This task is typically

[1] https://wearesocial.com/global-digital-report-2019, accessed on 27 December 2019

[2] https://twitter.com/

formulated as a binary classification problem. Input is social media posts, and a classifier determines each post's label (i.e. positive/negative) based on a set of features. This sub-task is essential for identifying new rumours. Early research on rumour detection focussed on exploring novel hand-crafted features that provide the optimum representation of rumours (Qazvinian et al., 2011; Kwon et al., 2017; Kwon et al., 2013; Yang et al., 2012; Sun et al., 2013; Zhao et al., 2015; Zhang et al., 2015c; Wu et al., 2015; Ma et al., 2015; Liu et al., 2016; Hamidian and Diab, 2015; Hamidian and Diab, 2016). Recent research streams report that feature engineering is labour-intensive and time-consuming and have attempted to leverage deep neural networks which require little or no feature engineering (Chen et al., 2018; Jin et al., 2017b; Ma et al., 2018b; Ma et al., 2016; Ma et al., 2018a; Ruchansky et al., 2017; Nguyen et al., 2017; Yu et al., 2017). Results obtained by deep learning-based methods show improvements over results achieved using methods based on feature engineering. Secondly, *rumour tracking* aims to collect posts related to identified rumours. This task can be extended to studies on the temporal development of rumours and the subsequent sub-task in the rumour resolution system, that is, *stance classification*. Its purpose is to assign stance labels (i.e. opinions) such as deny, question, and support to posts related to rumours (i.e. the output of rumour tracking). Rumour stances can be leveraged by the task of *rumour verification*. It refers to the determination of the truthfulness of rumours such as "True", "False", and "Unverified". This task is commonly formulated as a multi-class classification problem and the most difficult task among the four sub-tasks in the rumour resolution process.

This thesis focuses on the first component (i.e. rumour detection). Specifically, the early identification of newly emerging rumours during *breaking news events* such as terrorist attacks and hostage-takings is studied. Rumours that appear during breaking news events are usually event-specific and novel in terms of contents. Therefore, it is impossible to proceed to subsequent sub-tasks in the rumour resolution process without first identifying them. Detecting emerging rumours as early as possible during time-sensitive situations is crucial for not only decision-makers such as emergency responders and journalists, but also the public as it is the very first step to be done to minimise adverse effects of rumours (e.g. false beliefs and myths; unnecessary public expenditure on research and public campaigns aimed at debunk them; and biased public opinion on political and societal decisions (Lewandowsky et al., 2012)). For example, an agency of the United States called The Federal Emergency Management Agency (FEMA)[3] creates rumour control pages when natural disasters occur so as to avoid spreading false information. Without such attempts to identify rumours during the early stages of their evolution, citizens can be confused by unfounded or conflicting rumours regardless of whether they were generated purposely or unintentionally. Despite a recent rise in the popularity of research on rumours on social media, several challenges have yet to be solved.

Firstly, ***limited labelled data*** poses a challenge to rumour detection. Although a large amount of data is available, the manual annotation of data for the rumour detection task is highly laborious (Zubiaga et al., 2016a).

[3] https://www.fema.gov/

Therefore, SOTA rumour detection methods have to rely on publicly available rumour data sets which consist of a limited number of social media posts compared to data generated during real-world events. This challenge is more problematic for Deep Neural Networks. They require large-scale hand-labelled training data. To better exploit the advantages of deep learning algorithms in rumour detection, it is required to address labelled data scarcity.

Secondly, publicly available rumour data suffer from ***imbalanced class distributions*** (Kochkina et al., 2018a). Existing methods for handling class imbalance (e.g. oversampling and the use of synthetic data (Xu and Chen, 2015)) may cause overfitting and poor generalisation performance. A new methodology for rumour data augmentation with the minimum of human supervision is necessary.

Another challenge is that analysing every social media post published during breaking news events is not viable due to ***the rapid speed, vast volume, and noise of data*** generated by users with ambiguous authorship and uncertain authenticity. A classifier classifies *each individual post* into two groups–rumour and non-rumour–in a typical message-level rumour detection system. It is arguable whether applying message-level rumour detection models to real-world events is practically feasible and useful. To address this issue, existing work (Zubiaga et al., 2016b) selects candidate tweets based on their popularity which is often represented by the number of reposts. This approach can inhibit detecting rumours as early as possible because some rumours do not get much attention in their early stage. To overcome this limitation, new methods for efficiently selecting high-quality candidates (i.e. input to rumour detection models) are needed.

Finally, ***using each social media post in isolation as a unit of analysis*** has limited potential to advance SOTA performance on rumour detection. Most existing approaches for rumour detection are limited to individual source tweets rather than taking *contexts* surrounding them into account. *Source tweets* refer to tweets that initiate a new Twitter conversation (i.e. not replying to existing tweets; Hoi (2015)). Although not much work has so far exploited contextual information for rumour detection, there have recently been a few attempts (Kochkina et al., 2018a; Ma et al., 2018b). In the case of message-level rumour detection, *contextual information* typically refers to information obtained from conversational threads of source tweets in Twitter. Social media messages, particularly tweets, are short and contain very limited context on their own. Conversational threads can provide an understanding of propagation patterns, which are different between rumour and non-rumours, as well as users' reactions to rumours. Consequently, using them helps rumour detection models better understand what distinguishes rumours from non-rumours (Zubiaga et al., 2018b). Before attempts to incorporate them into rumour detection, most early work on rumour detection used metadata provided by social media APIs in the hope that it can provide contextual information in which rumours spread. For example, Twitter provides metadata such as the number of times a tweet has been retweeted and favoured by other users. This information is often used as hand-crafted features indicating the popularity of tweets in rumour studies. However, such metadata is not enough to characterise online rumours. For instance, tweets with the same number of retweets can display very different propagation patterns (Meyer, 2018). Propagation patterns play an important role in rumour detection as variants of rumours share similar spreading patterns distinguished from those of non-rumours (Liu et al., 2017; Kwon et al., 2017). Moreover, the development and behaviour of rumours on social media are strongly related to how users react to them (see details in Section 2.1.3). Considering the impact of contextual information on the characterisation of online rumours, it is required to investigate how to design and exploit the propagation structure of conversational threads for the task of ERD.

1.2 RESEARCH QUESTIONS

The primary research question for ERD is "Is it possible to identify rumours on social media before they become viral?" To address it, this thesis identifies research questions in respective of ERD to address the current challenges described in the previous section.

Research questions related to limited labelled data and class imbalance:

RQ 1.1 To what extent could the size of existing training data for rumour detection be extended with approaches based on semantic relatedness?

RQ 1.2 Does fine-tuning a SOTA Neural Language Model (NLM) using a domain-specific corpus improve representations of rumours?

RQ 1.3 Does data augmentation improve the performance of deep learning-based ERD architectures? How can this be assessed?

Research questions related to large-scale data reduction:

RQ 2.1 What are signals which characterise rumours in the early stages of their evolution?

RQ 2.2 How can candidates for rumours (i.e. potential rumours) be selected with minimal human supervision and time delay?

Research questions related to ERD:

RQ 3.1 What contextual information can be leveraged into deep learning-based ERD? How can they be obtained and learnt?

RQ 3.2 How can rumour detection architectures be evaluated in realistic scenarios in which detection models are required to identify unseen rumours?

1.3 METHODOLOGY OVERVIEW AND EXPERIMENT DESIGN

An overview of research design for message-level rumour detection with weak supervision is given in Figure 1.1. This thesis studies three topics intending to propose an end-to-end framework for ERD applicable to real-world problems which are accompanied by several challenges such as the fast speed, enormous volume, and noise of social media data. There are two

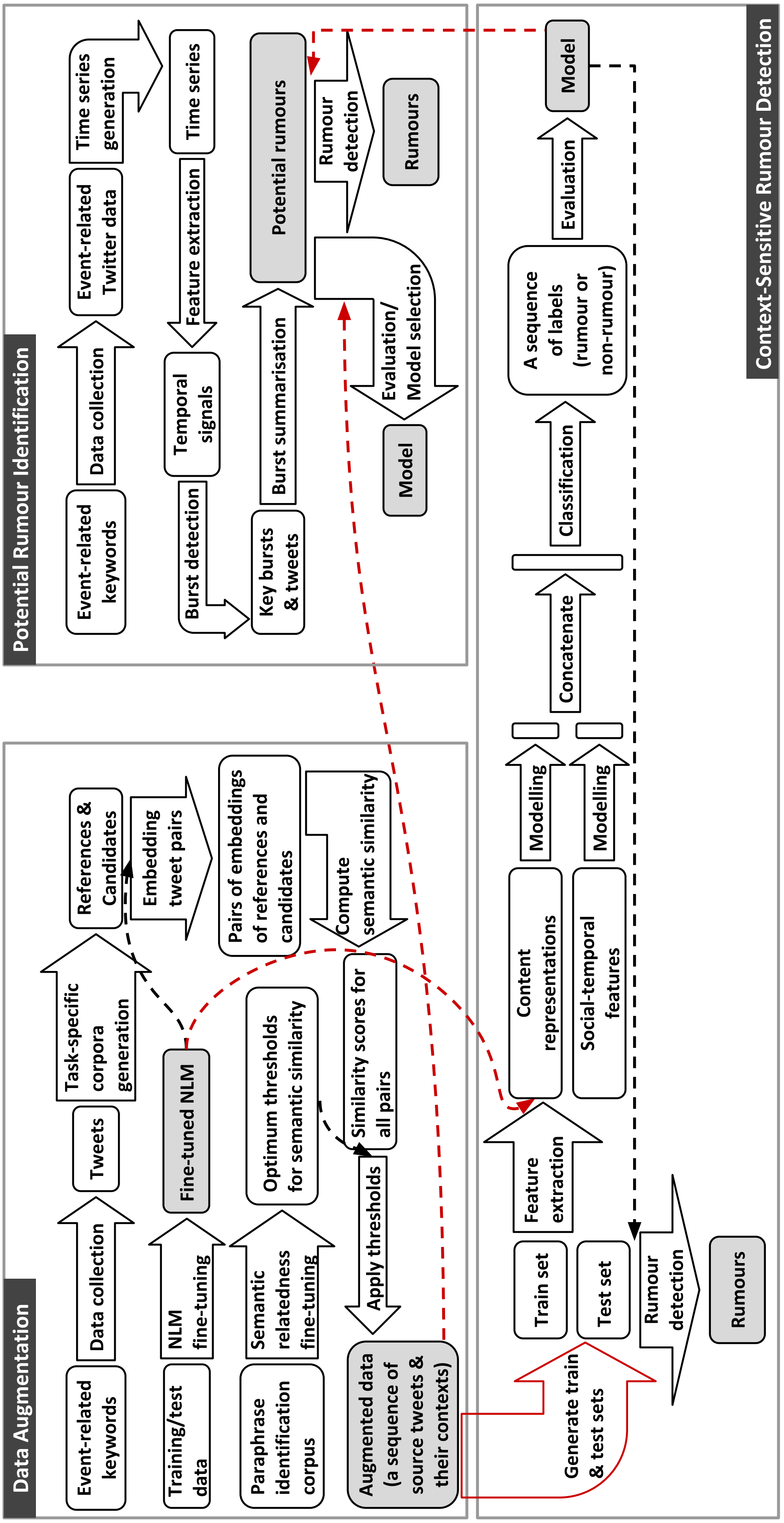


Figure 1.1: Overview of the proposed methodology for data augmentation.

strands of approaches for rumour detection. One is message-level rumour detection, which aims to classify whether a post is a rumour or non-rumour. The other is event-level rumour detection, which identifies whether an event, represented by a collection of relevant tweets, is related to a rumour or not. This thesis aims at *message-level* rumour detection.

Figure 1.1 illustrates how the three topics fit together. The first component of the framework is the augmentation of training data based on semantic relatedness between limited labelled data (i.e. references) and unlabelled data (i.e. candidates). In other words, it increases the training corpus with tweets taken from a large unlabelled corpus by annotating them using pairwise similarity with rumour stories which are part of a manually annotated corpus. The data augmentation framework includes two preliminary tasks: *NLM fine-tuning* and *semantic relatedness fine-tuning*. A SOTA NLM is fine-tuned on a large-scale corpus which contains tweets annotated with credibility ratings to get tweet representations effective for rumour detection. Semantic relatedness fine-tuning is performed in order to decide two thresholds for selecting rumour and non-rumour source tweets out of a set of unlabelled candidate tweets, respectively. For this task, a corpus designed for paraphrase identification and semantic similarity measurement is used. Using the fine-tuned NLM and two thresholds, data augmentation is performed. The output of data augmentation is rumour and non-rumour source tweets annotated with weak supervision and their contexts (i.e. retweets and replies). The effectiveness of augmented data is evaluated in the context of ERD using a SOTA deep learning-based rumour detection model. Evaluation metrics used include F1-score, precision, and recall.

The second component is the identification of candidate tweets for rumour detection (i.e. tweets which will be input to a rumour detection model) via key burst detection and summarisation. Key burst detection aims to detect bursts in the evolution of an event solely based on temporal signals. Given detected key bursts, a summarisation method is employed to rank tweets posted during each burst in order of significance. The top N most important tweets are included in a set of potential rumours (i.e. the final output). The output of potential rumour identification can be used in several ways. For instance, in a general-purpose application, generated summaries can help practitioners in several domains (e.g. journalists and emergency responders) efficiently and effectively understand trending topics regarding an event of interest. In a domain-specific application, generated summaries can be used as input to a tweet-level rumour detection model as illustrated in Figure 1.1. This thesis proposes a novel evaluation approach for the framework which measures the quality of detected bursts and generated summaries in terms of ERD. The augmented data with weak labels (i.e. annotations obtained via weak supervision) is used to to guide evaluation.

The last component is context-aware rumour detection. Train, hold-out, and test sets are formed as temporally ordered sequences of source tweets. For the representation of tweet contents, the NLM fined-tuned in the data augmentation framework is employed. For the representation of social-temporal contexts of rumours and non-rumours, hand-crafted features are extracted from replies of rumour and non-rumour source tweets. Two complementary representations are learnt separately using two different Deep Neural Network (DNN)s and learnt representations are concatenated in order to determine labels (rumour or non-rumour) of source tweets. Evaluation is performed based on F1-score, precision, and recall.

1.4 CONTRIBUTION

The contributions of this thesis are to research methods for developing an entirely automated message-level ERD using weak supervision and contextual information. This section summarises them to offer a brief insight about how they are investigated.

To address labelled data scarcity and class imbalance which hinder achieving the full potential of deep learning techniques, data augmentation is researched in Chapter 4. Unlike current artificial data augmentation methods based on modifications to existing data or reliance on limited knowledge bases, the method proposed in this thesis uses large-scale, real-world social media data. It can not only increase the amount of training data but most importantly help to increase the diversity of the original data.

One approach for the second challenge (i.e. the infeasible analysis of an entire corpus of social media messages) is to automatically reduce data by eliminating less significant data in the initial stages of data analysis (Sharifi et al., 2013). This can often be done by producing summaries which offer insights about further data exploration (Sharifi et al., 2013). This thesis argues that a preliminary step for selecting candidates for rumours (i.e. *potential rumours*) is crucial to demonstrate the true and practical value of rumour detection in real-world applications (Hoi, 2015). The term "potential rumours" refers to claims which 1) are the centre of attention and 2) should be further examined by human experts or analysed by automated rumour detection models to be confirmed as rumours. To this end, Section 2.5 studies candidates for early signals for rumours and identifies requirements for methods for identifying potential rumours. Chapter 5 investigates a framework involving key burst detection and summarisation.

As described in the previous section, contextual data provides richer representations of tweets bearing rumours. However, only a few recent studies (Lukasik et al., 2015; Zubiaga et al., 2017; Kochkina et al., 2018a; Nguyen, 2017; Tarnpradab and Hua, 2019) have exploited conversational context for message-level rumour detection. Little work has examined a DNN architecture taking different types of contextual information as a part of input in addition to contents. Chapter 6 proposes a context-aware DNN framework which performs rumour classification at individual message level. It jointly learns contents and contexts of input source messages. Unlike existing studies relying on limited manually labelled training data, Chapter 6 employs large training sets generated via weak supervision.

1.5 THESIS STRUCTURE

The remainder of this thesis is structured as follows:

Chapter 2 describes the task of rumour detection. Different definitions of rumour, characteristics of rumours, and their propagation on social media are described. It also investigates several features used to characterise rumours on social media in order to identify early signals for rumours. Related research, research gaps and its limitations are discussed in depth. These discussions and analysis motivate the research presented in this thesis.

Chapter 3 describes the *aim and objectives* of the research conducted in this thesis as well as an *overview* of the rumour detection methodology of this thesis.

Chapter 4 proposes novel *data augmentation* strategies based on a SOTA NLM and semantic relatedness in order to increase the size of labelled training data for rumour detection methods. This chapter shows that data augmentation helps to improve the performance of a SOTA DNN rumour detection model by addressing limited labelled data and class invariances in existing publicly available rumour data sets. This chapter is based on a publication in the proceedings of the 7th International Conference on Learning Representations (Han et al., 2019a) and that in the proceedings of 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (Han et al., 2019b).

Chapter 5 proposes a novel data reduction framework involving *key burst detection* based on temporal signals and *burst summarisation* via text ranking algorithms in order to select candidates (i.e. potential rumours) for feasible ERD. This chapter shows that the proposed methodology can help to efficiently and effectively identify newsworthy stories and rumours during breaking news events, and hence ERD can benefit from the proposed framework. This chapter is based on a publication in the proceedings of the 16th International Conference on Information Systems for Crisis Response And Management (Han and Ciravegna, 2019).

Chapter 6 proposes a solution for the problem of ERD in which training data augmented based on semantic relatedness (i.e. *weak supervision*) contains noise. A novel Neural Network (NN) architecture which benefits from the inclusion of hand-crafted features and SOTA sentence embeddings is proposed. This chapter shows that combining social-temporal context information with linguistic features can effectively model salient features of rumours, and hence the proposed solution advances the SOTA. This chapter is based on my work which is in the proceedings the 12th Edition of its Language Resources and Evaluation Conference.

Chapter 7 summarises the research of this thesis and presents future work.

Part II

2 BACKGROUND ON RUMOUR DETECTION

Understanding and uncovering rumours and how they spread have been of practical and theoretical interest to psychologists, sociologists, computer scientists, historians, and journalists. Chapter 1 outlined the research problems and questions, methodology, and contributions of this thesis. This chapter now details background on online rumours and rumour detection. It first introduces different definitions of rumour proposed by researchers in social science and computer science as well as major dictionaries. Exploring several definitions helps understanding what they are from different perspectives. This chapter then presents their characteristics and spreading on social media. They are often characterised by social media users' reactions to them and factors influencing their diffusion. The exploitation of features that effectively distinguish them from non-rumours is the key to the performance of a rumour detection method. Next, an overview of the task of rumour detection is proposed. This chapter also contains related work for rumour detection and its limitations.

2.1 DEFINITION AND CHARACTERISTICS OF RUMOUR

2.1.1 *Rumour Definition*

One of the most important preliminaries in studying rumours is to answer the question, "What is a rumour?". Different researchers have given different definitions and its true meaning is disputed. This thesis compares diverse definitions.

Rumours defined in social psychology

In *social psychology*, for example, a rumour has been defined as "a proposition for belief of topical reference disseminated without official verification (Knapp, 1944)", "an unverified account or explanation of events, circulating from person to person and pertaining to an object, event, or issue of public concern (Peterson and Gist, 1951)", and "a specific (or topical) proposition for belief, passed along from person to person, usually by word of mouth, without secure standards of evidence being present (Allport and Postman, 1965)". More recently, it is defined as "an unverified and instrumentally relevant information statement in circulation that arises in contexts of ambiguity, danger or potential threat, and that functions to help people make sense and manage risk" (DiFonzo and Bordia, 2007).

Rumours defined in major dictionaries

Major *dictionaries* provide their definitions. For example, the Oxford English Dictionaries defines it as "a currently circulating story or report of uncertain or doubtful truth" [1] and the Merriam Webster dictionary defines it as "a statement or report current without known authority for its truth" [2].

Rumours defined in social media studies

The majority of *studies on rumours on social media* define it as a piece of information that is *unverified* at the time of posting, which is consistent with

[1] https://en.oxforddictionaries.com/definition/rumour
[2] https://www.merriam-webster.com/dictionary/rumor

those given by the major dictionaries (Zubiaga et al., 2018a). For example, recent work defines it as “a controversial and fact-checkable statement” (Zhao et al., 2015). Some work proposes a similar but more detailed definition such as “a circulating story of questionable veracity, which is apparently credible but hard to verify, and produces sufficient scepticism and/or anxiety so as to motivate finding out the actual truth” (Damoulas, 2015; Zubiaga et al., 2015). Several researchers have focused on their falsity. For instance, it is “a statement whose truth value is unverified or deliberately false” (Qazvinian et al., 2011; Dayani et al., 2015). False claims are statements which are proven to be false in the end, and they can be categorised into four types: rumours, disinformation (manipulated false information), misinformation (accidental false information), and speculation (Derczynski and Bontcheva, 2014). False rumours are defined as baseless statements which emerged during the crisis and are confirmed to be false at some point (Mendoza et al., 2010). A more general definition of rumour is “any information that is circulating on social media and is incompatible with information from credible sources” (Jain et al., 2016).

Based on a thorough search of related work, this thesis adopts the following definition: “information of unverified veracity which is appealing enough to make the public or different social groups become sceptical, doubtful, or supportive about its credibility and eager to spread, verify, or debunk it”. This definition reflects diverse characteristics and users’ behaviour which have been proposed by research in social sciences and the majority of rumour studies on social media.

2.1.2 *Types of Rumour Detection*

Message-level rumour detection

There are two strands of approaches for rumour detection. This thesis studies message-level rumour detection. One is ***message-level*** rumour detection, which aims to classify whether every single post in an input corpus is a rumour or non-rumour. In general, the objective of this type is to identify several different sub-events (i.e. rumours) which belong to an event. Here, an event itself is neutral and does not represent a rumour or non-rumour. Input to message-level detection models is any type of messages related to an event which potentially produces several rumours. For example, on 7 January 2015, an event commonly referred to as “Charlie Hebdo shooting” took place. Two armed Muslim brothers forced their way through the offices of a French satirical newspaper, Charlie Hebdo, in Paris, France. During this event, a wide range of rumours emerged and spread worldwide. Appendix a.1 shows examples. In this example,“Charlie Hebdo shooting” is the event of interest and it is not possible to classify it into a rumour or non-rumour because the scope of related messages is not limited to rumours. Refer to a publicly available rumour data set called *PHEME (6392078*; Kochkina et al. (2018a) and Zubiaga et al. (2016a)) in Section 2.4.1 for more examples of events in message-level classification. Most publicly available data sets for this type of rumour detection use general keywords (e.g. “#charliehebdo”, “#jesuischarlie”, “charlie hebdo”, and “paris”) to collect tweets. Table 4.2 shows more examples.

Event-level rumour detection

Event-level rumour detection aims to identify whether an event is related to a rumour or not given a collection of messages related to it. A major difference between this and message-level detection is that an event in the former is a specific rumour. For example, a rumour that a singer Prince would perform a secret show in Toronto, on 4 November, 2014, started circulating the day before. This rumour itself is referred to as an event in event-level detection. Input to models is messages about the event (e.g. messages supporting, denying, and questioning about the event). This thesis refers to these types of events as *preselected rumours*. Refer to the *PHEME (6392078;* Kochkina et al. (2018a) and Zubiaga et al. (2016a)) and Qazvinian data (Qazvinian et al., 2011) described in Sections 2.4.1 and 2.4.4 for more examples. For data collection, a event-specific query can be defined. For example, a query "`Obama & (muslim|islam)`" can be used to collect message related to a rumour event "Is Barak Obama Muslim?".

This thesis studies message-level rumour detection.

2.1.3 *Rumour Characteristics*

A wide range of work on rumours on social media has studied their emergence and evolution over time in order to characterise them and their diffusion on social media. Some of this research has focused on people's reactions, while others have studied temporal, structural and linguistic patterns of rumour evolution.

Studies that characterise rumours by exploring users' reactions (i.e. stances) to rumours.

Several studies have found that the development of online rumours is deeply related to variations in human activities over time. This finding has motivated researchers to analyse discussions among users and to study patterns of people's reactions to rumours and non-rumours. Understanding how individuals react to rumours can provide important cues to uncover properties seemingly different rumours have in common and features that distinguish rumours from non-rumours. Different users take different actions when they encounter a rumour on social media. Some users may blindly believe and repost news if it was received by credible sources or based on their existing beliefs. Others may look for external sources to find evidence to understand, verify, or dispel new information. Identifying types of reactions expressed towards rumours has been actively studied as it helps to characterise them.

Qazvinian et al. (2011)

Castillo et al. (2013)

Maddock et al. (2015)

An early study (Qazvinian et al., 2011) shows that linguistic features of tweets such as *n*-grams (i.e. a sequence of *n* adjacent words in a given text) and part-of-speech (i.e. categories of lexical units) are useful cues to identify users' reactions. In this work, users' beliefs are classified in two groups: *believe* versus *deny/question/neutral*. Another study (Castillo et al., 2013) categorises users' reactions that appear during crises into four categories: *affirming*, *denying*, *questioning*, and *unrelated/unknown*. Two findings of a case study on a real-world crisis event are as follows: 1) false rumours are more likely to be questioned than true rumours, and 2) the ratio of denying reactions to affirming reactions was nearly one to one. The veracity of rumours (i.e. whether a rumour is proven to be false or true) can be verified in the late stages of their diffusion based on evidence. Maddock et al. (2015) study patterns of their propagation during crises and identify seven behavioural types of reactions: *misinformation*, *speculation*, *correction*, *question*, *hedge*, *unrelated*, and *neutral/others*. Unlike (Castillo et al., 2013), however, this study does not analyse differences in user reactions between false and true rumours.

Zubiaga et al. (2016a)

Zubiaga et al. (2016a) undertake an extensive analysis of stances towards various newsworthy events and roles of different types of users in rumour spreading. The authors find that users tend to support rumours in the early stages of their diffusion rather than denying and refuting them regardless of the level of veracity. This is due to a lack of counter-evidence, which makes debunking them more challenging than verifying them. Another notable finding is that it is more challenging to determine the veracity of a false rumour than a true rumour. It takes about 13-14 hours to debunk false information on average, while it only takes a couple of hours for true rumours to be verified. Li et al. (2016) study users' beliefs shown in a large number of false rumours, observe how they evolve, and characterise roles of different types of users in the propagation of false rumours. The authors find that people generally tend to disseminate rumours without expressing their beliefs when they lack evidence to verify them (Buckner, 1965), and that users supporting them make up the smallest proportion of all users when they have not been verified or debunked.

Li et al. (2016)

Mendoza et al. (2010)

Another work (Mendoza et al., 2010) on tweets and Twitter users during an earthquake presents that the majority of users tend to support true rumours, while users are more likely to deny and question false rumours.

Studies that characterise rumours by exploring factors affecting their evolution.

Another approach to the characterisation of rumours is the exploitation of temporal, structural, and/or linguistic properties of their propagation. Findings of rumour studies in social sciences and psychology (Bordia and Rosnow, 1998; Rosnow, 1991; Shibutani, 1966; DiFonzo and Bordia, 2007; DiFonzo and Bordia, 2002; Bordia et al., 1999; Sunstein, 2010) have inspired and given a fresh insight into such research.

Castillo et al. (2011)

Early work on the credibility assessment of events on Twitter (Castillo et al., 2011) introduces content-, user-, topic-, and propagation-based features. Those features have been exploited and explored by various studies on online rumours. Rumours have different diffusion patterns from non-rumours.

Kwon et al. (2013)

Kwon et al. (2013) study temporal, structural, and linguistic differences between rumours and non-rumours. Firstly, temporal evolution patterns of rumours tend to have several and periodic peaks, while those of non-rumours typically have a single remarkable spike during their lifetime. As to structural features, the authors analyse diffusion networks in which *nodes* are users involved in rumour spreading and *edges* represent follower-followee relationships and reposting of rumours. Rumours exhibit sparser networks than non-rumours. Finally, linguistic features are used to compare users' reactions. Users are more likely to use negative (e.g. not, never), cognitive (e.g. cause, know), and tentative (e.g. perhaps, guess) expressions for rumours.

Kwon et al. (2017)

In more recent work, Kwon et al. (2017) further examine features related to user profiles, networks, and temporal patterns of rumour evolution. They identify differences between rumours and non-rumours. Their findings include that linguistic and user features are good signals for rumours in their early stages, while network and temporal features play a significant role in rumour detection over longer periods.

Liu et al. (2017)

Similarly, structural, temporal, user, and linguistic features of rumours and non-rumours on Sina Weibo, a Chinese microblogging site, have been studied (Liu et al., 2017). Their findings include that rumours are more likely to encourage people to repost them than non-rumours. Liu et al. (2015) present that features related to user profiles and reactions significantly contribute to the characterisation of rumours. Some studies (Ma et al., 2017; Kwon et al., 2013; Kwon et al., 2017) have found that rumours tend to spread from low-impact users to influencers, whereas non-rumours have the opposite tendency.

Liu et al. (2015)

Ma et al. (2017), Kwon et al. (2013), and Kwon et al. (2017)

2.2 DEFINITION OF RUMOUR DETECTION

As described in Section 2.1.2, message-level rumour detection is defined as the task of determining which social media posts report rumours and disseminate information that has not been verified at the time of posting (Zubiaga et al., 2018a). In other words, rumour detection is the task of identifying social media posts that satisfy a definition of rumour described in Section 2.1.1 and several characteristics detailed in Section 2.1.3. A message identified as a rumour may be proved or disproved later. The identification of the veracity of social media posts is beyond the scope of rumour detection. Formally, the task is formulated as a binary classification.

Definition 1 Rumour Detection An input set is denoted by

$$\mathcal{T} = \{(x_1, \mathbf{f_1}, t_1), (x_2, \mathbf{f_2}, t_2), \cdots, (x_N, \mathbf{f_N}, t_N)\},$$

where x_i denotes a source tweet, $\mathbf{f_i}$ is its features, and t_i is the posting time of the source tweet for $i \in [1, N]$. All source tweets are chronologically ordered. A classifier takes $\mathcal{T}$ as input, and assigns a binary label $y_i \in \{0, 1\}$ to each post x_i based on its textual content and $\mathbf{f_i}$. In general, y_i is 1 if x_i is a rumour, and 0 otherwise.

2.3 RELATED WORK

2.3.1 *Related Work On Data Augmentation*

Automatic data augmentation has been employed in a wide range of ML tasks as it helps to improve the generalisation performance of ML models, in particular, deep learning algorithms. Data augmentation usually makes use of transformations to which deep learning models invariant. For example, common transformations for images include flipping, rotating, scaling, cropping, and adding noise. This thesis focuses on the augmentation of textual data. The most common approach for the task is to replace words or phrases with synonyms (Zhang et al., 2015a; Kobayashi, 2018; Vosoughi et al., 2016; Vijayaraghavan et al., 2016; Kolomiyets et al., 2011).

In one work on text classification (Zhang et al., 2015a), a WordNet thesaurus (Miller, 1998), in which synonyms for a word or phrase are grouped and ordered by semantic relatedness, is used to replace words in training corpora including reviews, news articles, and DBpedia data sets. The number of words to be replaced (r) and the index of the synonym of a given word are randomly and respectively determined from a geometric distribution with parameter $p = 0.5$ in which $P(r) \sim p^r$. The authors present that augmented data improves the performance of Convolutional Neural Networks (CNNs) for

Zhang et al. (2015a)

text classification. In particular, character-level CNNs trained on augmented data achieves the best performance.

Previous research (Vosoughi et al., 2016; Vijayaraghavan et al., 2016) applies the method of Zhang et al. (2015a) to tweets and shows that data augmentation can bring performance gains in deep learning tasks on noisy and short social media texts.

Vosoughi et al. (2016)

Vosoughi et al. (2016) augment domain-independent English tweets to train an encoder-decoder embedding model built with character-level CNNs and LSTM. Stop words, user names, and hashtags are not replaced by the method. POS tags of replaced words should be consistent with those of words in the original texts. The number of tweets before data augmentation is not presented, but the author report that 3 million tweets in total are available after data augmentation.

Vijayaraghavan et al. (2016)

Another work (Vijayaraghavan et al., 2016) on tweet stance classification employs the same technique but uses Word2Vec (Mikolov et al., 2013) instead of the WordNet thesaurus (Miller, 1998). Synonyms of a given word are ranked based on cosine similarity between the Word2Vec vector of a given word and that of each synonym. Items with similarity less than a threshold are excluded. The reported number of augmented tweets is $500,000$.

Despite the wide use of synonyms in text data augmentation and their contribution to performance enhancement, the use of paradigmatic relations can provide a wider range of substitutes for a given word (Kobayashi, 2018).

Kobayashi (2018)

Kobayashi (2018) proposes methods for context-aware data augmentation based on a Bidirectional Language Model (BiLM). At each time step, a probability at a target word is computed forward and backward based on a probability distribution of its surrounding words. The outputs of forward and backward computation are concatenated and fed into a feed-forward NN which outputs words with a probability distribution. Given the output of BiLM, words for augmentation are sampled from an annealed distribution, $p_\tau(\cdot|S\{w_i\}) \propto p(\cdot|S\{w_i\})^{\frac{1}{\tau}}$. If $\tau \to \infty$, words are sampled from a uniform distribution. If $\tau \to 0$, words are always words with the highest probability. The authors also propose a variation of the proposed method by incorporating sentiment labels (i.e. positive and negative) of words. For example, given an input sentence "The actors are fantastic.", their context-aware data augmentation method often augments the sentence by assigning high probabilities to some negative words such as "bad" and "terrible" as a substitute for the term "fantastic". To prevent this issue, a label (y) indicating the sentiment of each word in the input sentence is incorporated into the annealed distribution, resulting in $p_\tau(\cdot|y, S\{w_i\}) \propto p(\cdot|y, S\{w_i\})^{\frac{1}{\tau}}$. Their method is evaluated for six different text classification tasks such as sentiment classification and opinion polarity detection with CNNs and Recurrent Neural Networks (RNNs). Contextual data augmentation with sentiment labels makes marginal improvements over performances of synonym-based methods by achieving accuracy of 78.20 on average.

Abulaish and Sah (2019)

Recently, a data augmentation method which combines $n-$grams and Latent Dirichlet Allocation (LDA; Blei et al. (2003)) has been proposed (Abulaish and Sah, 2019). Firstly, data preprocessing such as removing URLs and stop words and stemming is applied to a collection of reviews, and then processed reviews are classified into positive and negative reviews based on star rating. Next, LDA is used to extract and rank keywords from positive and negative review corpora separately. The top 500 keywords with the highest relevance score are obtained for each type of review. In data augmentation, each review is augmented by combining the original review with all of its trigrams that contain at least one keyword from the LDA review keywords of the same class type (i.e. positive or negative). The method is evaluated on its effectiveness in polarity classification (negative or positive) of reviews using CNNs. The results show that data augmentation can help reduce variations between training and validation accuracy and overfitting.

Whereas most work on text data augmentation generates variations of the original text based on the transformation of words and phrases, a recent study augments tweets by translating a tweet to a different language and then translating it back to the original language. Luque and Pérez (2018) exploits data augmentation to increase the size of training data for the sentiment analysis of a Spanish corpus. Specifically, each tweet written in Spanish is first translated into English, French, Portuguese and Arabic using Google Translate [3]. Converted tweets are translated back into Spanish.

Luque and Pérez (2018)

Unlike current artificial data augmentation methods based on modifications to existing data or reliance on limited knowledge bases, the method proposed in this thesis uses large-scale real-world social media data. It can not only increase the amount of training data but most importantly help to increase the quality and diversity of original data.

2.3.2 *Related Work on Potential Rumour Identification*

2.3.2.1 *Burst Detection Based on Temporal Signals*

Temporal patterns of information diffusion on social media have been widely studied. In particular, several studies have shown that temporal features play a key role in rumour detection (Kwon et al., 2013; Kwon et al., 2017; Liu et al., 2017; Ma et al., 2015). In this section, related studies, which aim to detect bursts in event evolution on social media exclusively based on temporal patterns, are introduced.

Hsieh et al. (2012)

Hsieh et al. (2012) propose a system that detects key moments when the number of tweets is above a threshold calculated using the mean (μ_t) and standard deviation (σ_t) of the number of tweets observed up to current time t. Given a time series which consists of time windows and the number of messages at each time window, a threshold at each time step is defined by $\alpha * (\mu_t + x * \sigma_t)$, where $0.7 \leq \alpha \leq 1.0$ and $1.5 \leq x \leq 2.0$. The authors do not describe how to decide α and x. The length of each time window is set to 30 seconds. The experiments are conducted on Twitter data sets related to sports games. Tweets were collected using related keywords. For example, “MUFootballClub” and “fcbarcelona” were used to collect tweets related to a UEFA Champions League match between FC Barcelona and Manchester United. For ground truth for evaluation, the authors obtain highlights (e.g. goals and home runs) from recorded live streaming videos. Timestamps of detected bursts are manually compared with those of highlights, and precision is reported. The results show that the performance of the proposed burst detection method greatly varies between domains of sports matches (i.e. football, basketball, baseball, and tennis). The highest precision is 0.83 for a football game and the lowest is 0.52 for a basketball game. Such results

[3] https://translate.google.com/

indicate that the performance of the method is highly dependent on temporal patterns of event evolution, which is not generalisable to new data.

Zubiaga et al. (2012)

Zubiaga et al. (2012) describe an outlier-based burst detection method that detects key bursts for scheduled events. The method starts to learn temporal patterns of event evolution 15 minutes before an event starts. Note that this setting is not ideal for the early identification of potential rumours. Given a time series in which the length of each time window is 60 seconds, a time window is labelled as a key burst if the tweeting rate (i.e. the number of tweets) at the window is above 90% of the previous rates. Although the authors claim that their method can detect consecutive bursts, Meladianos et al. (2015) show that it detects a large number of spurious outliers and misses bursts with relatively low tweet rates throughout the evolution of an event. Experiments are conduced on Twitter data sets related to 26 football games. Seven types of highlights (i.e. *goals, penalties, red cards, disallowed goals, game starts, game ends, and stops/resumptions*) of the game obtained from online articles are used as ground truth. The proposed method achieves F1-score of 0.63, precision of 0.51, and recall of 0.84.

Nichols et al. (2012)

Nichols et al. (2012) argue that burst detection should be based on *variations* in the number of posts of adjacent time windows rather than absolute volumes. According to the authors, detection methods based on absolute volumes identity all small fluctuations appearing during a long period of a high volume of posts, and might miss some time windows with relatively fewer posts. The proposed gradient-based approach first identifies key peaks if a slope (i.e. a difference in the number of tweets between the current time window and the previous one) is above a threshold. For each key peak, a time window where its slope starts to increase (i.e. "Start Time") and that where its slope start to increase again after a decrease from the peak (i.e. "End Time") are identified. Finally, each key burst is represented as a tuple ("Start Time", "Peak Time", "End Time"). Specifically, given a time series in which the length of each time window is 60 seconds, slopes for all time windows are computed. Given a set of slopes, their threshold is defined by $3 * (\texttt{median of slopes})$. For the selection of the threshold, the authors visualise time series for data sets for 36 sports games and manually inspected graphs. However, it should be noted that the same data sets are used to evaluate the method, which means their thresholds are optimised for the data sets used in the experiments. It is not guaranteed that the method with the proposed threshold will achieve reported performance or close to it on new data sets. In other words, the evaluation of its generalisability requires further research.

To evaluate the method, online articles about highlights of sports matches are used as ground truth. Categories of key moments identified by authors are as follows: *goals, penalties, red cards, yellow cards, disallowed goals, game start, game end,* and *half time*. The results of experiments conducted on three football matches show that the proposed method tends to achieve precision between 0.89 and 0.92 and recall between 0.62 and 0.91. The analysis of recall for different key moment categories shows that high recall is achieved for *goals, red cards, penalties, game ends,* and *half time*, while recall is low for the others. This is because the events with high recall have larger spikes. Remember that the motivation behind their gradient-based method is capturing key moments with fewer posts. However, their results demonstrate that their method still suffers from the issue most existing burst detection methods have.

Doman et al. (2014) propose a sub-event detection framework, Twitter Enthusiasm Degrees (TED), to generate highlights of sports games. Input to the TED is time series in which the length of each time window is set to 30 seconds. At each time window, a TED value defined by $E \times L \times (1 - R)$, where E is the number of exclamation marks in all tweets posted during the time window, L is the number of tweet including repeated expressions such as "GOOOAAALLL", and R is the number of retweets. E and L are measures of users' excitement. Specifically, high E and L indicate that users are actively involved in event evolution. On the other hand, high R indicates that an event is dying out because the authors assume that retweets represent delayed user reactions. Next, a time window is annotated as a highlight if its TED value is above a threshold defined by $\mu + \alpha \times \sigma$. μ and σ are the mean and the standard deviation of TED values for all time windows of a game. α is a parameter to be learnt. Highlight videos provided by news media are used as ground truth. The highlights detected by the method with $\alpha = 1$ and $\alpha = 2$ are manually and visually compared with the ground truth. While both values for α tend to detect the start and end of sports matches and goal events correctly with a few spurious bursts, they identify several spurious bursts (i.e. false positives) for other minor events such as "hit", in particular, $\alpha = 1$ detects more false positives than $\alpha = 2$ does.

Doman et al. (2014)

One recent work (Peng et al., 2018) on emerging product topic detection proposes a method to identify time windows where the topic popularity of products (e.g. films) emerges as part of a topic prediction framework. Given a set of reviews of a product (i.e. topic), the method generates a time series in which the length of each time window is one day. An emerging score of the topic t at each time window c is defined by

Peng et al. (2018)

$$ES(t,c) = \frac{tp(t,c) - EWMA(tp(t,1), tp(t,2), \cdots, tp(t,c-1))}{1 + EWMStd(tp(t,1), tp(t,2), \cdots, tp(t,c-1))},$$

where $tp(t,c)$ is the number of reviews of the product t at time c, $EWMA$ is the exponentially weighted moving average, and $EWMStd$ is the exponentially weighted moving standard deviation. If $ES(t,c)$ is above a threshold, the time window is annotated as a key burst. $EWMA$ requires a constant smoothing factor between 0 and 1 which controls the rate at which the influence of previous observations decay exponentially. However, the work does not explain how they control the parameter. As the method for burst detection is proposed for the task of predicting topic popularity, burst detection results are not evaluated.

This chapter highlights two issues that existing methods have. One is that most existing studies conducted experiments with sporting matches data sets. This is because ground truth such as highlights of a match provided by media outlets makes it easier to evaluate proposed methods. However, source code and data sets are not publicly available. Some source code is reproduced for the experiments of Chapter 5 and a novel approach for evaluating burst detection in the context of rumour studies is proposed. The other is that the thresholds proposed by most existing studies look arbitrary and were fine-tuned for specific tasks on specific corpora and domains. It is unlikely that they can generalise well to other data sets. Moreover, not many existing studies explain how their thresholds and parameters for burst detection affect results. This makes it difficult to adapt their methods to new events. This thesis contributes to burst detection by addressing these limitations.

2.3.2.2 *Event Summarisation*

Existing summarisation methods can be classified into two groups: *generative* and *extractive* summarisation. The former aims to produce a new text which summarises an input corpus. The latter aims to select some sentences (i.e. tweets) which are representative of an input corpus.

Erkan and Radev (2004)

Erkan and Radev (2004) propose *LexRank*, a graph-based extractive summarisation method which exploits cosine similarity between sentence pairs. Given a set of sentences, the method builds a similarity graph in which nodes are sentences and edges are similarity relationships between sentences. Next, the method builds a similarity matrix where each entry is the cosine similarity between the TF-IDF representations of two sentences. TF-IDF is used to compute the similarity. By incorporating a similarity threshold, the similarity matrix is converted into a binary matrix. Specifically, if the similarity of a sentence pair is above a threshold, the corresponding entry in the matrix is replaced with 1. Entries below the threshold are set to 0. The method then computes the degree of each sentence in the graph. Each entry in the similarity matrix is divided by the degree of the sentence of the corresponding row in the matrix. The similarity matrix allows the LexRank to measure the importance of sentences based on its relative importance to its neighbours. Finally, the algorithm outputs LexRank scores for a given set of sentences. No social media data set is used in their experiments, and ROUGE-N (Lin, 2004) is used for evaluation. It is a metric that evaluates automatic summarisation based on overlapping occurrences of n-grams between summaries of a proposed method and references.

Nenkova and Vanderwende (2005) propose a frequency-based method called *SumBasic* for extractive summarisation. Given an input corpus which consists of multiple sentences, it computes probability distributions $p(w_i)$ over all words w_i in the input. $p(w_i)$ is defined as the number of times a word w_i appears in the input divided by the total number of words in the input. The weight of each input sentence is the average of its constituent word's probabilities. The sentence with the highest score and containing the word with the highest probability is included in the output summary. For each word in the selected sentence, its probability is updated with a new probability defined by $p(w_i) * p(w_i)$, which is an estimate of the probability that the word w_i will appear in the output summary twice. This procedure is repeated until the desired length of the summary is reached. The experiments are conducted on long documents built using online articles rather than a social media corpus. Summaries obtained using the SumBasic and baselines including the LexRank are compared with manual summaries based on ROUGE-N. The experimental results show that the SumBasic outperforms the LexRank in terms of ROUGE-1.

Sharifi et al. (2013)

Sharifi et al. (2013) propose two generative summarisation methods in two different settings. The first setting is to generate a single summary for an input corpus. One method is a graph-based algorithm called *Phrase*

Reinforcement generates summaries by searching for the most frequent phrases in an input corpus. Given a starting phrase (e.g. a trending topic) and a set of related posts, phrases appearing before and after the starting phrase (i.e. root node) form sub-graphs on the left-hand and right-hand side of the root node. After iterating all posts in the input corpus, each node is weighted according to its frequency and the distance from the root node. Finally, the algorithm searches for paths with the highest weights in the graph and generates a summary. The other method is *Hybrid TF-IDF* which is based on the Term Frequency Inverse Document Frequency (TF-IDF) that deals with the sensitivity of the standard TF-IDF to document lengths. Since tweets are short, considering each tweet or a set of tweets at each time step as a single document can be problematic when applying standard TF-IDF. Therefore, the authors propose a hybrid version which considers a set of tweets as a single document when computing TF and each tweet as a single document when computing IDF. For evaluation, human annotators generate manual summaries (i.e. references). F-measure, precision, and recall based on ROUGE-N (Lin, 2004) are employed to evaluate results. The experimental results for generating a single summary for an input corpus show that Hybrid TF-IDF outperforms Phrase Reinforcement by showing performance close to manual summaries.

In the second setting, the Hybrid TF-IDF is extended to generate multiple summaries for an input corpus based on a threshold. Specifically, given input sentences weighted by the Hybrid TF-IDF developed for the first setting, the method computes cosine similarity between a candidate sentence and a sentence which is already chosen as a final summary. If the similarity is above a threshold ranging from 0 to 0.99, the candidate tweet is incorporated into the final set of summaries. Comprehensive experiments comparing eight different summarisation algorithms are conducted. The results show that a SOTA method called SumBasic (Nenkova and Vanderwende, 2005) outperforms other baselines and the Hybrid TF-IDF in terms of F-measure (0.2544) and recall (03274), while the Hybrid TF-IDF achieves the highest precision of 0.2499.

Nichols et al. (2012)

Nichols et al. (2012) propose an extractive algorithm based on phrase graphs and weighting schemes proposed by Sharifi et al. (2013). The method uses phrase graphs to score input sentences. As described above, a phrase graph is constructed using a set of posts at each time step. Next, the score of each post is defined as the sum of weights of nodes appearing in the post. Finally, the method outputs the top N posts with the highest score as a summary of each time window. Experiments are conducted on data related sports games. Online articles and manual summaries are used as references. Results are evaluated in two ways. Firstly, evaluation results based on ROUGE-N show that the proposed method outperforms Hybrid TF-IDF (Sharifi et al., 2013). Secondly, humans manually evaluate the proposed method and Hybrid-TF-IDF in terms of readability, grammaticality, and content. The results show that the proposed method provides more understandable and informative summaries that the SOTA baseline does.

Meladianos et al. (2015)

Meladianos et al. (2015) propose an extractive summarisation method based on graphs of words and K-cores. A graph-of-words refers to the graphical representations of input texts (see Section 5.3.3.2 for details). K-cores of a graph G refer to the maximal subgraph where the degrees of all vertices is at least K (Seidman, 1983a). K-cores with the largest core number represent the most cohesive subregions of a graph. Therefore, the use of K-cores for ranking keywords allows identifying *influential keywords* from a collection of noisy tweets (Tixier et al., 2016). At each time step, given a set of posts, an undirected weighted graph, in which each node is a word appearing in the input and each edge represents the co-occurrence of two nodes, is generated. Next, K-cores are computed using the graph and core numbers are used as term weights. The score of each post is defined as the sum of weights of nodes appearing in the post. Finally, a post with the highest score is chosen as a summary of the time window. Experiments are conducted on Twitter data sets related to football matches. Online articles are used to generate ground truth. The results show that the proposed method outperforms baselines by achieving micro-average F1-score of 0.68 and macro-average F1-score of 0.72 on average.

Meladianos et al. (2018a)

In more recent work, Meladianos et al. (2018a) extract representative tweets by optimising a non-decreasing sub-modular function. Specifically, given a graph-of-words at each time window, a function f is defined as the sum of the weights of all edges connecting all pairs of words appearing in an input set of posts. A summary of each time window is obtained by maximising this function subject to a cardinality constraint. Experiments are conducted on data sets related to 20 sports matches. Following (Nichols et al., 2012), 8 key sub-events (e.g. goals and red cards) are considered. Ground truth for the 8 categories is collected using articles provided by FIFA.com. The work presents a few examples of summaries generated by the proposed method, and manually compares them with the ground truth. Evaluation is not comprehensive because only one of 20 football matches was used to show that the method can produce informative summaries.

Liu et al. (2012)

Liu et al. (2012) propose a graph-based summarisation algorithm that incorporates three different features to assign a weight to each post in an input corpus. A function for scoring each post in an input corpus considers the number of reposts of a post, the number of followers of the post's author, and readability. To consider user diversity in the final summary, the method limits the number of posts from the same user in the final summary. The authors report that there is no publicly available data set for tweet summarisation, and hence they generate small and manually annotated data sets. The data sets are not publicly available. ROUGE-N is used for evaluation. The results show that the proposed method outperforms LexRank (Erkan and Radev, 2004) by achieving ROUGE-1 of 0.4562 and ROUGE-2 of 0.3692.

Alsaedi et al. (2016a)

Alsaedi et al. (2016a) propose three summarisation methods: *Temporal TF-IDF*, *Retweet Voting*, and *Temporal Centroid Representation*. They are based on existing methods but consider temporal dynamics of event evolution. Firstly, *Temporal TF-IDF* considers a set of tweets as a document. At each time step, it computes term weights in a document by considering word distributions in a collection of documents at prior time windows. The work does not explain how to score each tweet using term weights computed via Temporal TF-IDF. Secondly, at each time step, *Retweet Voting* method computes a difference in *Retweet Score* between the current time window and the previous one, and then selects tweets with high values as a summary of the time window. *Retweet Score* is defined as the ratio of the number of retweets of each tweet to the total number of retweets of all posts in a input set of tweets. Finally, at each time step, *Temporal Centroid Representation* computes cosine similarity between the TF-IDF representation of a tweet and that of the centroid of each cluster (i.e. a set of tweets). ROUGE-1 is used to compare the proposed methods with several SOTA baselines. The overall results show that the *Temporal TF-IDF* outperforms baselines for English, Arabic, and Japanese tweets. The authors also manually evaluate the three methods proposed in their work based on quality, relevance, and usefulness. The results show that the *Temporal TF-IDF* produces high-quality and useful summaries, while the *Temporal Centroid* produces the most relevant summaries.

Chakrabarti and Punera (2011)

Chakrabarti and Punera (2011) propose *SummHMM* which extracts key tweets from a collection of tweets posted during time windows segmented using modified Hidden Markov Models (HMMs). The authors argue that each segment of a timeline of an event should contain only one sub-event. Input to the model is a set of tweets. The model first learns three types of symbol distributions and transition probabilities that best fit the input. The three symbols include different types of sub-events, event-specific terms such as proper nouns, and noisy and irrelevant terms. Next, the optimal segmentation of an event is obtained. At each segment (i.e. time window), each tweet is represented as a vector of TF-IDF weights of its constituent words. The weight of each tweet is defined as the sum of the cosine similarity between the tweet and every tweet in the input corpus. The top N tweets with the highest score are selected as a summary of the time window. Experiments are conducted on Twitter data sets related to football matches. The data sets are manually labelled specifically for football matches. Precision and recall are employed to evaluate summarisation performance. The work does not compare the proposed method with SOTA methods, but propose two baselines. The overall results show that the SummHMM outperforms the baselines.

Chapter 5 proposes graph-based methods based on three different term weighting schemes and compares them with SOTA graph- and frequency-based methods. While existing summarisation methods for social media posts were evaluated over small data due to the difficulty of generating ground truth, the experiments of Chapter 5 exploit larger data created with weak supervision in Chapter 4. As for evaluation metrics, the ROUGE-N (Lin, 2004) is the most common method for evaluating summarisation performance. Chapter 5 proposes a novel domain-specific evaluation approach for the identification of potential rumours on Twitter. Specifically, it evaluates whether extracted summary tweets are qualified for potential rumours using large-scale ground truth with weak labels. Rumour source tweets in the augmented data (see Section 4.6.2) are used as the ground truth.

2.3.3 *Related Work On Rumour Detection*

There are two strands of rumour detection approaches according to machine learning techniques. **Traditional rumour detection methods** represent social media posts as a set of features that are useful for distinguishing rumours from non-rumours. Hand-crafted features such as content-, user-, and network-based features have been extensively studied. Recently, modern **representation learning techniques** such as deep learning architectures have become increasingly popular by providing significant improvements to SOTA results with little or no feature engineering. This section introduces and discusses different approaches to rumour detection. Although a large majority of related studies have used Twitter, it is expected that most features can easily be transferred to other platforms such as Facebook. In this description of the SOTA, this chapter expects readers to be familiar with the basics of ML techniques such as supervised, weakly supervised, and unsupervised learning including RNNs and CNNs. For an exhaustive introduction, see (Shalev-Shwartz and Ben-David, 2014; Hastie et al., 2005).

2.3.3.1 *Supervised Learning Approaches*

Supervised learning-based methods are conventional approaches and aim to distinguish rumours from non-rumours based on manually curated features. The most widely used approach is to use hand-crafted features such as content-, user-, and network-based features (Qazvinian et al., 2011; Kwon et al., 2017; Kwon et al., 2013; Yang et al., 2012; Sun et al., 2013; Zhao et al., 2015; Zhang et al., 2015c; Wu et al., 2015; Ma et al., 2015; Liu et al., 2016; Zubiaga et al., 2017; Hamidian and Diab, 2015; Hamidian and Diab, 2016).

A general description of hand-crafted features and examples

Content-based features are related to characteristics of texts. Table 2.1 shows examples of such features which are used in feature-based methods for rumour detection. *User-based features* consider characteristics of users' profile and behaviour. The former refers to properties of users and the latter refers to information that can be extracted from user's activities on social media. Table 2.2 shows examples of such features which are used in feature-based methods for rumour detection. "Account age" refers to the time interval between the posting time of each message and the creation of its author's account. "User originality" is defined as the ratio of the number of original tweets a user has posted to the number of posts the user has retweeted (Vosoughi, 2015). "User credibility" indicates whether a user is verified or not. "User controversiality" is associated with the sentiment of replies. *Network-based features* consider propagation patterns of information and are extracted from networks constructed based on user activities. Two most popular networks are friendship and diffusion networks (Kwon et al., 2017). In the case of Twitter, a friendship network is generated using follower-followee relationships and a temporal diffusion network (or tree) is constructed using information flow among users of a friendship network. While network-based features can represent dynamic propagation patterns of rumours, extracting them requires complex feature engineering (Hamidian and Diab, 2015). Note that most network-based features provide aggregate-level information rather than information specific to a single tweet and user. Table 2.3 shows examples of network-based features which are used in feature-based methods for rumour detection. Note that features listed in Table 2.1, 2.2, and 2.3 are some general examples.

Table 2.1: Examples of content-based features employed in supervised learning-based rumour detection methods

Description
Length of the message (i.e. the number of words in the message)
Opinion words (i.e. positive and negative words)
Emoticons
URLs
Twitter-specific characters (i.e. #, @, an RT)
Punctuations (e.g. question and exclamation marks)
Vulgar words
The use of capitalised words
Abbreviations
Multimedia (e.g. photos and videos)
Originality (i.e. whether the message is original or a repost)
Part-of-speech tags (e.g. adjectives and interjections)

Table 2.2: Examples of user-based features employed in supervised learning-based rumour detection methods

Description
Account age
User reputation (e.g. the number of followers, the ratio of followers and followees)
User credibility (i.e. verified or not)
User gender
User type (e.g. organisation, individual, and celebrity)
User originality
User controversiality
Number of messages the user has liked so far

Table 2.3: Examples of network-based features employed in supervised learning-based rumour detection methods

Description
Depth of the tweet (i.e. the longest path from the original tweet to the target tweet in a diffusion tree)
Fraction of new users
Fraction of original messages
Fraction of messages containing URLs
Fraction of isolated nodes

See Section 2.1.2 for preselected rumours.

Some research on rumour detection aims at identifying preselected rumours (Qazvinian et al., 2011; Hamidian and Diab, 2015; Hamidian and Diab, 2016; Kwon et al., 2013; Kwon et al., 2017; Ma et al., 2015).

Qazvinian et al. (2011)

Qazvinian et al. (2011) formulate the task as a retrieval task. First of all, tweets matching manually defined Regular Expression (REGEX) patterns of preselected rumours are collected. REGEX is a set of characters representing patterns for matching text. For example, a query "`Obama & (muslim|islam)`" is used to collect tweets related to a rumour "Is Barack Obama muslim?". The retrieved tweets are manually categorised into rumours and non-rumours (see Section 2.4.4 for details of their data).

They propose a ranking model which consists of Bayesian classifiers built on 9 features categorised into three different types of features of training data. Content-based features include lexical and part-of-speech patterns for unigrams and bigrams. Network-based features represent whether a user is one who posted a tweet or a retweet. Twitter-specific features include hashtags and URLs. Each Bayesian classifier for each feature computes the log-likelihood ratio which is the likelihood that a given classification result would be expected in a rumour-related tweet to the likelihood that the same result would be expected in non-rumour tweet. For each query (e.g. "Is Barack Obama muslim?"), the model is expected to retrieve relevant tweets based on the proposed features.

As the data contains five different queries (see Section 2.4.4), 5-fold CV is performed for evaluation. F1-score, precision, recall, and accuracy are used to evaluate the performance of the proposed ranking model. The authors perform an ablation study which aims to analyse how a specific feature affects the performance of a model by removing some features. The experimental results show that the model based on all the 9 features achieves the highest precision (0.944) and that based on content-based features achieves the highest F1-score (0.932) and accuracy (0.941).

Hamidian and Diab (2015)

Hamidian and Diab (2015) address the task of multi-label rumour classification. The authors use the data and all the features proposed by Qazvinian et al. (2011). Tweets related to five different rumours are classified into 6 different types: *Non-rumour, Endorses rumour, Denies rumour, Questions rumour, Neutral, and Undetermined*.

This work proposes new features on top of Qazvinian's features. Firstly, *pragmatic* features are proposed to capture contexts expressed in tweets. Their pragmatic features include sentiment (i.e. 6 levels indicating positivity and negativity), named entities, event types, and emoticons. The feature "event type" is obtained based on entities and event phrases. For example, an entity "iPhone" and event phrase "announcement" result in an event type "Product Launch" (Ritter et al., 2012). Next, "Time" and "User ID" features are added to the Twitter-specific and network-based features proposed by Qazvinian et al. (2011). In specific, the "Time" feature indicates whether a tweet is posted during a "Busy Day" or "Regular Day". The "User ID" feature indicates whether a user retweeted or replied to a rumour in the past are proposed. However, specific definitions of "Busy Day" and "Regular Day" are not given in their work. Table 2.4 shows the full list of the proposed features.

The proposed two-step model based on the C4.5 decision tree algorithm (Quinlan, 1993) first classifies whether an input tweet is "Non-rumour", "Undetermined", and "Related to rumour". In the second step, it breaks down tweets labelled as "Related to rumour" into four classes based on stances (i.e. endorsement, denial, question, and neutral). The experimental results show that their model achieves a lower F1-score (0.83) than the ranking model by (Qazvinian et al., 2011) with and without the newly proposed features. Note that this work performs multi-class rumour classification. The results of the ablation analysis show that the best classification performance in terms of precision and F-measure is achieved by using content-based features. Twitter-specific and network features have a limited impact on rumour detection.

Table 2.4: Features used in (Hamidian and Diab, 2015).

Category	Feature description	Type
Twitter-specific and network-based	Time	Binary
	Hashtag	Binary
	Hashtag content	String
	URL	Binary
	Retweet	Binary
	Reply	Binary
	User ID	Binary
Content	Content unigrams	String
	Content bigrams	String
	POS unigrams	String
	POS bigrams	String
Pragmatic	Named entities	String
	Event type	String
	Sentiment	String
	Emoticon	Categorical

Hamidian and Diab (2016)

In more recent work, Hamidian and Diab (2016) study the task of rumour detection on Qazvinian's data. The main contribution of this work is a new feature called Tweet Latent Vector (TLV) which represents a tweet as a 100-dimensional vector. Each input tweet is preprocessed. Preprocessing includes tokenisation, stemming, and removing infrequent words. The importance of words in a preprocessed tweet is computed via TF-IDF. Finally, Semantic Textual Similarity (STS) model (Guo and Diab, 2012) built on various corpora including Wiktionary, WordNet, OntoNotes, and Brown corpus is applied to extract TLV.

In their experiments, a Support Vector Machine (SVM) Tree Kernel classifier (Moschitti, 2004) is employed to classify input tweets into rumours and non-rumours. The authors compare the effectiveness of three different feature sets: 1) features proposed in (Qazvinian et al., 2011), 2) those in (Hamidian and Diab, 2015), and 3) content unigrams+TLV. The experimental results show that the content unigrams+TLV features outperform the benchmark features, achieving precision of 0.972 and recall of 0.99. This supports the findings of benchmark studies that content-based features are most effective in achieving the best performance on rumour detection.

The three studies (Qazvinian et al., 2011; Hamidian and Diab, 2015; Hamidian and Diab, 2016) report promising rumour detection results in

Limitations of Qazvinian et al. (2011), Hamidian and Diab (2015), and Hamidian and Diab (2016).

terms of precision, recall, and F1-score. However, they have several limitations to show that their methods can generalise and transfer to large-scale, real-world settings and new domains. The most fundamental limitation is that the data used in their experiments is small and contains very specific rumour cases. Therefore, it is not very likely that their models can perform well for a wide range of real-world rumours which exhibit different propagation and linguistic patterns. Another limitation is that their features do not consider the temporal evolution of rumours. This is not realistic as features which play a key role in identifying rumours can vary according to stages of the life cycle of rumours (Nguyen et al., 2017).

Kwon et al. (2013)

Kwon et al. (2013) addresses event-level rumour detection based on temporal, structural, and linguistic differences between the evolution of rumours and that of non-rumours. A description of the features used in this work is shown in Table 2.5. This section now details the three types of features. Firstly, temporal evolution patterns of rumours tend to have several and periodic peaks, while those of non-rumours typically have a single remarkable spike during their lifetime. Based on this finding, the authors propose a new time series fitting model called *Periodic External Shocks (PES)* which is able to capture periodic bursts exhibited during the diffusion of rumours and non-rumours. PES is an extension of SpikeM (Matsubara et al., 2012) which aims to model temporal diffusion patterns of information on social media.

Before showing how the PES works, explaining the SpikeM is necessary. The *SpikeM* models how information becomes popular and diminishes over time on social media. It captures the power-law decay and periodicity of real-world data and prevents rise and fall patterns from diverging to infinity. It observes changes in the number of users who post messages related to an event over time. Its parameters are explained as follows:

- n_d and n denote the total time duration of time series data and a timestamp, respectively (i.e. $n = 0, 1, ..., n_d$). The time when posts about an event are published for the first time is denoted by n_b.
- No users post until time n_b, but S_b users immediately post about the event at time n_b. The external shock, denoted by $S(n)$, can be given by

$$S(n) = \begin{cases} 0 & (n \neq n_b) \\ S_b & (n = n_b) \end{cases}$$

- While $B(n)$ denotes the cumulative number of users who have posted about the event until time n, $\Delta B(n)$ denotes the number of users who have just found out a rumour at time n. The cumulative number of users who have not posted the event until time n is denoted by $U(n)$. Moreover, the model assumes that there are a finite number of users N.

$$\Delta B(n) = \sum_{t=0}^{n} \Delta B(t), \quad B(n) + U(n) = N, \quad B(0) = 0, \quad U(0) = N$$

- The periodicity function $p(\cdot)$ is incorporated into the model to capture periodic patterns of the user posting behaviour and is defined by

  $$p(n) = 1 - \frac{1}{2} P_a sin\left(\frac{2\pi}{P_P}(n + P_s) + 1\right)$$

  where P_a, P_p, and P_s denote the strength of periodicity, period, and translation along the x-axis, respectively.

- Finally, SpikeM model can be constructed. The power-law decay term is added to the model to explain an assumption: the influence of a user on the future states of information diffusion decreases over time. Let β and ϵ denote the strength of the transmission of an event between individuals and the noise component, respectively. The SpikeM is given by

  $$\Delta B(n+1) = p(n+1) \cdot (U(n) \cdot \sum_{t=n_b}^{n} (\Delta B(t) + S(t)) \cdot \beta \cdot (n+1-t)^{-1.5} + \epsilon)$$

  Considering the external shock, the summation part of SpikeM is the number of users who post about an event by being influenced by S_b users.

Although the SpikeM can describe the characteristics of information diffusion better than existing models can do, the authors report two limitations of the SpikeM. One is that the periodic function $p(n)$ of the SpikeM cannot distinguish rumours from non-rumours: the time series of rumours generally have several and periodic spikes, whilst those of non-rumours have a single striking spike. The other is that the external shock $S(n)$ might be repeated throughout the life cycle of a rumour. To apply the SpikeM to rumour detection by overcoming its limitations, the authors add a periodic external shock function, denoted by $q(t)$, to the initial external shock function $S(n)$. The PES is defined by

$$\Delta B(n+1) = p(n+1) \cdot \left[\frac{\beta}{N} \cdot U(n) \cdot \sum_{t=n_b}^{n} (\Delta B(t) + \bar{S}(t)) \cdot (n+1-t)^{-1.5} + \epsilon\right]$$

where $q(t)$ has parameters q_p, q_a, and q_s that denote the period, amplitude, and the shift of the periodic external shock, respectively. Other parameters of PES are the same as those of the SpikeM.

$$\bar{S}(t) = S(t) + q(t)$$
$$q(t) = q_a \left[1 + (\sin\left(\frac{2\pi}{q_p}(t + q_s)\right)\right]$$

As to structural features, the authors build friendship and diffusion networks. In both networks, nodes are users involved in rumour spreading. Edges in a friendship network represent follower-followee relationships and those in a diffusion network represent reposting of rumours. Some structural features are extracted from the largest connected subgraph (LCS) of a friendship network. The clustering coefficient of a vertex in a network in Table 2.5 is a measure of the degree to which nodes in the network tend to cluster together. Rumours exhibit sparser networks than non-rumours do. Finally, linguistic features are used to compare users' reactions to rumours with those to non-rumours. Users are more likely to use negative (e.g. not, never), cognitive (e.g. cause, know), and tentative (e.g. perhaps, guess) expressions for rumours.

The effectiveness of the proposed features was evaluated using three supervised classifiers: Decision Trees, Random Forests, and SVMs. The experiment results show that Random Forests achieve the best performance with their features and that temporal features related to these bursts contribute the most to improving rumour detection performance. Specifically, it is observed that the periodicity of external shock q_p achieves the highest predictive power in rumour classification among the top significant temporal, structural, and linguistic features. The fraction of information flow from low-impact users to high-impact users in a diffusion network also shows high predictive power. Overall, the classification results show that their features outperform baseline features (Castillo et al., 2011) in identifying rumours by achieving F1-score 0.893, precision 0.900, and recall 0.892. The baseline features employed in this work are shown in Table 2.6.

Table 2.5: Features used in (Kwon et al., 2013).

Category	**Feature description**
Temporal	Periodicity of external shock (q_p) External shock periodicity offset (q_s) Interaction periodicity offset (p_s)
Structural	Average clustering coefficients of the friendship network Density of the largest connected subgraph (LCS) Average clustering coefficients of the LCS Fraction of isolated nodes Fraction of low-to-high diffusion
Linguistic	Positivity (love, nice, sweet) Negations (no, not, never) Social processes (mate, talk, they, child) Cognitive mechanisms (cause, know, ought) Exclusion (but, without, exclude) Insights (think, know, consider) Tentativenss (maybe, perhaps, guess) See (view, saw, seen) Hear (listen, hearing)

Table 2.6: Some of features used for determining the credibility of tweets (Castillo et al., 2011). Only 15 features used as baseline features in (Kwon et al., 2013) are described.

Feature description
Fraction of tweets containing a URL
Fraction of tweets containing negative sentiment
Fraction of tweets containing positive sentiment
Fraction of tweets containing a question mark
Fraction of tweets containing a mention
Fraction of tweets containing a smiley emotion
Fraction of tweets containing the first person pronoun
Spreader's average number of posts
Spreader's average number of friends
Spreader's average number of followers
Spreader's average number of days since registration
Average sentiment score in tweets
Number of distinct short URLs in tweets
Maximum level of the diffusion tree
Fraction of tweets by the most prolific spreader

Ma et al. (2015)

Ma et al. (2015) point out that the model proposed by Kwon et al. (2013) has a limited number of parameters to capture complex temporal variations of social context features. To overcome this limitation, they propose a new model called *Dynamic Series-Time Structure (DSTS)* which fits time series of various social context features. The input of the DSTS model is a set of posts related to an event (i.e. rumour) denoted by E. A vector representation of E obtained using the DSTS is denoted by $V(E)$ and defined by

$$\begin{aligned} V(E) &= (\mathbf{F}_0, \mathbf{F}_1, \cdots, \mathbf{F}_N; \mathbf{S}_0, \mathbf{S}_1, \cdots, \mathbf{S}_N) \\ \mathbf{F}_t &= (\widetilde{f}_{t,1}, \widetilde{f}_{t,2}, \cdots, \widetilde{f}_{t,D}) \\ \mathbf{S}_t &= \frac{\mathbf{F}_{t+1} - \mathbf{F}_t}{Interval(E)}, \end{aligned}$$

where $\mathbf{F}_t \in \mathbb{R}^d$ is a vector of normalised features, i.e. $f_{t,k}$ $(k = \{1, 2, \cdots, D\})$ (see Table 2.7), extracted from messages that have been posted up to time t. $Interval(E)$ is the length of each time interval in hours. Social context features employed in this work include content-, user-, and propagation-based features (see Table 2.7). While most of them were selected from existing work, some content-based features such as topic distributions obtained using Latent Dirichlet Allocation (LDA; Blei et al. (2003)) are new. The experimental results show that SVMs with the proposed features outperform several baselines in terms of recall (0.909), F1-score (0.894), and accuracy (0.896).

Kwon et al. (2017)

Kwon et al. (2017) extend their earlier work (Kwon et al., 2013) by incorporating additional features including user-based features. Their user features are associated with probability distributions of the number of followers, friends, and tweets. Table 2.8 lists them. *Kurtosis* is a measure of the degree to which input data is heavy- or light-tailed compared to the normal distribution. *Skewness* is a measure of the asymmetry of a probability

Table 2.7: Features used in (Ma et al., 2015).

Category	Feature description
Content	LDA-based topic distributions of messages
	Average length of messages
	Number of positive (negative) words in messages
	Average sentiment score of messages
	Fraction of messages with a URL
	Fraction of messages with smiling (frowning) emoticons
	Fraction of messages with the first person pronouns
	Fraction of messages with hashtags
	Fraction of messages with mentions (@)
	Fraction of messages with a question mark
	Fraction of messages with an exclamation mark
	Fraction of messages with multiple question or exclamation marks
User	Fraction of users that have personal description
	Fraction of users that have a profile picture
	Fraction of verified users
	Fraction of verified users of each type (e.g. celebrities)
	Fraction of male (female) users
	Fraction of users located in large (small) cities
	Average number of friends (i.e. followee) of users
	Average number of followers of users
	Average number of posts of users
	Average account age of users in days
	Average reputation score of users (i.e. (#followers/#followees)
Propagation	Average number of reposts
	Average number of comments (i.e. replies)
	Number of messages

distribution. The authors study the impact of different types of features on rumour detection over time windows with varying lengths. The experimental results show that linguistic and user features are good signals for ERD. The combination of both features shows a solid performance regardless of time window lengths, achieving up to F1-score of 0.84, precision of 0.86, and recall of 0.84. On the other hand, network and temporal features play a significant role in rumour detection over longer time periods. In particular, temporal features achive F1-score of 0.88, precision of 0.87, and recall of 0.89 for the 56−day window. These results are almost equivalent to the results achieved by a model exploiting all types of features.

Table 2.8: User features used in (Kwon et al., 2017).

Feature description
Kurtoisis of the number of followers, followers, friends, and tweets
Skewness of the number of followers, followers, friends, and tweets
Minimum of the number of followers, followers, friends, and tweets
25% quantile of the number of followers, followers, friends, and tweets
Median of the number of followers, followers, friends, and tweets
75% of the number of followers, followers, friends, and tweets
Maximum of the number of followers, followers, friends, and tweets
Average of the number of followers, followers, friends, and tweets
Standard deviation of the number of followers, followers, friends, and tweets

Research exploiting replies and/or reposts for rumour detection.

Using each individual social media post in isolation as a unit of analysis has limited potential to advance SOTA performances on rumour detection. Recently, there have been attempts to exploit *context* for rumour detection.

Ma et al. (2017)

Ma et al. (2017) propose a novel method based on propagation tree kernels (PTKs). In Natural Language Processing (NLP), tree kernels are used to compare sentences represented by syntactic tree structures (i.e. parse trees; Collins and Duffy (2002)). In the proposed architecture, propagation trees of rumour and non-rumour source tweets are built. Specifically, the root of a tree is a source tweet and nodes of subtrees are its contexts (replies). Given a propagation tree of a source tweet, the proposed model classifies the source tweet into a rumour or non-rumour.

Given two propagation trees $T_1 = \langle V_1, E_1 \rangle$ and $T_2 = \langle V_2, E_2 \rangle$, the proposed PTK denoted by $K_P(T_1, T_2)$ is defined by

$$\sum_{v_i \in V_1} \Lambda(v_i, v_i') + \sum_{v_j \in V_2} \Lambda(v_j', v_j),$$

where $\Lambda(v, v')$ measures the similarity between two subtrees having v and v' as roots. For each node $v_i \in V_1$, v_i' is defined as a node of V_2 which is the most similar to v_i. $v_j' \in V_1$ is obtained for $v_j \in V_2$ in the same way. A similarity function between two different nodes from two different trees comprises two components measuring user-based and content-based similarities. Therefore, the PTK can capture user- and content-related as well as temporal dynamics of rumour spreading. The authors also propose context-sensitive PTK (cPTK), which considers how information has propagated from a source post to the root of the current subtree. In other words, cPTK incorporates functions which measure the similarity between ancestor nodes of the two target nodes ($v_i \in V_1, v_i' \in V_2$) to compute the similarity between two trees T_1 and T_2.

These proposed tree kernel functions (i.e. PTK and cPTK) are applied to a kernel-based SVM. The experiments are conducted over "Twitter 15/16 data" in which tweets are classified into four classes: *Non-rumour, False rumour, True rumour, and Unverified Rumour*. The description and statistics of the data used in this work are described in Section 2.4.2. Several baselines including supervised and deep learning models are compared with their models. The results show that the cPTK and PTK outperform baselines for all types of rumours in terms of F1-measure. The cPTK outperforms PTK all but the *Non-rumour* class. In particular, the cPTK achieves F1-score of 0.709 for the *False-rumour* class. As for early detection, the cPTK also outperforms baselines, achieving accuracy of 0.75 when considering propagation paths within the first 24 hours after the posting time of a source tweet.

Nguyen et al. (2017)

Recent work (Nguyen et al., 2017) exploits a comprehensive set of features for rumour detection. Their rumour detection setting is similar to (Ma et al., 2015). Specifically, a model predicts whether an event is related to a rumour or news given a set of posts about the event. In their architecture, tweets are encoded via a hybrid of CNNs and RNNs fined-tuned on a domain-specific Twitter corpus associated with rumours and news. Fine-tuning embedding models with a task-specific corpus provides performance gain in several NLP tasks (Kim, 2014). The model is also used to output probabilities that individual tweets related to an event are related to news.

On top of content, user, Twitter-specific, and temporal features proposed in existing studies, they propose *Ensemble* and *Epidemiological* features. Ensemble features include *CreditScore* and *CrowdWisdom*. The *CreditScore* at a time interval is defined as the average of the prediction probabilities of all tweets in the interval. CrowdWisdom is the fraction of tweets containing *Debunking words* such as "hoax", "rumour", and "not true". Epidemiological features includes parameters of two epidemic models (i.e. SIS and SEIZ) as Kwon et al. (2013) and Kwon et al. (2017) use some parameters of time series fitting models SpikeM and PES. Epidemic models are originally used to describe the epidemic dynamics of the diffusion of infectious diseases. Some research exploits such models to model information diffusion on social media (Woo et al., 2011; Xiong et al., 2012). All features are represented as a single vector using DSTS model (Ma et al., 2015).

Random Forests are employed as a classifier. The experimental results show that their architecture outperforms baselines, achieving accuracy of 0.82 in the first one hour and accuracy of 0.91 in the first 48 hours. An analysis of feature importance shows that the feature *CreditScore* is the most effective feature for rumour detection. The feature *CrowdWisdon* shows good performance after 32 hours since the start of an event, achieving accuracy of 0.76 on its own.

Research on message-level rumour detection.

While the studies introduced above deal with the detection of preselected rumours, others study breaking events which potentially produce several rumours. Specifically, the latter aims to identify newly emerging rumours from a collection of tweets related to events as early as possible.

Zubiaga et al. (2017)

For instance, Zubiaga et al. (2017) aim to identify rumours which circulated during five real-world breaking news events (i.e. *PHEME* data) via a sequential model called Conditional Random Fields (CRFs). While most existing studies attempted to learn temporal propagation patterns of rumours on social media by incorporating temporal and network-based features, this work leverages the power of a sequential model to learn the contextual information of individual tweets. It should be noted that the term "context" used in this work is slightly different from that used in this thesis. Specifically, Zubiaga's *contexts* refer to source tweets that have been posted up to the posting time of a tweet to be classified. In this thesis, *contexts* refer to conversational threads (i.e. replies) of source tweets. The authors use conversational threads to annotate tweets, but do not use them in their model.

For each event, the input of the CRF is a graph in which nodes are tweets and edges indicate temporal relations between nodes. Two nodes are linked if one node is a tweet posted before the other tweet in an event. Given a sequence of graphs for all source tweets, the CRF outputs that of binary labels (i.e. rumour and non-rumour). The main advantage of using the CRF is that it considers labels of each node's neighbours (i.e. contextual information). The model is based on content-based features extracted from tweet texts and social features obtained using the metadata of users. Their features are adopted from existing studies.

In the experiments, the proposed model is compared with several non-sequential models. The results demonstrate that their sequential model outperforms non-sequential classifiers in terms of precision (0.667) and F1-score (0.607) with all features. Similar to findings of several existing studies, content-based features are effective in boosting performance. Content-based features alone can achieve similar results obtained by using the combination of linguistic and social features. In contrast, the performance (F1-measure) decreases by 27% with social features alone.

Tolosi et al. (2016)

Tolosi et al. (2016) study rumour detection exclusively based on user-, content-, and URL-based features which are considered to be independent of events. Their hypothesis is that event-specific features, features related to retweets and replies, and propagation-based features are not available in the early stages of rumour diffusion. To avoid confusion over event-dependent and event-independent features, the features used in this work are presented in Table 2.9. Experiments are conducted using a classification tree with the features and the model is evaluated via Leave-one-out cross-validation (LOOCV) in which one event is used as a test set and the remaining events are used as a training set on each iteration. The model achieves F1-score of 0.65. The results show that even the proposed features exhibit domain-specific characteristics. For example, an analysis of URLs cited in tweets shows that credible sources are more likely to appear in non-rumours. A probable reason for this observation can be that people tend to include credible sources in tweets to debunk or validate rumours rather than generate rumour-mongering tweets.

Table 2.9: Event-independent features used in (Tolosi et al., 2016).

Feature description
Probability that the user posts a rumour based on historical posting activities
Number of followers
Number of followees
Number of tweets the user has posted so far
Probability that the tweet is related to a rumour based on URLs
Tweet length
Ratio of occurrences of topic-independent capitalised terms* in rumours to those in non-rumours
Presence of capitalisation
Presence of punctuations

* BREAKING, JUST, MORE, PHOTO, VIDEO, NEWS, UPDATE, DEVELOPING, LIVE, WATCH, NOW, DETAILS, LATEST, OMG, UPDATED, STORY

2.3.3.2 *Weakly-Supervised Learning Approaches*

Supervised learning-based methods rely heavily on hand-crafted features which may require domain knowledge that is not always available as well as labour-intensive and time-consuming feature engineering (Lai et al., 2015; Severyn and Moschitti, 2015a; Wang et al., 2012). To overcome these limitations, some work on rumour detection has used phrasal patterns (e.g. “Is this true?”, “Really?”, “It is not true.” etc.) of rumours (Resnick et al., 2014; Zhao et al., 2015) and semantic relatedness between references and candidates for rumours (Jin et al., 2017a) as weak supervision.

Resnick et al. (2014)

Resnick et al. (2014) demonstrate three challenges in analysing rumours on social media: 1) ERD, 2) difficulties in achieving both high precision and recall when retrieving posts related to identified rumours, and 3) understanding audiences of rumours for practical applications such as journalism. The authors then address each challenge by proposing a comprehensive framework, called *RumourLens*.

Its first component for detecting rumours identifies clusters of candidate rumours by searching for tweets which contain expressions appearing in controversial claims such as “Is it true?” The authors report that the proposed approach achieves higher recall than methods based on trending topics and hashtags. The second component, called ReQuery-ReClassify (ReQ-ReC), collects posts related to a particular rumour. The successful retrieval of related posts aims to achieve both high precision and high recall. High precision is achieved when retrieved results mostly consist of relevant instances. High recall is achieved when most of the relevant instances are retrieved. However, there exists a trade-off between precision and recall. The proposed retrieval method aims to achieve high precision. Given retrieved tweets, ReQ-ReC categorises them into “spreading”, “correcting”, and “unrelated” The final component analyses the users who participate in the diffusion of identified rumours. As rumour detection is not the main focus of this work, no experiment regarding the task is performed.

Zhao et al. (2015)

Zhao et al. (2015) discover that certain phrases expressing enquiries for verification and corrections such as “Rumour”, “Unconfirmed”, “Is it true?”, and "Really?" appear in the early stages of the life cycle of a rumour. Similar to the approach proposed by Resnick et al. (2014), tweets which contain such expressions (i.e. signal tweets) are clustered, and then a summary tweet is extracted from each cluster. Subsequently, tweets which are related to summary tweets but do not contain sceptical expressions are identified and merged into corresponding clusters. Statistical features of clusters are identified (see Table 2.10). Finally, classifiers such as SVMs and Decision Trees rank clusters based on the likelihood of containing rumours.

For evaluation, the fraction of rumours among the top N clusters (i.e. precision) is computed. The results show that the proposed model with Decision Trees shows the best performance when $N = 10$, achieving precision of 0.9. Performance decreases as N increases. According to recent studies (Zubiaga et al., 2017; Jin et al., 2017a) which employ Zhao's method as a baseline, it achieves very low recall (0.065 and 0.008) on the *PHEME (6392078)* data (see Section 2.4.1) and a Twitter corpus about the 2016 U.S. presidential election, respectively. These results show that Zhao's method cannot generalise to new data.

Table 2.10: Statistical features used to rank clusters (Zhao et al., 2015).

Feature description
Ratio of signal tweets to all tweets in the cluster
Ratio of the entropy of the word frequency distribution of signal tweets to that of all tweets in the cluster
Average number of words in each signal tweet in the cluster
Average number of words in every tweet in the cluster
Percentage of retweets among signal tweets in the cluster
Percentage of retweets among all tweets in the cluster
Average number of URLs in each signal tweet in the cluster
Average number of URLs in every tweet in the cluster
Average number of Hashtags in each signal tweet in the cluster
Average number of Hashtags in every tweet in the cluster
Average number of Mentions (@) in each signal tweet in the cluster
Average number of Mentions (@) in every tweet in the cluster

Jin et al. (2017a)

Jin et al. (2017a) formulate the problem of rumour detection as the task of text matching to minimise the use of manual labour for data annotations. Articles about rumours verified by a fact-checking website called Snope.com are used to generate references. References and tweets are represented by real-valued vectors using embedding models and cosine similarity of a pair of reference and tweet vectors is computed. In the experiments, several embedding algorithms (i.e. TF-IDF, BM25, Word2Vec, and Doc2Vec) are tested. BM25 computes a relevance score between a document (i.e. tweet) and a particular query (i.e. reference) based on term frequency and document length. The results show that BM25 achieves the best F1-score (0.820).

2.3.3.3 *Unsupervised and Reinforcement Learning Approaches*

Deep learning techniques have been increasingly popular within the research community of rumour detection by providing significant improvements to SOTA results with little or no feature engineering (Chen et al., 2018; Ma et al., 2016; Ma et al., 2018b; Ruchansky et al., 2017; Jin et al., 2017b; Yu et al., 2017; Nguyen et al., 2017; Kochkina et al., 2018a; Liu and Wu, 2018; Ma et al., 2018a). Recently, attention mechanisms, which allow DNNs to learn relationships between different positions in the sequence by jointly attending to different representation segments at different positions, have shown new SOTA performance in a wide range of ML tasks such as sequence labelling and transduction. They are an extension of encoder-decoder models. They allow an encoder to build contexts from parts of an input sequence without encoding the entire input (Bahdanau et al., 2015). They also enable DNNs to selectively focus on the most important and useful segments of the sequence and to effectively learn long-range dependencies (Vaswani et al., 2017). This chapter introduces some research which leverages attention for rumour detection. Deep learning-based rumour detection architectures usually employ one of CNNs and RNNs or a hybrid of CNNs and RNNs to learn characteristics of rumours and rumour diffusion. Little work exploits reinforcement learning for rumour detection. This chapter introduces one study (Zhou et al., 2019a) which leverages it to determine when to perform rumour classification.

A rumour detection architecture based on CNNs.

CNNs are able to learn not only lexical, syntactic, and semantic characteristics of input sentences (Severyn and Moschitti, 2015b), but also high-level interactions between linguistic features of the input (Yu et al., 2017).

Yu et al. (2017)

Yu et al. (2017) propose a CNN-based misinformation detection architecture called *Convolutional Approach for Misinformation Identification (CAMI)*. The input of CAMI is a set of tweets related to an event. Firstly, tweets are chronologically ordered and divided into groups of equal size. Next, each group of tweets at a time window is represented using *Paragraph vector* (Le and Mikolov, 2014) which is an unsupervised algorithm that learns vector representations of texts of different lengths. Given a sequence of paragraphs, every paragraph of N words is mapped to a vector represented by a column in matrix $\mathbf{D}$ and every word is mapped to a vector represented by a column in matrix $\mathbf{W}$, and then the paragraph and word vectors are concatenated or averaged out. All concatenated or averaged paragraph vectors are concatenated to form paragraph representations for the input sequence. These low-level lexical features are mapped to high-level semantic features via CNNs. Finally, output (i.e. the probability that the input event is associated with misinformation) is obtained via fully connected layers and Softmax layer. The results show that the CAMI outperforms baselines by achieving F1-score of 0.793, precision of 0.744, and recall of 0.848 on a publicly available Twitter data (Ma et al. (2016); see Section 2.4.3).

Rumour detection architectures based on RNNs.

Despite the effectiveness of CNNs in capturing discriminative features of input sentences and embedding the input into low-dimensional vectors, CNNs are not able to preserve dynamic temporal aspects of rumour diffusion which can be captured by RNNs.

Ma et al. (2016) propose various models based on RNNs: 1) basic RNNs, 2) single-layer LSTM and Gated Recurrent Units (GRUs), and 3) multi-layer GRUs. Given a set of events, each of which consists of relevant posts, the proposed task is to classify each event into a rumour or non-rumour. For each event, a time series is generated using posting times of relevant posts. A set of tweets at each time interval is represented as TF-IDF values of all terms in the set. For experiments, a Twitter corpus comprising around 1 million tweets associated with 498 rumour and 494 non-rumour events and a Sina Weibo corpus comprising around 4 million posts related to 2, 313 rumour and 2, 351 non-rumour events are constructed. More details about the data are given in Section 2.4.3. Overall, the results show that the proposed models, particularly the 2-layer GRUs, outperform the SOTA rumour detection models based on hand-crafted features. The 2-layer GRU achieves F1-score of 0.898 and that of 0.914 for the Twitter and Weibo data sets, respectively. *Ma et al. (2016)*

Ma et al. (2018b) propose *bottom-up* and *top-down* Recursive Neural Networks (RvNN) models for rumour detection. The motivation behind this work is that a reply is usually directed to the closest ancestor message rather than the root of the propagation path. The authors claim that recursive networks can model such structures and capture indicative and discriminative signals for rumours. RvNN is a type of NNs which has tree structures. In NLP, for example, the input of RvNN is a sentence. Words in the sentence are leaf nodes in a parse tree. RvNN recursively represents a parent node as a function of its children nodes for all nodes. The learnt hidden states of nodes are used for various NLP tasks. *Ma et al. (2018b)*

The two proposed methods are based on tree structures of rumour diffusion (i.e. relations between source tweets and their replies). Given a source tweet and its replies, a tree is constructed for each source tweet. In the bottom-up model, responsive nodes point to nodes they are replying to. Each input node (i.e. tweet) is represented as a vector of TF-IDF values of words in the tweet. When recursion reaches the root node (i.e. source tweet), its state is used to predict its label. In the top-down model, paths are generated from a source tweet to replies. The representation of each node is computed by combining itself and its parent node. The output of recursion is representations of several leaf nodes. Note that the actual classification is performed in batches, which means the actual input is a sequence of several source tweets and their replies and the output should be a sequence of labels. As different source tweets have different numbers of leaf nodes, the representations of leaf nodes are first fed into a max pooling layer to get a fixed-size vector. Softmax function is applied to predict the label of the root node.

Experiments are conducted over the *Twitter 15/16* data in which tweets are classified into four classes: *Non-rumour, False rumour, True rumour, and Unverified Rumour* (see Section 2.4). The model is trained using squared errors. Comprehensive evaluation results show that the top-down model advances several SOTA baselines for all classes but *Non-rumour*, achieving up to F1-score of 0.835 and accuracy of 0.737.

Ruchansky et al. (2017) propose a model called CSI which consists of three different modules. The first module aims to model temporal propagation patterns of related posts of an event. To this end, LSTMs learn temporal representations of user engagement and textual representations of related *Ruchansky et al. (2017)*

posts. At each time step, a feature vector of the input article is fed to a LSTM cell. Features for representing the input include the number of related posts the input received, time intervals between posts, a vector of features of the input's source user, and embeddings of each post. The last hidden state of LSTM networks is fed into a fully connected layer with a hyperbolic tangent (tanh) activation function. The output vector is concatenated with the output of the second module, and the concatenated vector is passed to the last module.

The second module aims to model user features. To extract them, a weighted graph, in which nodes are users and an edge indicates the engagement of two nodes, is first constructed. Edges are weighted based on the number of posts which both users engaged with. Given a graph, an adjacency matrix is generated and a low-dimensional representation for each user is obtained. This representation is fed into a fully connected layer with a tanh activation function followed by another fully connected layer with a sigmoid activation function.

The final module integrates the outputs of the first two modules to predict a label for each input post. Specifically, masking is applied to the user features from the second module to only consider features of users engaged with the input post. The average of the selected user vectors is concatenated with the output vector of the first module. The concatenated vector is fed into the last fully connected layer with a sigmoid activation function.

Overall, the results show that the CSI outperforms baselines by achieving F1-score of 0.894 and accuracy of 0.892 on a publicly available Twitter data set (Ma et al., 2016). The authors conduct an ablation study by testing the model with content features only (i.e. CI) and that with content and temporal features (i.e. CI-t). S is omitted from the model names as user information is not incorporated in the models. CI and CI-t achieve F1-scores of 0.846 and 0.848, respectively. This indicates that the proposed temporal features are not effective signals for rumours.

Rumour detection architectures based on hybrids of CNNs and RNNs.

Some work has proposed hybrids of CNNs and RNNs in the hope that they can model not only higher-level textual and social representations of rumours but also temporal dynamics of rumour spreading.

Liu and Wu (2018)

Liu and Wu (2018) propose a hybrid of CNNs and RNNs which is capable of learning rumour propagation based on features of users who have participated in rumour spreading. Input to the proposed architecture is a sequence of embeddings of users who participated in the diffusion of a source tweet (i.e. a propagation path). The task is to produce a label for each source tweet given its propagation path. Each user embedding consists of eight features representing the characteristics of users (see Table 2.11). They are not novel and can simply be extracted from the metadata of Twitter objects. A sequence of user embeddings is fed into GRU (Cho et al., 2014), a variant of RNNs. Output vectors (i.e. hidden states) are aggregated into a single vector by mean pooling.

In parallel, the same input sequence is fed into CNNs to learn local variations of user characteristics. A 1-dimensional convolution with a filter of height h and a ReLU activation function is applied to h consecutive user vectors, where h is smaller than the length of the input sequence. This results in a scalar feature for the subsequence. The same convolution is repeated for different subsequence of length h. Consequently, a sequence of length (n-h+1) vectors (i.e. local variations of user features) is obtained. The final vector is obtained by applying mean pooling. In the next step, the outputs obtained via RNNs and CNNs are concatenated into a single vector. This vector is fed into a multi-layer feed-forward NN and Softmax layer. The final output is a probability distribution over classes for the input source tweet.

In the experiments, propagation paths available in the first 24 hours after the start of an event are considered to evaluate models under the setting of ERD. The results show that the proposed hybrid model with user embeddings outperforms SOTA baselines. Specifically, it achieves accuracy of 0.842 and F1-score of 0.875 for the class *False rumour* in *Twitter 15* data. As for *Twitter 16* data, it achieves accuracy of 0.863 and F1-score of 0.898. The findings of this study supports the findings of recent work (Kwon et al., 2017). Kwon reported that linguistic and user-based features are effective in identifying rumours over short and long time periods. In particular, they are useful than temporal and structural features for ERD because the latter are not usually available in the early stages of rumour spreading. Liu and Wu (2018) claim that content-based features are less visible than user features at the very early stages.

Table 2.11: User features used for representing a user embedding (Liu and Wu, 2018).

Feature description	**Type**
Length of user description	Integer
Length of username	Integer
Number of followers	Integer
Number of followees (i.e. friends)	Integer
Number of statuses (i.e. number of posts issued by the user)	Integer
Account age	Integer
Is verified	Binary
IS geo enabled	Binary

Research on multi-task learning of rumours.

Recently, rumour detection has been studied as part of multi-task learning that aim to perform several rumour-related tasks (refer to Section 1.1 for details) such as stance classification and rumour verification at the same time. A motivation behind this is that representations useful for one task (e.g. rumour detection) can help to improve the performance of another task (e.g. stance classification) due to correlations between them (Ma et al., 2018a).

Kochkina et al. (2018a)

Kochkina et al. (2018a) propose a context-aware LSTM-based architecture that jointly performs rumour detection, stance classification, and rumour verification. Conversational threads (i.e. replies) of source tweets are leveraged as contexts. For a source tweet, its conversational threads are decomposed into several branches according to Twitter mentions (i.e. @username). Each source tweet and replies in each decomposed conversational branch are preprocessed (i.e. removing nonalphabetic characters, lowercasing, and tokenisation). An input sequence consists word2vec embeddings of the preprocessed source tweet and replies. The word vectors were pre-trained on Google News data set [4].

The proposed multi-task learning architecture consists of a shared LSTM layer followed by task-specific layers. First of all, the task of stance classification requires labels for every reply in a conversational thread. Therefore, the hidden state for every time step of the shared LSTM is fed into multiple dense ReLU layers followed by a Softmax layer. For the other two tasks, the hidden state at the last time step in the shared LSTM is fed into another LSTM layer followed by multiple dense ReLU layers and a Softmax layer. Experiments are conducted by performing LOOCV on the *PHEME (6392078)* data (Zubiaga et al. (2016a); see Section 2.4.1). For evaluation, macro-average F-score and accuracy are employed. The results show that the model which jointly learns the three tasks outperforms baselines for rumour verification and models that learn two tasks (i.e. Verification+Stance and Verification+Detection), achieving macro-average F-measure of 0.396 and accuracy of 0.492 for five breaking news events. This indicates that the joint learning of different rumour-related tasks boosts the performance of a single task.

Ma et al. (2018a) Ma et al. (2018a) propose two GRU-based architectures with *shared layers* and/or *task-specific layers* for jointly learning rumour detection and stance classification tasks. One is the uniform shared-layer architecture (MT-US) and the other is the enhanced shared-layer architecture (MT-ES). Shared layers are used to learn common features of two different tasks and task-specific layers are used to learn task-specific representations of input sequences. Unlike (Kochkina et al., 2018a) in which different tasks share the same input, each task receives a sequence of posts as input. Each post is encoded as a vector of TF-IDF values of words in the post.

In the MT-US, a task-specific sequence of post embeddings is input to a GRU shared layer in which weight matrices for input vectors and hidden states are shared between different tasks. Input sequences are mapped into low-dimensional vectors via the shared layer and they are fed into a Softmax layer that predicts class probabilities. Output vectors for stance classification and rumour detection are different. For the former, the Softmax layer outputs class predictions for every reply. For the latter, it outputs a single vector of probabilities over different classes for the source tweet. The MT-ES is proposed to overcome a major limitation of the MT-US. As weight matrices are shared between two different tasks, it is assumed that highly weighted representations are always important for both tasks. However, it is likely that some features should get more attention in one task than the other.

In the MT-ES, task-specific GRU layers are incorporated. Specifically, a task-specific input sequence is fed into shared and task-specific GRU layers. The hidden state of task-specific layers at each time step is computed based on the current input, the previous hidden state, and the hidden state from the shared layer. Output vectors of task-specific layers are fed into a Softmax layer. Subsequent procedures are the same as the MT-ES.

The experimental results show that the MT-ES outperforms baselines in terms of macro-average F1 by achieving 0.464. The evaluation results per class show that baselines and the proposed architectures perform poorly for “True rumour”. In contrast, “Non rumour” class is the easiest class to predict.

[4] https://code.google.com/archive/p/word2vec/

Rumour detection architectures exploiting attention mechanisms.

The following parts of this section introduce recent research which leverages attention mechanisms for rumour detection. They have shown to achieve new SOTA results in several NLP tasks such as machine translations. Several studies on rumour detection have exploited attention mechanisms in various ways. A literature review on such studies will provide a better insight about online rumours and their evolution. In general, attention mechanisms are applied on top of RNNs. For instance, given the final hidden states of RNNs, a *score function* is employed to score each element (e.g. word) in an input sequence (e.g. a sentence). Scores for each element are normalised via a Softmax function, and these normalised scores are called *attention weights*. Finally, the attention-weighted representation of the input is obtained as a weighted sum of the final hidden states. Equations for attention mechanisms are explained in Section 2.6.3.

Jin et al. (2017b)

Jin et al. (2017b) propose a novel architecture called *att-RNN*. It consists of LSTM with attention mechanisms and CNNs and detects rumours based on multi-modal features.

While most previous work exploits textual and social features, this work incorporates image features included in social media posts. LSTM networks with attention mechanisms are employed to learn textual and social representations of posts. Pre-trained deep CNNs generate vector representations of images. The input of the architecture is a tweet represented as a set of three instances: textual content, social context, and visual content.

As for textual content representation, each tweet is encoded as a word embedding. Social contexts are represented as a vector of features obtained from metadata provided by social media platforms and some semantic features such as polarity. The social context vector is fed into a fully connected layer so that it can have the same dimension as the text embedding. Textual content and social context representations are concatenated and the merged vector is passed to LSTM. The hidden states for individual words are averaged out, resulting in a single vector representing jointly learnt textual and contextual characteristics of the input tweet. This vector will be concatenated with a vector representing images.

Each image in the input tweet is represented as a $512-$dimensional vector via VGG-19 network (Simonyan and Zisserman, 2014). This vector is further fine-tuned with an auxiliary data set via two fully connected layers. Images inserted in a tweet are correlated with its text and social contexts. To incorporate this insight, attention mechanisms are employed. Specifically, the hidden states of LSTM networks are fed to fully connected layers with a ReLU activation function followed by another layer with a Softmax function. This results in a 512-dimensional vector of attention weights. The fine-tuned visual representation vector is weighted using the attention vector, which enables a model to pay more attention to certain features for rumour detection. Finally, the joint representation for text and social context and the attention-weighted image vector are concatenated into a single vector which represents multimodal features of the input tweet. The aggregated vector is fed into a Softmax layer which outputs a label (i.e. rumour or non-rumour) for the input tweet.

In experiments, several baselines from a logistic classifier to models for visual question answering and image captioning are employed. As for data sets, a publicly available Twitter data set designed for image verification (Boididou et al., 2014) and a Weibo data set generated by the authors are used. The results show that the utilisation of three different types of features is highly effective in rumour detection. The att-RNN outperforms several baselines by achieving F1-scores of 0.689 and 0.676 for rumour and non-rumour classes, respectively, on the Twitter data. The results of an ablation study show that the use of visual features (i.e. embedded images) improves F1-score by around 6%.

Chen et al. (2018)

Chen et al. (2018) also propose a framework (*CallAtRumour*) based on RNNs with attention mechanisms for ERD. Unlike (Jin et al., 2017b), this work applies attention mechanisms to sentence embeddings. The task is to determine whether an event is related to a rumour or non-rumour given a collection of related posts.

Input is a set of posts related to an event. The first step is to produce a TF-IDF dictionary for the most frequent K words in the input corpus. This ensures that every post embedding has the same length. Each post is represented as a vector of TF-IDF values of words in it. After encoding all posts, they are divided into a number of time intervals. At each time step, a sequence of post embeddings is fed to stacked LSTM.

Next, attention mechanisms are applied to the output (i.e. the hidden states of the last layer) of the LSTM and produce a vector of attention weights for each of the K words. In other words, an attention vector represents the relative importance of each word for rumour detection. The input sequence at the next time step is represented as a weighted sum of word embeddings at that time step. This procedure is recursively performed. At the last time step, the hidden state of the last layer is fed to a sigmoid layer which determines the input event is a rumour or not. The framework is trained using cross-entropy.

The experimental results show that the proposed framework outperforms several baselines by noticeable margins ranging from 5% to 20% (F1-score). The proposed framework achieves F1-score of 0.8715, precision of 0.8863, and recall of 0.8571 on a publicly available Twitter data set (Ma et al., 2016). Another experiment on ERD is performed by incrementally increasing training data size in chronological order. The results show that attention mechanisms are more effective in the early stages of rumour diffusion and that the proposed framework can achieve around 70% (precision and recall) within 20 hours since the start of an event.

Guo et al. (2018)

Guo et al. (2018) propose a rumour detection architecture based on hierarchical bidirectional LSTM (biLSTM) with attention mechanisms. The task is to determine whether an event is related to a rumour or non-rumour given a collection of related posts. As in (Chen et al., 2018), all posts are encoded into word embeddings and divided into time intervals. In this work, however, 32−dimensional word embeddings pre-trained on data sets used in the experiments are used rather than TF-IDF. Note that data used for pre-training word vectors is split into training (80%) and test (20%) sets in their experiments. This means that information about test data is used in training, which is not suitable for real-world scenarios. On top of content representations, the proposed architecture exploits hand-crafted social contexts. Table 2.12 lists features used in the method. It should be noted that the features categorised into "Post Texts" are not social features although the authors claim that their hand-crafted features provide social representations of input posts.

The framework consists of three main parts: 1) word-level layers, 2) post-level layers, and 3) sub-event-level layers. Each part consists of biLSTM with attention. A structural difference is that social features are incorporated into attention mechanisms in post-level and sub-event-level layers. For word-level learning, input embeddings are fed into biLSTM. Next, all hidden states of the last forward and backward layers are concatenated into a single vector. This is recursively applied to all words in each post. Attention mechanisms are applied to the final hidden state for each word. Given an attention vector and the hidden states for all words in the input corpus, each post vector, i.e. the j^{th} post in i^{th} sub-event, is represented as a weighted sum of the hidden states. In post-level learning, post vectors are fed into biSLTM to obtain representations of sub-events. For computing attention weights, the final hidden states of biLSTM and social features are used. Finally, a sub-event vector is defined as a weighted sum of the hidden states for all post vectors. Sub-event learning is exactly the same as the post-level learning. The final output is a high-level representation of the input event. It is fed into a fully connected layer followed by a Softmax layer which outputs a label (i.e. rumour or non-rumour) for the input event. Model is trained using cross-entropy.

Experiments are conduced on Ma data (Ma et al. (2016); see Section 2.4.3). The results for Twitter data show that the proposed model outperforms several baselines in terms of F1-score, precision, and accuracy for the rumour class. Specifically, it achieves F1-score of 0.948, precision of 0.730, recall of 0.825, and accuracy of 0.844. A SOTA baseline, CallAtRumour (Chen et al., 2018), achieves the highest recall (0.780). According to the results of an ablation study, attention mechanisms are helpful for rumour detection. The proposed hand-crafted features are generally useful and "Post Texts" features are the most effective. The authors conduct a similar experiment to (Chen et al., 2018) to evaluate their framework in the setting of early detection. They do not presents results in terms of the time periods needed to achieve certain performance (e.g. F1-score of 0.9). The results show that the proposed framework performs marginally better than baselines when 200 posts are used for training. Margins become wider as the number of posts increases.

Veyseh et al. (2019a)

Veyseh et al. (2019a) propose a context-aware framework based on self-attention mechanisms. It consists of two components. The first component classifies conversation threads (i.e. a sequence of a source post and its replies) into rumours and non-rumours based on representations obtained via self-attention. The other predicts latent labels of each post in the input sequence in order to emphasise the importance of the source tweet in the final representation of the input sequence. Remember that attention models are an extension of an encoder-decoder model. Self-attention, also known as intra-attention, is an attention mechanism relating different locations of a single sequence to compute its representation (Vaswani et al., 2017). In the self-attention model, the encoded representation of the input is considered as a set of pairs of a *key* and *value*. Specifically, keys and values are hidden states of an encoder. The output obtained using an decoder is referred to as a *query*. In an decoder, the output is obtained by mapping a *query* from the previous time step and the set of keys and values at the current time step.

The input of the framework is a sequence of a source post and its replies. Each tweet is encoded into a 300−dimensional vector using GloVe (Pennington et al., 2014) and max pooling is applied to post embeddings. To learn latent semantic relations between a source post and its replies, the pairwise similarity between two tweets computed via self-attention mechanism is

Table 2.12: Features used in (Guo et al., 2018).

Category	Feature Description
User Profile	Fraction of users with a description
	Fraction of users with an avatar
	Fraction of verified users
	Average reputation score
	Average number of followers
	Average number of friends
	Average time length of registration
	Average number of post per user
Propagation	Fraction of reposted tweets
	Average number of replies
	Average number of reposts
Post Texts	Total number of posts
	Average length of posts
	Average sentiment score
	Fraction of posts containing ? (!, ?!, @, URL, #)
	Fraction of posts with positive (negative) sentiment

exploited. Firstly, key and query vectors for each tweet embedding are computed and the pairwise similarity between two posts' key and query vectors is recursively computed over all posts in the input sequence. These similarity scores are used as attention weights. Finally, each tweet is represented as a weighted sum of hidden states. Given attention-weighted representations of all posts, the representation of the input sequence (i.e. the source tweet and its replies) is obtained via max pooling. This representation is fed into a $2-$layer feed-forward NN with a Softmax function to obtain a probability distribution over classes for the input sequence. The component is trained using the negative log-likelihood function.

The second component compares a label obtained solely based on the source tweet and that produced based on the attention-weighted representation of the entire input sequence. Specifically, each representation is fed into a separate feed-forward NN with a Softmax to produce a probability distribution over classes. This component is also trained using the negative log-likelihood function. The entire framework is trained using the final loss defined as a weighted sum of losses of the two components. Experiments are conducted on the Twitter 15/16 data (Ma et al., 2017). The results show that the proposed framework outperforms SOTA baselines for all but the "True rumour" class in terms of F1-score. As for the "False rumour" class, it achieves F1-scores of 0.764 and 0.751 for the Twitter 15 and Twitter 16, respectively. The best F1-score for the "True rumour" class is achieved by the top-down RvNN (Ma et al. (2018b); 0.821 for Twitter 15 and 0.835 for Twitter 16). The results of an ablation study show that self-attention greatly improves model performance. For instance, F1-score increases by 13.5% with self-attention for the "False rumour" class. Although the impact of the second component is smaller than that of self-attention, it helps to boost model performance.

Table 2.13: Tweet content features used in (Li et al., 2019).

Feature Description
Number of negation words*
Number of swear words*
Presence of a period (a question mark, exclamation mark)
Ratio of capital letters
Presence of URL
Presence of images
Word2Vec cosine similarity between the tweet and source tweet
Word2Vec cosine similarity between the tweet and thread
Word count
Character count
Whether the tweet is the source tweet of the conversation

* not, no, nobody, nothing, none, never, neither, nor, nowhere, hardly, scarcely, barely, don't, isn't, wasn't, shouldn't, wouldn't, couldn't, doesn't

** http://aurbano.eu/blog/2008/04/04/bad-words-list/

Li et al. (2019)

Li et al. (2019) propose a multi-task learning architecture for rumour detection and stance classification. It consists of one shared layer and two task-specific layers. The input of the framework is a sequence of a source tweet and its replies. Conversational threads are decomposed as described in (Kochkina et al., 2018a). Each tweet is represented as a vector using word2vec embeddings (Mikolov et al., 2013).

The shared LSTM layer is used to learn common features of two different tasks. The output of this layer is fed into LSTM networks in each task-specific layer. In the stance classification layer, the tweet content embedding is concatenated with a vector of hand-crafted features. Features used in this module are listed in Table 2.13. The aggregated vector is fed into standard LSTM which consists of a fully connected layer followed by a Softmax layer which produces a stance label for the input sequence. The hidden states of LSTM are used in attention mechanisms in the rumour detection module. In the rumour detection layer, the tweet content embedding is concatenated with a user embedding which represents the credibility of users engaged in the conversation. User features used for embedding are listed in Table 2.14. The merged representation is fed into LSTM. At each time step, the hidden state of LSTM is concatenated with the hidden state obtained from the stance classification module.

These concatenated vectors are used to compute an attention vector and the final representation of the input sequence is defined as a weighted sum of the concatenated hidden states. The attention-weighted representation is fed into a fully connected layer followed by a Softmax layer which outputs a label (i.e. rumour or non-rumour) for the input thread. The experiments are conducted on two publicly available data sets: RumourEval (Derczynski et al., 2017) and PHEME (Zubiaga et al., 2016a). The PHEME data used in this work is an early version of the *PHEME (6392078)* described in Section 2.4.1 and contains tweets for five events. 5-fold LOOCV is performed. The

LOOCV: one event is used as a test set and the remaining events are used as a training set on each iteration

Table 2.14: User features used in (Li et al., 2019).

Feature Description
Is trusted/satirical news account
Has trusted/satirical news URLs
Whether the user profile has a URL from the top domains
Client application name
Whether the user profile has a person name
Whether the user profile has location
Whether the user profile includes profession information

results show that the proposed framework outperforms baselines by achieving macro-average F1 scores of 0.418 and 0.606 on the PHEME and RumourEval, respectively.

Geng et al. (2019)

Geng et al. (2019) propose a GRU-based multi-view model which consists of content view, reply view, and sentiment view modules. In other words, the architecture jointly learns source post contents, context contents, and context stances for rumour detection. Here, context refers to replies of each source tweet. Note that most existing deep learning-based architectures based on multi-modal features aggregated different types of representations before feeding them into the last layer (e.g. a Softmax layer) for classification. In the proposed framework, each module performs classification and the final decision is made via majority voting.

Given a source post and its replies, the framework determines whether the source post is a rumour or non-rumour. In the content view module, the source post representation is obtained via an embedding layer initialised with pre-trained word embeddings. It is fed into bidirectional GRU networks, in which each GRU unit is assigned to the embedding of each source post representation. Self-attention mechanisms are applied to the hidden states of the last GRU layer. The final representation of the source tweet is a weighted sum of the hidden states. It is fed into a Softmax layer which outputs a label (i.e. rumour or non-rumour) for the source tweet.

In the reply view module, the representation of each reply is obtained via an embedding layer. Next, at each time step, each reply representation is fed into GRU. In this module, regular GRU networks rather than bidirectional ones are employed. All hidden states of the last layer are averaged out and fed into a Softmax layer to obtain a label.

In the sentiment view module, the representation of each reply's sentiment is obtained using a BERT (Devlin et al., 2018) encoder fined-tuned on a corpus built for sentiment analysis. The hidden states of the last layer of the fine-tuned BERT are used as sentiment embeddings of replies. They are fed into regular GRU networks. The following steps are the same as the reply view module. Given three labels obtained from each module, the final label of the source tweet is determined via majority voting.

Experiments are conducted on a Weibo corpus. Weibo is a Chinese social media platform. The results show that the proposed framework outperforms several baselines for both rumour and non-rumour classes. Specifically, it achieves F1-score of 0.955, precision of 0.944, recall of 0.966, and accuracy of 0.956. The authors conduct an additional experiment to evaluate the effectiveness of the BERT in rumour detection. For this experiment, a BERT model is pre-trained using the input corpus for rumour detection, which obviously results in a good classification result (i.e. F1-Score of 0.960 on the rumour class). Note that the pre-trained BERT in this experiment is different from the BERT pre-trained on a corpus for sentiment analysis in the sentiment view module. The results show that the pre-trained BERT model outperforms the proposed multi-view model. Moreover, a model which combines the three modules and pre-trained BERT only show a marginal performance gain (i.e. 0.04% in F1-score) compared to the pre-trained BERT model for the rumour class.

A rumour detection architecture based on reinforcement learning.

Zhou et al. (2019a)

Little work exploits reinforcement learning for rumour detection. Zhou et al. (2019a) propose a GRU-based architecture and integrates reinforcement learning to guide ERD based on classification accuracy. The framework consists of a rumour detection module (RDM) and a checkpoint module (CM). The input of the former is a source tweet and its replies. Firstly, the input is fed into an embedding layer followed by a max pooling layer which extracts important representations. Next, GRU is applied to the output of max pooling layer and the final state is fed into a Softmax layer which gives a probability distribution over binary classes. The latter (i.e. CM) aims to determine the minimum number of posts for identifying a rumour. Its input is the hidden states of GRU in the RDM. Reinforcement learning learns the optimum decision for a given task using rewards and punishment. In this architecture, the RDM's classification accuracy is used as rewards and an increase in the number of posts used for training is used as punishment. Specifically, a two-layer feed-forward network computes "action values" by learning a function which computes the overall expected reward assuming the model performs an action (i.e. performing classification or receiving more posts for training) at each time step.

As two modules work together, they are jointly trained. Firstly, the RDM is pre-trained using cross-entropy and parameters after pre-training are fixed for training the CM. As described above, the hidden state of GRU at each time step is used to compute action values. Training converges as rewards stabilises. Experiments are conducted on Ma data (Ma et al., 2016). RDM (without CM) trained with the entire training data is used as a baseline to show the effectiveness of the CM. The full architecture (RDM+CM) is more effective than the RDM only model for ERD as the latter takes around 12.5 and 16.6 more hours to achieve the accuracy obtained by the former for Weibo and Twitter data sets, respectively. The full architecture achieves F1-score of 0.785, precision of 0.843, recall of 0.735, and accuracy of 0.858 on the Twitter corpus.

2.4 DATA FOR RUMOUR DETECTION

2.4.1 *PHEME (6392078; Kochkina et al. (2018a) and Zubiaga et al. (2016a))*

PHEME data consists of rumour and non-rumour source tweets and their replies associated with 9 real-world events. Table 2.15 shows the total number of tweets including source tweets and replies as well as the number of rumour and non-rumour source tweets for the task of rumour detection. The number of unique rumours for each event is also given in the table. Data sizes and class distributions vary greatly between events. Overall, the number of rumours is smaller than that of non-rumours for all the events. The total number of tweets in the PHEME data is limited to $105,354$. The 9 events are categorised into two types: (1) breaking news events which potentially produce several rumours and (2) preselected rumours.

The following five events are related to **breaking news**:

- **Germanwings plane crash**: Germanwings Flight 9525, which was an international passenger flight from Barcelona in Spain to Düsseldorf in Germany, was deliberately crashed by the co-pilot in 24 March, 2015. All 144 passengers and a crew of six were killed.
- **Sydney siege**: a gunman held hostages including ten customers and eight employees of a Lindt chocolate café in Martin Place in Sydney, Australia, between 15 and 16 December, 2014. Three people including the hostage taker were killed.
- **Ferguson unrest**: protests and riots sparked by the fatal shooting of an African American teenager by a white Ferguson police officer in Ferguson, Missouri, on 9 August 2014. Riots have lasted until 25 August 2014.
- **Ottawa shooting**: a Canadian soldier was shot dead during shootings at Parliament Hill in Ottawa, Canada, on 22 October, 2014.
- **Charlie Hebdo shooting**: two armed Muslim brothers forced their way through the offices of a French satirical newspaper, Charlie Hebdo, in Paris, France, on 7 January, 2015, and killed 12 people.

The following four events are **preselected rumours**:

- **Putin missing**: a rumour that the Russian president Vladimir Putin had not been seen in public for 10 days since 5 March, 2015. He appeared in public for the first time in 10 days.
- **Prince to play in Toronto**: a rumour that a singer Prince would perform a secret show in Toronto, on 4 November, 2014, started circulating the day before.
- **Gurlittt collection**: a rumour that a Nazi-era art collection belonging to Cornelius Gurlitt is being accepted by the Bern Museum of Fine Arts circulated in November 2014. The rumour turned out to be true.
- **Michael Essien contracted Ebola**: a rumour that the AC Milan midfielder Michael Essien had contracted Ebola was posted by a Twitter user on 12 October, 2014. The rumour was denied by Essien.

The generation of the PHEME data set is threefold. Firstly, given a collection of tweets related to newsworthy events, tweets that prompted a large number of retweets (100) are sampled. Secondly, given rumour criteria, journalists manually identify rumour and non-rumour source tweets. Finally, replies of the source tweets are collected and journalists manually annotate the source tweets.

Table 2.15: The number of rumour and non-rumour source tweets in the PHEME (6392078) data

Event	**Tweets**	**Rumours**	**Non-rumours**	**Unique rumours**
Germanwings-crash	4, 489	238	231	19
Sydney siege	23, 996	522	699	61
Ferguson unrest	24, 175	284	859	41
Ottawa shooting	12, 284	470	420	51
Charlie Hebdo	38, 268	458	1, 621	60
Putin missing	835	126	112	6
Prince toronto	902	229	4	5
Gurlitt	179	61	77	3
Ebola Essien	226	14	0	1
Total.	105, 354	2, 402	4, 023	247

2.4.2 *Twitter15/16 (Ma et al., 2017; Ma et al., 2016; Liu et al., 2015)*

Twitter15/16 data is an extension of Twitter15 (Liu et al., 2015) and Twitter16 (Ma et al., 2016). Ma et al. (2017) reconstructed the benchmark data sets by adding propagation threads (i.e. retweets and replies) and constructing propagation trees for each source tweet. The original Twitter15 data includes tweets reporting newsworthy rumours and non-rumours confirmed by a fact-checking website (Snopes.com) and a rumour-tracking website (emergent.info) until March 2015. Similarly, the original Twitter16 data consists of tweets reporting rumours and non-rumours confirmed by Snopes.com between March and December 2015. To balance class distributions, Ma et al. (2016) add events related to non-rumours using publicly available data sets. Table 2.16 shows the number of rumour and non-rumour source tweets as well as that of their threads (i.e. retweets and replies).

Table 2.16: The number of rumour and non-rumour source tweets, and threads in Twitter15/16 data set.

	Twitter15	**Twitter16**
Non-rumours	374	205
False rumours	370	205
True rumours	372	205
Unverified rumours	374	203
Threads	333, 612	204, 820

2.4.3 *Ma data (Ma et al., 2016)*

This data contains rumour and non-rumour source tweets and corresponding threads (i.e. replies and retweets) for two social media platforms, Twitter and Sina Weibo. For the Twitter data, rumours and non-rumours confirmed by a rumour debunking website, Snopes.com, are used as source tweets. Threads of each source tweet are also collected. Unlike Twitter, Weibo has a system for reporting rumours. The Weibo data set is built using the system. Table 2.17 shows the statistics of each data set. The "Total" row of the table shows the total number of source tweets and threads.

Table 2.17: The number of rumour and non-rumour source tweets in the data generated by Ma et al. (2016).

	Twitter	**Weibo**
Non-rumours	494	2,351
Rumours	498	2,313
Total	1,101,985	3,805,656

2.4.4 *Qazvinian data (Qazvinian et al., 2011)*

This data includes 10,417 tweets reporting five preselected rumours confirmed by a premier reference site ThoughtCo.[5] as described in Table 2.18. To build the data set, tweets which match pre-defined REGEX patterns for the five rumours are collected using Twitter's search API. For example, tweets matching a query "`Obama & (muslim|islam)`" are collected. Subsequently, two human annotators manually examine the collected tweets to filter out tweets that match the REGEX patterns but are not actually relevant to rumours.

Table 2.18: The description and statistics of the Qazvinian data.

Event	**Description**	**Veracity**	**Tweets**
obama	Is Barack Obama muslim?	False	4,975
airfrance	Air France mid-air crash photos?	False	505
cellphone	Cell phone numbers going public?	Mostly false	215
michelle	Michelle Obama hired too many staff?	Partly true	299
palin	Sarah Palin getting divorced?	False	4,423

2.5 EARLY SIGNALS FOR RUMOURS ON SOCIAL MEDIA

RQ 2.1: What are signals which characterise rumours in the early stages of their evolution?

This section is highly related to Chapter 5 and addresses the RQ 2.1. Based on the literature review in Section 2.3, this section aims to understand what are weak yet effective signals that can discriminate between rumours and

[5] https://www.thoughtco.com/

non-rumours even in the very early stages of event diffusion using publicly available rumour data sets.

2.5.1 *Motivation*

While some authors have explored signals for rumours which are generally useful for distinguishing rumours from non-rumours, not much work has studied "early" signals for rumours. Similar work to the task of this section is (Zhao et al., 2015). The key motivation is to identify potential rumours before a widespread of related memes with a low computational cost. According to the authors, the computational cost of existing approaches for ERD tends to be high because they first identify trending or bursty topics from an entire corpus using topic models such as Latent Dirichlet Allocation (LDA; Blei et al. (2003)), and then detect rumours. To avoid this and detect rumours as early as possible, they propose a method for ERD based on enquiry and correction patterns of social media users. A set of REGEX such as "is (that | this | it) true" and "(rumor | debunk)" are used to identify tweets expressing questions and corrections (*signal tweets*). The method then extracts a single summary tweet for each cluster of similar signal tweets. Finally, potential rumours are identified by ranking candidate rumour clusters based on their statistical features. The identification of signal tweets based on linguistic patterns and the use of clusters show a reduction in computational cost compared with existing techniques based on topic modelling. However, its results are dependent on the selection of REGEX patterns. For example, based on a quick analysis I performed, the majority of hand-labelled rumour source tweets in the *PHEME* data introduced in Section 2.4 do not contain the proposed REGEX patterns; only 74 out of 2,402 rumour source tweets can be identified using the method and 45 tweets belong to one event, "Prince to play in Toronto". Other possible signals such as user behavioural patterns and the structure of information diffusion become available or meaningful when events have already become widespread (Kwon et al., 2017; Zhao et al., 2015), and therefore they are not "early" signals.

Several studies on rumours on social media have found that rumours posted during the early stages of event diffusion are mostly simple copies of sources (i.e. retweets; Maddock et al. (2015) and Zhao et al. (2015)). In other words, when a rumour starts to get popular, it will get many retweets. This will eventually result in a high volume in event time series plots. Based on the insight gained from the related work, this section proposes that *bursts* identified using temporal signals are the key to the early identification of potential rumours. Temporal signals should be robust to short-term fluctuations exhibited in event evolution on social media and be portable to unseen events in different domains. Several studies have studied bursts in the popularity of events on social media (Hu et al., 2017; Kong et al., 2015; Matsubara et al., 2012; Wang et al., 2016). The main focus of existing approaches is often "peaks". However, peaks indicate that rumours have already become popular (i.e. a large amount of information regarding the rumours is already available), and hence they are not "early" signals for rumours (Zhao et al., 2015). In this thesis, a "burst" is defined as a sudden increase in the number of social media posts for a short period following the definition of a major dictionary[6] and existing work leveraging bursts in Twitter (Lee et al., 2011; Myers and Leskovec, 2014). In particular, bursts and peaks used to identify potential rumours are collectively referred to as *key bursts*.

To recap, this section studies several features used to characterise rumours on social media in order to identify early signals for rumours. These features have been extensively studied by early work on rumour detection. Based on a thorough analysis and discussion, it is found out that *temporal signals* are key signals for rumours.

2.5.2 *Candidates for Early Signals for Rumours*

This section studies three types of *contextual features* (i.e. tweet-level, user-based, and temporal features) of rumours and non-rumours to identify early signals for rumours. As detailed in Section 2.1.3, several studies have claimed that contextual features and temporal patterns of event evolution are effective in characterising rumours on social media. In particular, early approaches for rumour detection are based on supervised learning techniques, and thus researchers have extensively studied manually curated features related to contents, users, networks, and temporal features to seek distinguishing features of online rumours (Qazvinian et al., 2011; Kwon et al., 2017; Yang et al., 2012; Sun et al., 2013; Zhao et al., 2015; Zhang et al., 2015c; Wu et al., 2015; Ma et al., 2015; Liu et al., 2016; Zubiaga et al., 2017; Hamidian and Diab, 2016). These studies have shown that hand-crafted features have the potential for distinguishing rumours from non-rumours. Recent research states that the behaviour of rumour spreading on social media, which is characterised by bursts and/or spikes, provides an important signal for the nature of rumours (Shin et al., 2018a). Based on these findings, this section investigates whether they can provide weak forms of supervision for identifying potential rumours.

This section explores shallow features, which can be extracted from Twitter's API without painstakingly complicated and domain-specific feature engineering, to determine early signals for rumours. In specific, 24 hand-crafted features grouped into two categories (i.e. tweet-level and user-based features) are investigated. They have extensively been studied by early work on rumour detection based on supervised learning (see Section 2.3.3.1). A prime requirement for an early signal is that it should be able to distinguish between rumours and non-rumours.

Tweet-level and user-based features are extracted from every source tweet of each event in the augmented data created in Chapter 4 (see Section 2.5.3). Table 2.19 shows the list of features which have been widely explored in studies on rumours on Twitter and are examined in this chapter. For each feature, a scatter plot for the two classes is generated to examine whether the feature can be useful in separating them. The x-axis of each plot is source tweet index and the y-axis is the feature value of each tweet.

As for temporal features, rumour and non-rumour source tweets of each event are ordered in chronological order and a time series plot for each class is generated. The x-axis is time and y-axis is the number of tweets. Whether

[6] https://dictionary.cambridge.org/dictionary/english/burst

Table 2.19: Description of hand-crafted features.

Tweet-level features
Number of retweets
Number of favourites
Whether tweet has a question mark
Whether tweet contains URLs
Number of URLs embedded in tweet
Whether tweet has native media*
Number of words in tweet
User-level features
Number of posts user has posted
Number of public lists user belongs to
Number of followers
Number of followings
User reputation (i.e. followers/(followings+1))
User reputation (i.e. followers/(followings+followers+1))
Number of tweets user has liked so far (aka "user favourites")
Account age in days
Whether user is verified
User engagement (i.e. # posts / (account age+1))
Following rate (i.e. followings / (account age+1))
Favourite rate (i.e. user favourites / (account age+1))
Whether user has a description
Number of words in user description
Number of characters in user's name including white space
Whether user has a background profile image
Whether geolocation is enabled

* multimedia shared with the Tweet user-interface not via an external link

temporal patterns of different classes exhibit different characteristics (e.g. burstiness) is examined.

2.5.3 *Data*

For the analysis of ***tweet-level*** and ***user-based*** features, rumour and non-rumour source tweets for the 9 events in the *PHEME* (6392078*; see Section 2.4.1)* data are used. As the original PHEME data is small for feature analysis, this thesis uses augmented data which will be detailed in Chapter 4. Features listed in Table 2.19 are extracted from all source tweets in the temporally filtered augmented data for 5 breaking news events[7] (i.e. the output of Chapter 4) and the *PHEME (*6392078*)*[8] for the 4 preselected rumour events. For the sake of consistency and clarity, the temporally filtered augmented *PHEME* will hereafter be referred to as *Aug-PHEME-filtered* as defined in Section 4.6.2.2 in Chapter 4. To remove the influence of class imbalance in the input data on feature analysis results, the data was balanced. For the analysis of ***temporal patterns***, *Aug-PHEME-filtered*'s five events are used (see Table 4.9). As the 4 preselected rumour event data sets consists of few source tweets, temporal patterns cannot be obtained. The *PHEME* and *Aug-PHEME-filtered* are used in this experiment because they are the most popular and the largest, publicly available and labelled data for message-level rumour detection and cover a range of real-world events.

2.5.4 *Results*

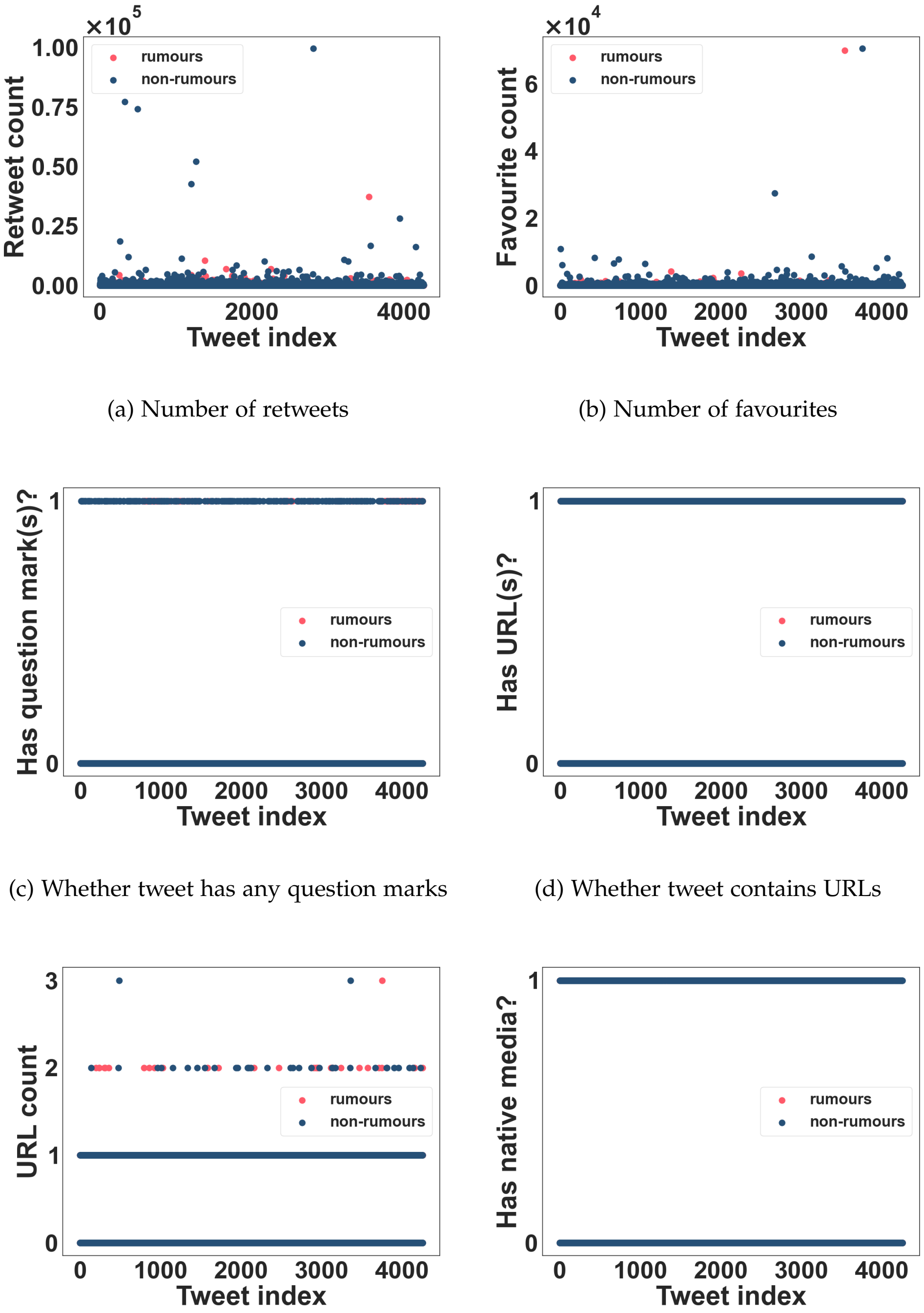
Figure 2.1 visualises tweet-level and user-based features for rumour and non-rumour source tweets. Binary features (Figure 2.1c, 2.1d, 2.1f, 2.1h, 2.1q, 2.1w, and 2.1x) such as "Whether a user is verified or not" for rumours and non-rumours are completely overlapping. The results also show that rumours and non-rumours are not separable using numerical features (e.g. Figure 2.1a, 2.1b, 2.1e, etc.), which indicates that such features are not useful for characterising rumours on their own. It should be noted that this section does not aim to identify the optimal combination of numerical features for rumour detection. Rather, its aim is to identify simple yet discriminative early signals (i.e. features which can distinguish rumours from non-rumours without training ML model). Therefore, studying to what extent combinations of two or more features can discriminate between rumours and non-rumours is beyond the scope of this section. Such a study will be conducted in Chapter 6.

Following visualisation practice in (Kwon et al., 2017), the Log-Log plots of Complementary Cumulative Distribution Function (CCDF) of the features aggregated over rumour and non-rumour source tweets are also presented in Figure 2.2. The figure also shows that tweet-level and user-based features tend to overlap across the majority of quantile values, which suggests that they are not useful for distinguishing rumours from non-rumours. They might be complementary and their combinations might boost rumour detection

[7] available via https://zenodo.org/record/3269768

[8] available via https://figshare.com/articles/PHEME_dataset_for_Rumour_Detection_and_Veracity_Classification/6392078

models. However, the analysis results suggest that each numerical feature in isolation is not a good signal.



(a) Number of retweets

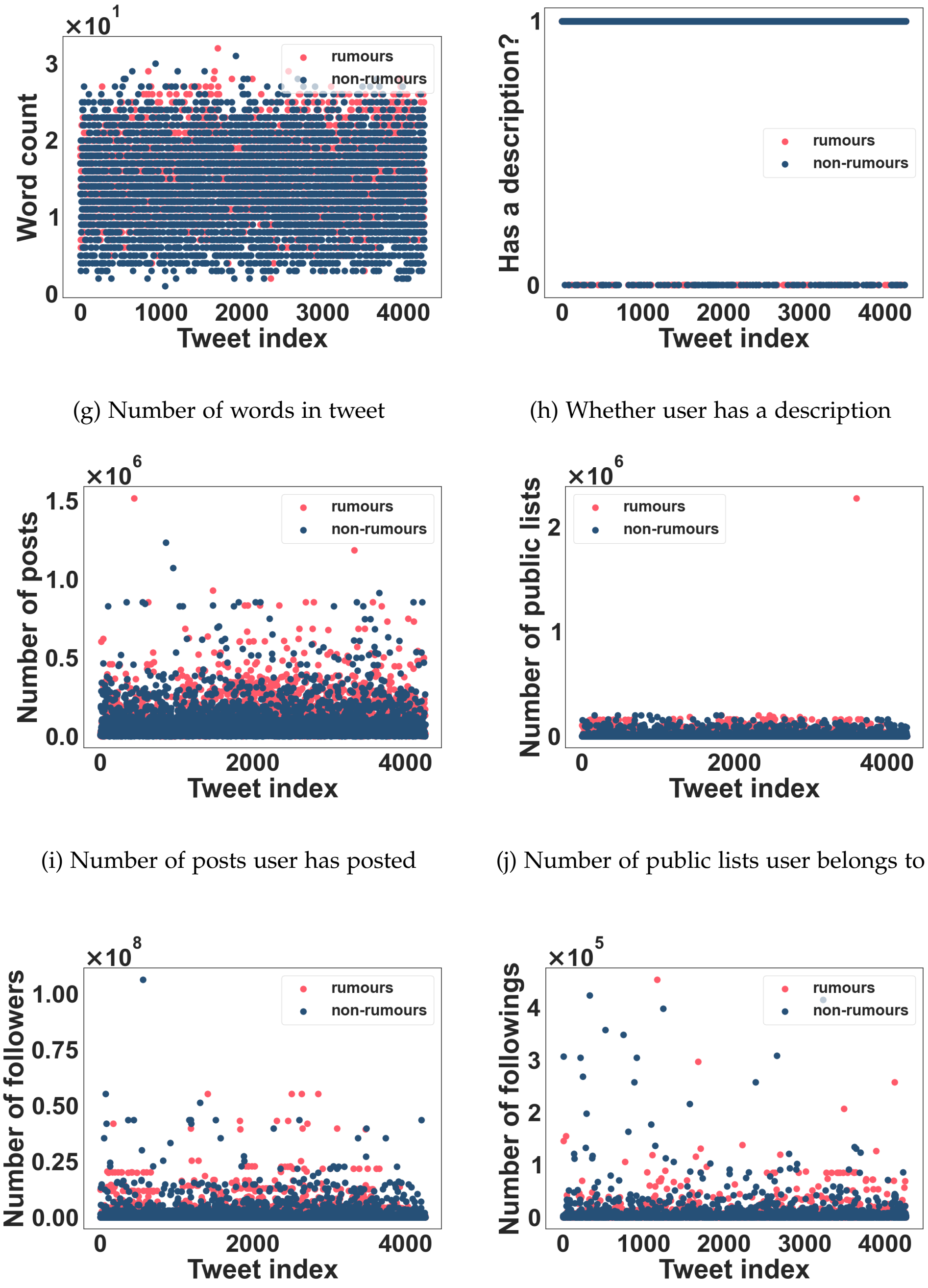
(b) Number of favourites

(c) Whether tweet has any question marks

(d) Whether tweet contains URLs

(e) Number of URLs embedded in tweet

(f) Whether tweet has native media



(g) Number of words in tweet

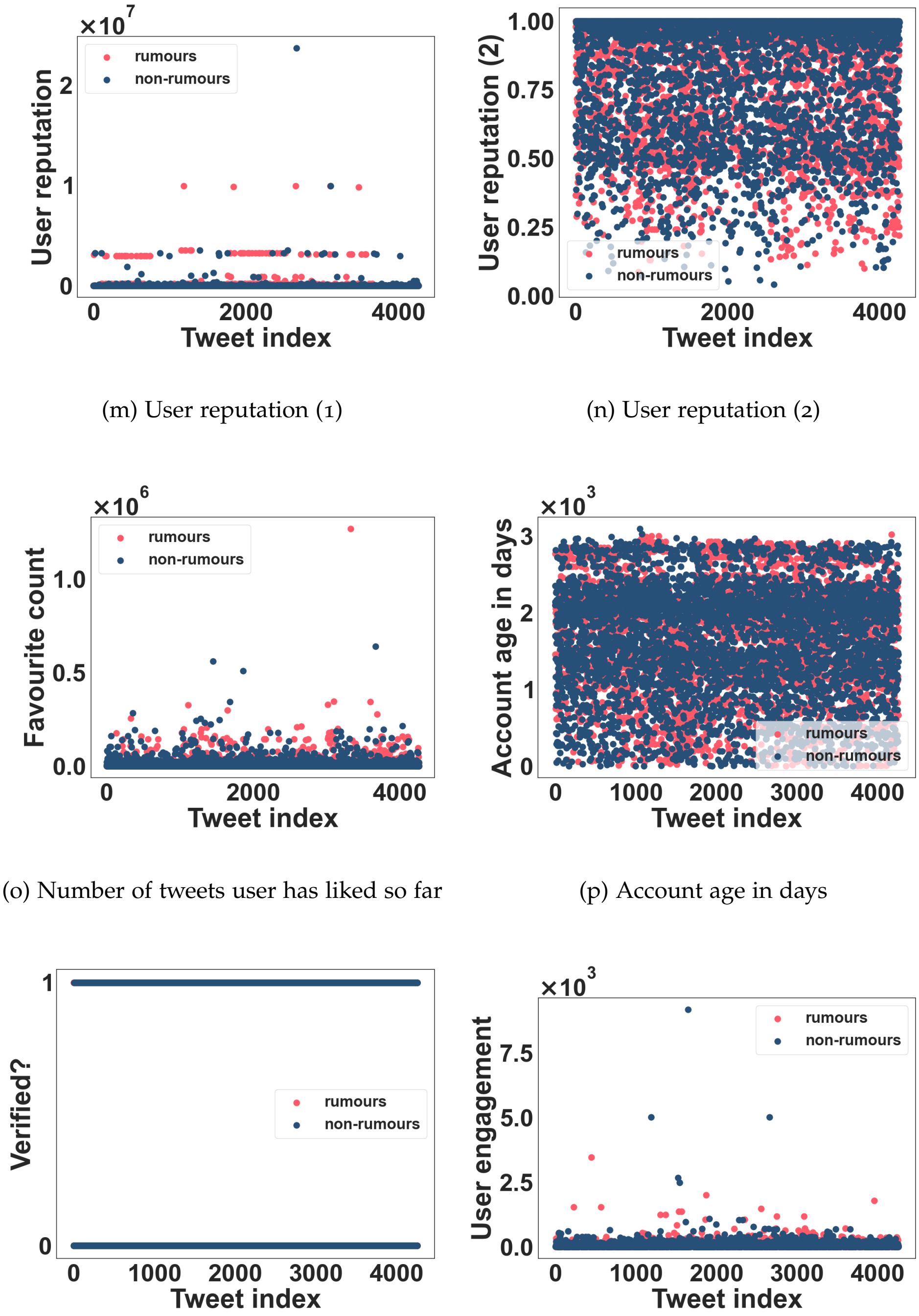
(h) Whether user has a description

(i) Number of posts user has posted

(j) Number of public lists user belongs to

(k) Number of followers

(l) Number of followings



(m) User reputation (1)

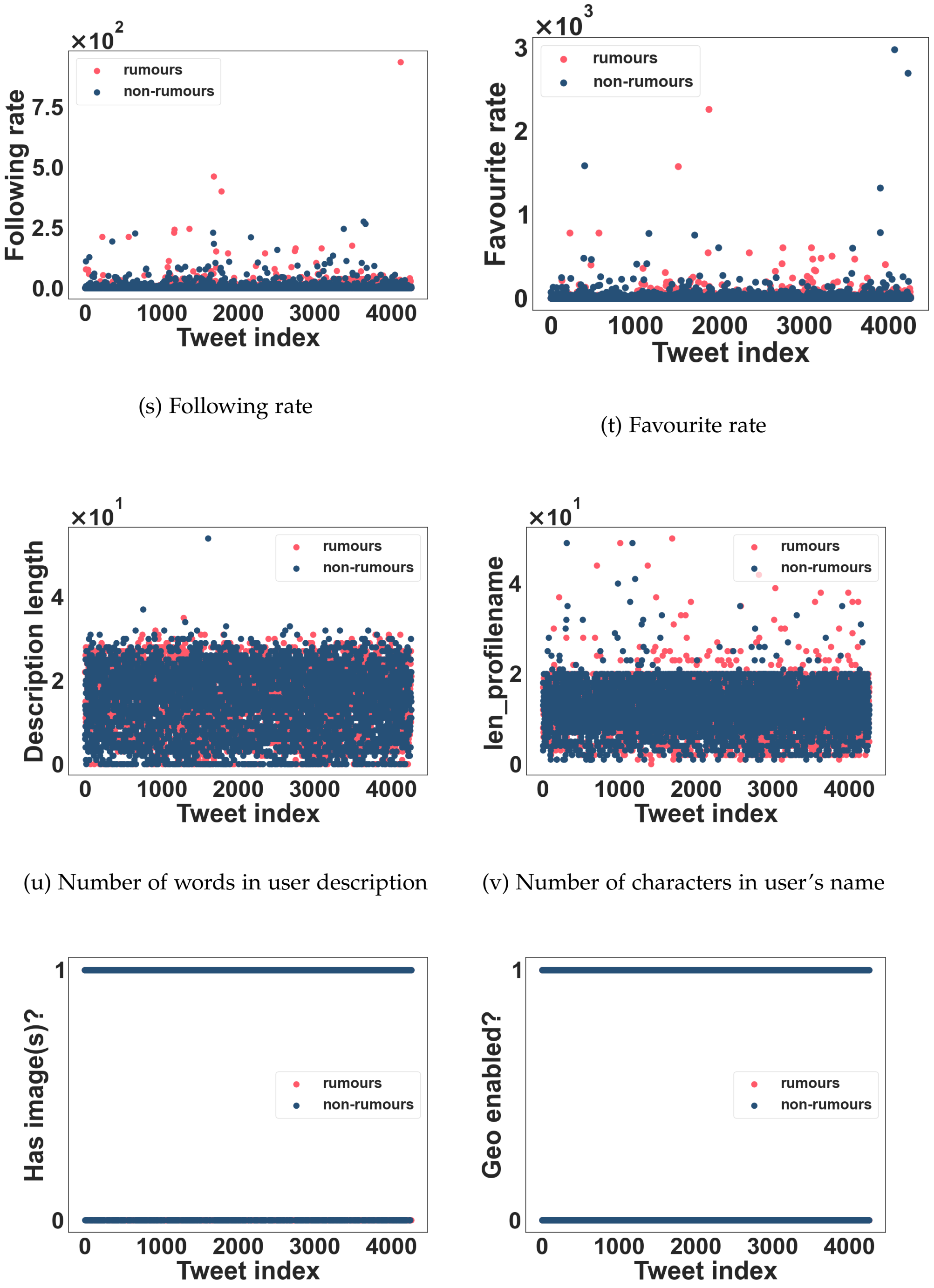
(n) User reputation (2)

(o) Number of tweets user has liked so far

(p) Account age in days

(q) Whether user is verified

(r) User engagement



(s) Following rate

(t) Favourite rate

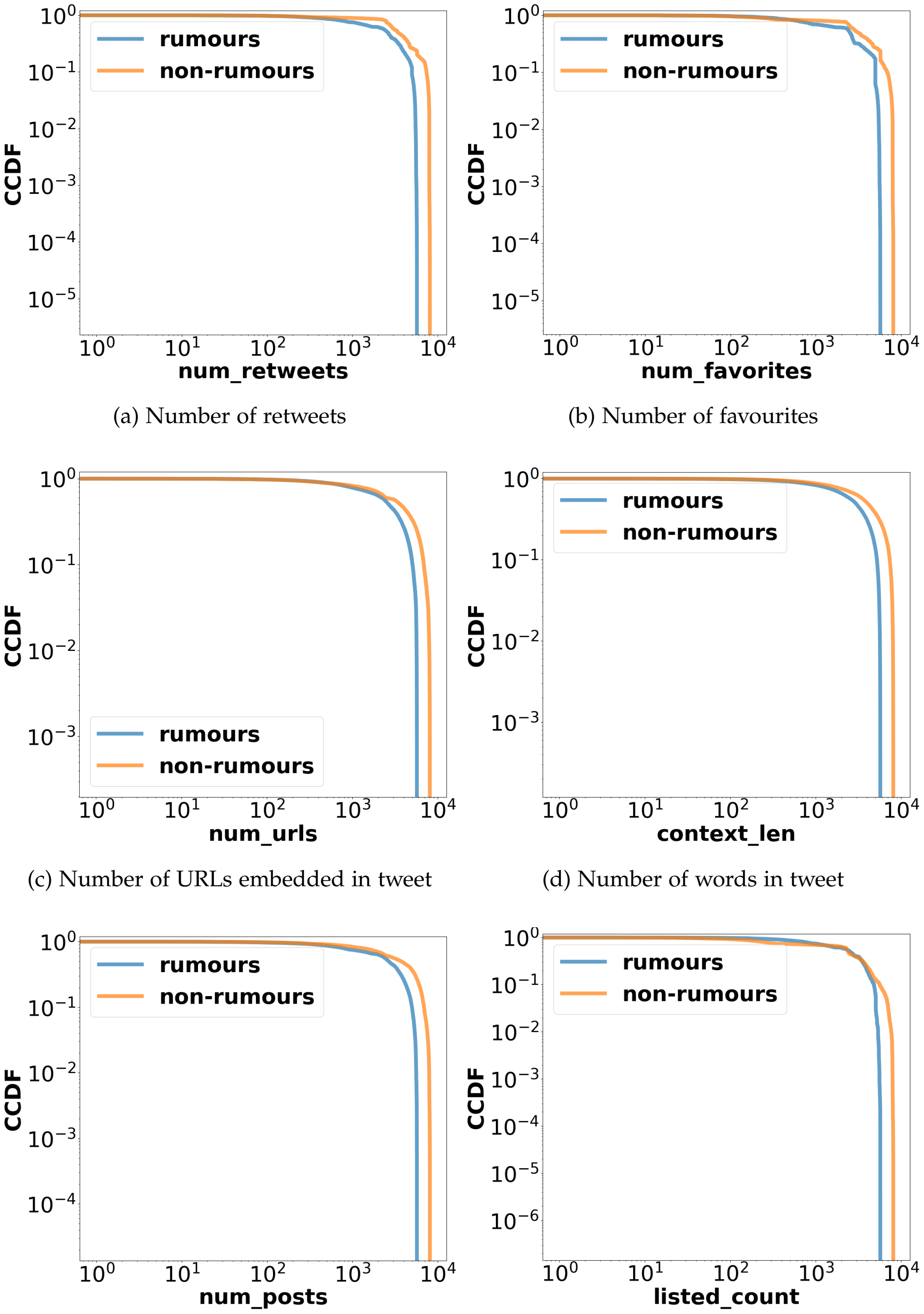
(u) Number of words in user description

(v) Number of characters in user's name

(w) Whether user has a profile image

(x) Whether geolocation is enabled

Figure 2.1: Visualisation of hand-crafted features for the five events in the PHEME5. The horizontal axis on the graph shows tweet index and the vertical axis represents feature values.



(a) Number of retweets

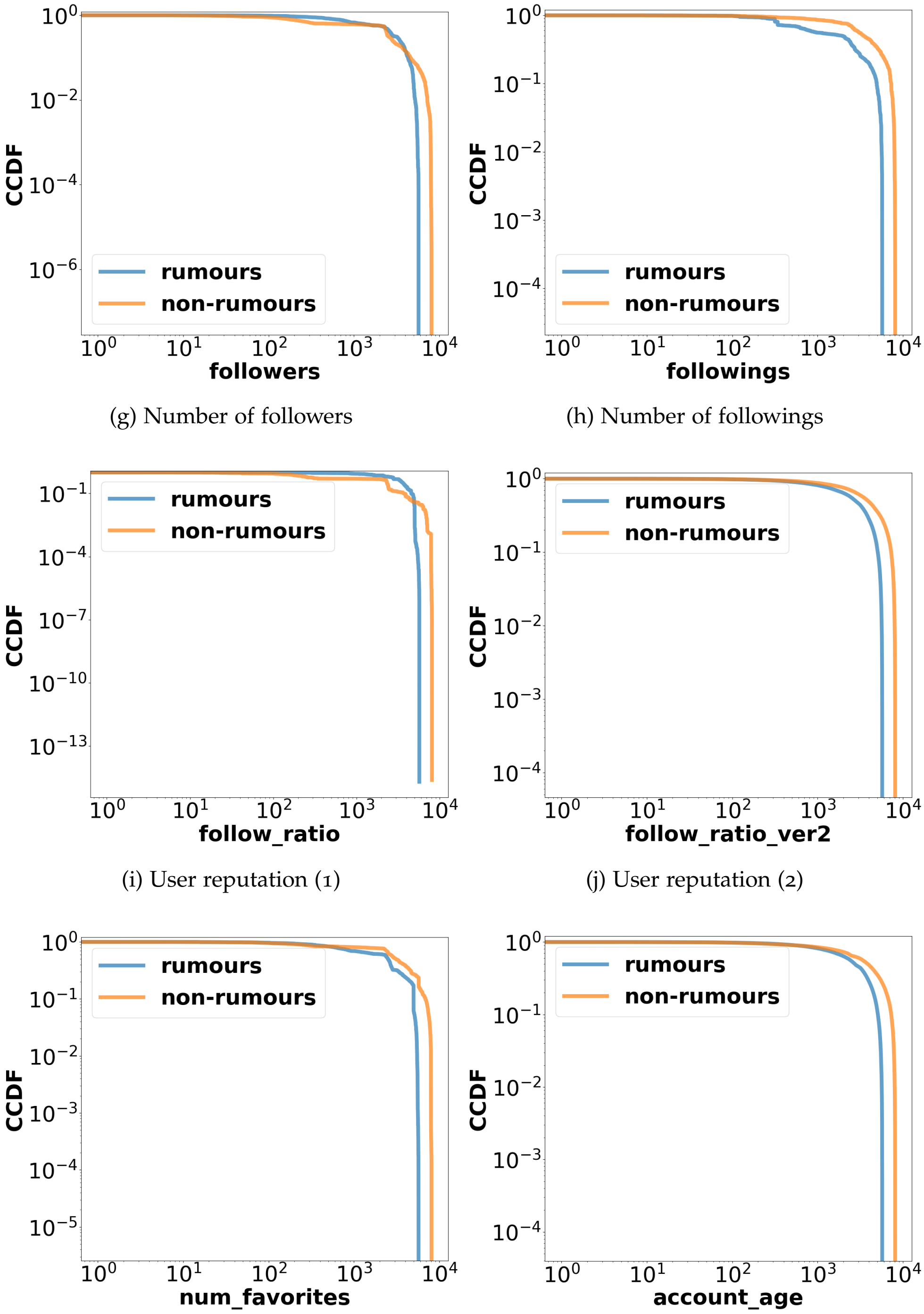
(b) Number of favourites

(c) Number of URLs embedded in tweet

(d) Number of words in tweet

(e) Number of words in tweet

(f) Whether user has a description



(g) Number of followers

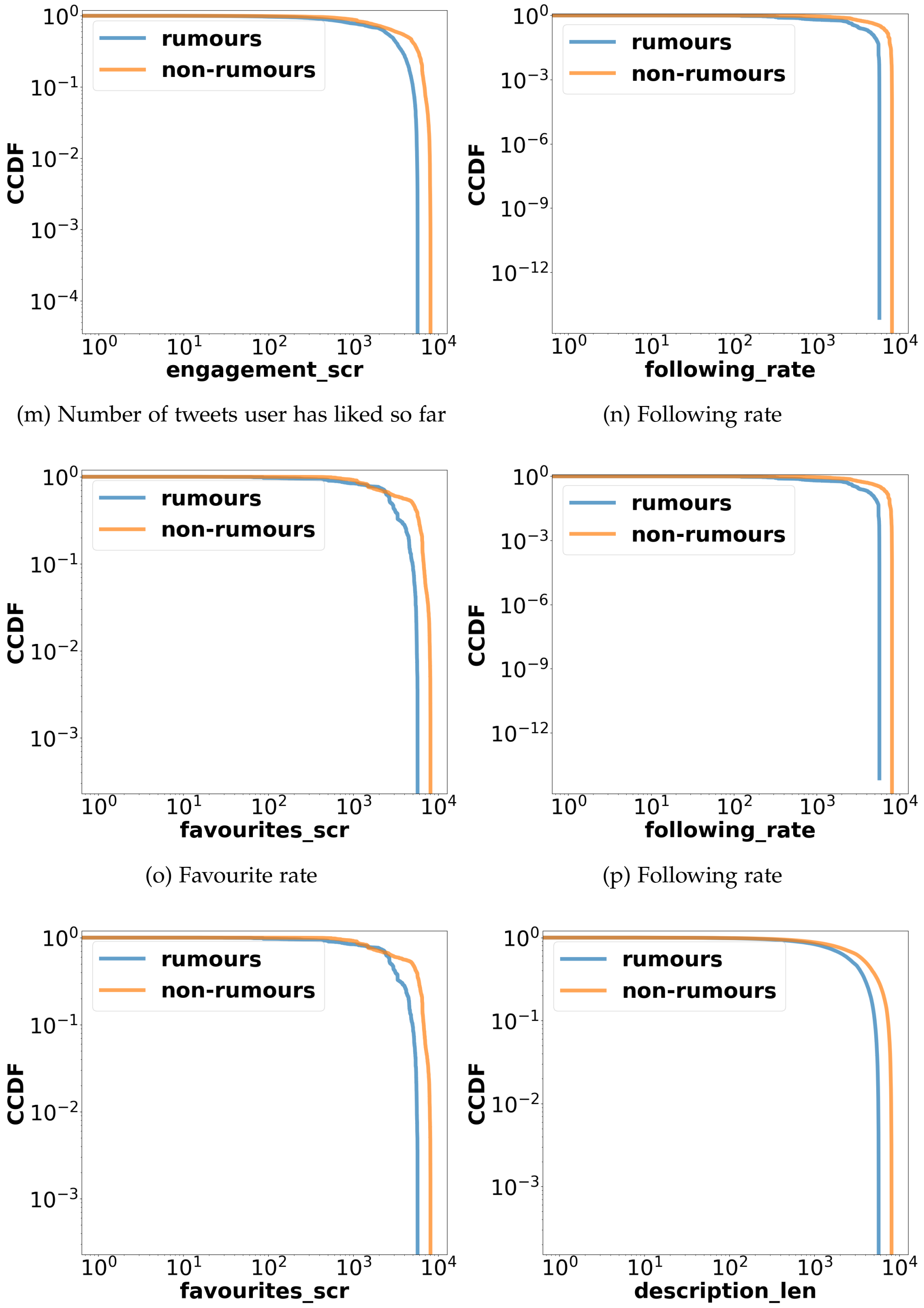
(h) Number of followings

(i) User reputation (1)

(j) User reputation (2)

(k) Number of tweets user has liked so far

(l) Account age in days



(m) Number of tweets user has liked so far

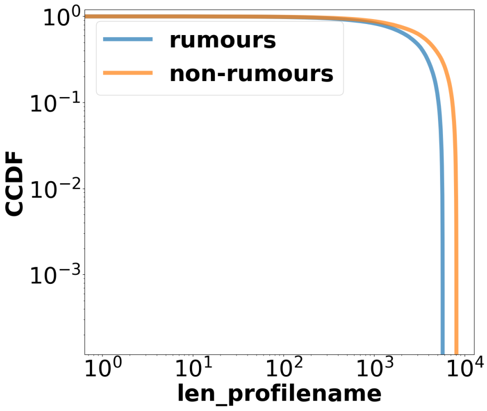
(n) Following rate

(o) Favourite rate

(p) Following rate

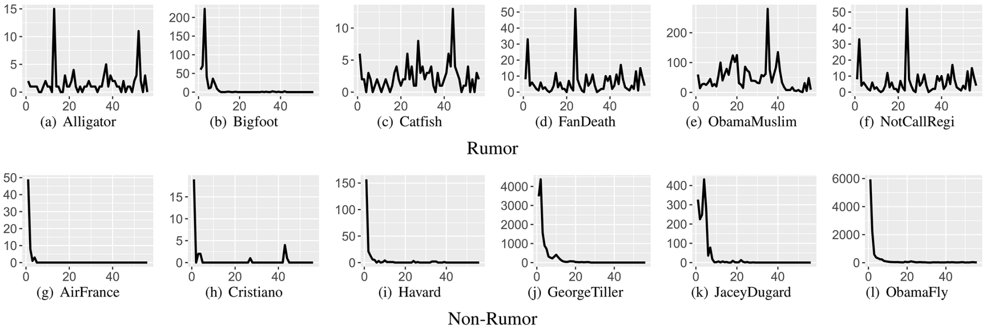
(q) Favourite rate

(r) Number of words in user description



(s) Number of characters in user's name

Figure 2.2: Log-Log plots of CCDF of numerical features for the five events in the PHEME5. The horizontal axis on the graph shows feature values and the vertical axis represents CCDF.



(a) Alligator (b) Bigfoot (c) Catfish (d) FanDeath (e) ObamaMuslim (f) NotCallRegi

Rumor

(g) AirFrance (h) Cristiano (i) Havard (j) GeorgeTiller (k) JaceyDugard (l) ObamaFly

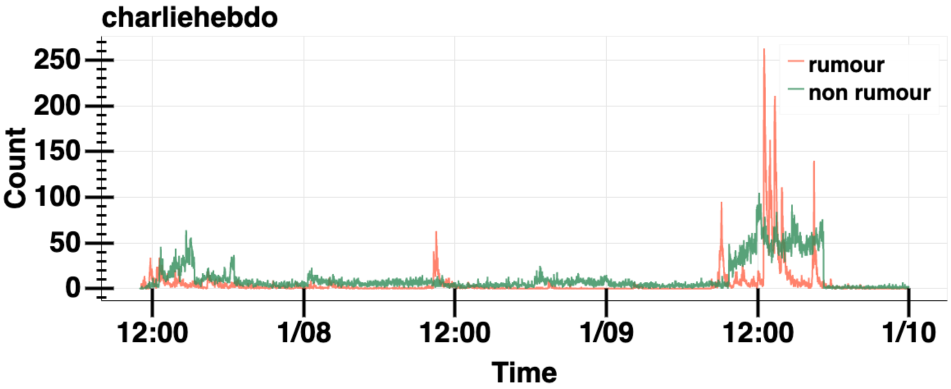
Non-Rumor

Figure 2.3: Time series plots of rumour and non-rumour events in (Kwon et al., 2017). The x-axis is time in days and y-axis is the number of tweets.

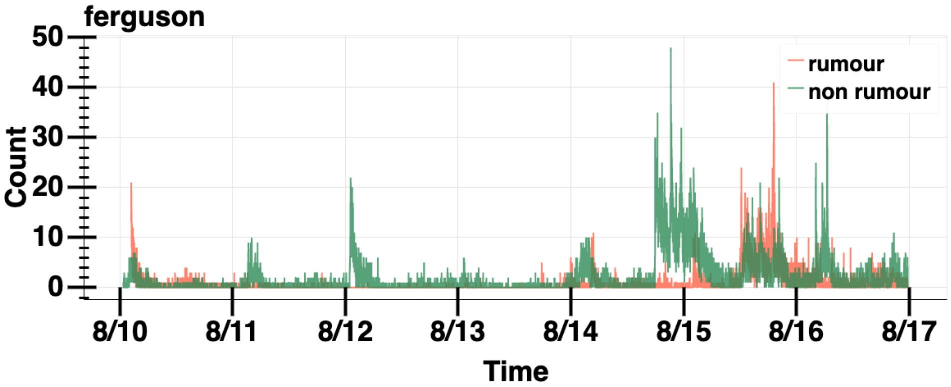
Figure 2.4 visualises time series plots of tweets with weak labels (i.e. rumours and non-rumours) in the Aug-PHEME-filtered. Different events have different evolution patterns. Although there are some overlaps between rumours and non-rumours, time series plots for rumours and those for non-rumours exhibit different temporal patterns (Kotteti et al., 2018; Kwon et al., 2017). Existing work (Kwon et al., 2017) conducted similar analysis. Its results illustrated that rumours tend to exhibit several recurring spikes while non-rumours show a single significant peak as shown in Figure 2.3. The results of this section show that rumours are more bursty and tend to form larger spikes in comparison with non-rumours in the very early stages of event diffusion (i.e. within the first few hours after an event occurs). Also, there exist some periods where the popularity of rumours shows spikes, while that of non-rumours is less active (e.g. less bursty, low frequency), and vice versa. Based on these observations, this chapter suggests that using temporal patterns as early signals is a promising attempt for the identification of potential rumours.

2.5.5 *Discussion*

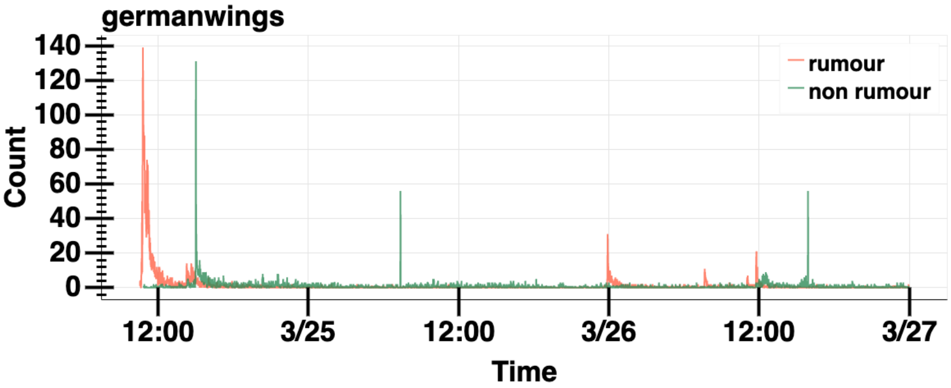
This section described background on the problem of the early identification of potential rumours which is researched in Chapter 5. In specific, this section studied why temporal signals are the key to the research problem by analysing tweet-level and user-based features of rumour and non-rumour source tweets. These features have been widely studied by early work on rumour detection which employs supervised learning techniques (see Section 2.3.3.1). Their findings report that these features are useful for characterising rumours. This section investigated whether a feature can separate rumours from non-rumours on its own in the early stages of event evolution without the aid of ML algorithms which require training. The feature analysis results show that temporal patterns are a promising early signal for rumours.



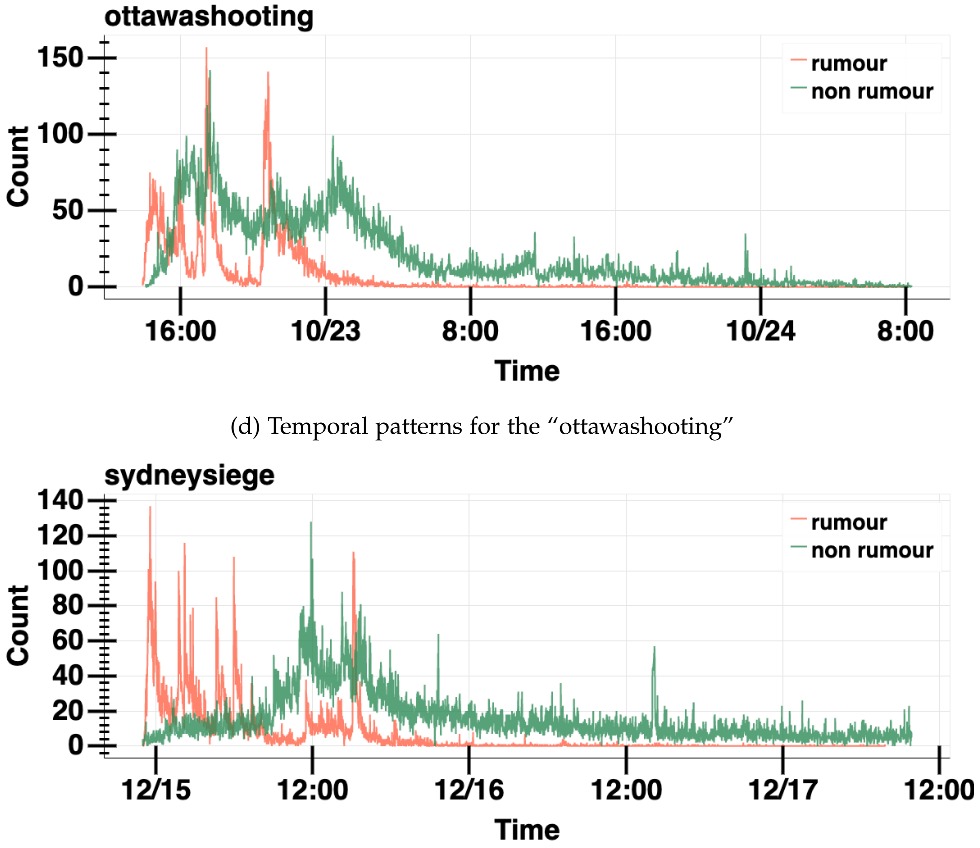
(a) Temporal patterns for the "charliehebdo"



(b) Temporal patterns for the "fergusonunrest"



(c) Temporal patterns for the "germanwings"



(d) Temporal patterns for the "ottawashooting"

(e) Temporal patterns for the "sydneysiege"

Figure 2.4: Temporal patterns for rumour and non-rumour source tweets in the Aug-PHEME-filtered. The horizontal axis on the graph shows time and the vertical axis represents the number of tweets.

2.6 BACKGROUND: MODELS

This section introduces three models which will be mentioned throughout this thesis: 1) an SOTA NLM called ELMo, 2) LSTM and 3) soft attention mechanisms. In particular, the first model is used in Chapter 4 and 6 and the others are used in Chapter 6.

2.6.1 *Embeddings for Language Models (ELMo)*

ELMo is adopted to learn the effective representation of tweets. ELMo provides deep, contextualised, and character-based word representation by using a bidirectional LSTM-based LM (Peters et al., 2018). At each time step (i.e. each token), a *forward LM* computes the probability of an input sequence by modelling that of the target token given historical observations (i.e. previous tokens in the sequence). A *backward LM* computes the probability for each token in reverse given future observations. ELMo computes a linear combination of the states of the two bidirectional LSTM layers and token embeddings to encode each word in the input.

Formally, for each token t_k, a $L-$layer BiLM computes a set of $2L+1$ representations defined as

$$\begin{aligned} R_k &= \{\mathbf{x}_k^{LM}, \overrightarrow{\mathbf{h}}_{k,j}^{LM}, \overleftarrow{\mathbf{h}}_{k,j}^{LM} \mid j = 1, \cdots, L\} \\ &= \{\mathbf{h}_{k,j}^{LM} \mid j = 0, \cdots, L\}, \end{aligned}$$

where $\mathbf{h}_{k,0}^{LM}$ is the token layer (i.e. context insensitive token representation) and $\mathbf{h}_{k,j}^{LM} = [\overrightarrow{\mathbf{h}}_{k,j}^{LM}, \overleftarrow{\mathbf{h}}_{k,j}^{LM}]$, for each bidirectional LSTM layer. A task-specific weight of all BiLM layers (i.e. ELMo vector) is defined by

$$\mathbf{ELMo}_k^{\text{task}} = \gamma^{\text{task}} \sum_{j=0}^{L} s_j^{\text{task}} \mathbf{h}_{k,j},$$

where $\mathbf{s}^{\text{task}} = \{s_j^{\text{task}} \mid j = 0, \cdots, L\}$ are softmax-normalised weights and γ^{task} is a scalar for scaling the entire ELMo vector. The trained BiLM weights are frozen and the ELMo vector $\mathbf{ELMo}_k^{\text{task}}$ and the token layer $\mathbf{x}_k$ are concatenated into a single representation $[\mathbf{x}_k; \mathbf{ELMo}_k^{\text{task}}]$. The ELMo enhanced representation of the input sequence is passed into a task-specific model (e.g. RNNs for classification). The ELMo vector can also be concatenated with each hidden state $\mathbf{h}_k$ of the task-specific model for further performance improvements by replacing $\mathbf{h}_k$ with $[\mathbf{h}_k; \mathbf{ELMo}_k^{\text{task}}]$.

ELMo represents each token based on contextual information obtained from a sequence to which it belongs to, and thus, it overcomes limitations of conventional word embeddings in which each token is represented as an average of its several different contexts (Perone et al., 2018).

2.6.2 *Long Short-Term Memory*

RNNs have been actively employed in sequence modelling. However, vanilla RNNs suffer from vanishing and exploding gradients which can prevent a network from further training. LSTM (Hochreiter and Schmidhuber, 1997) can tackle these problems by introducing memory cells and learn long-range dependencies in an input sequence. A recurrent layer has memory cells for storing memory state information and different gates which regulate the flow of information for cells. This structure allows LSTM networks to learn long-range dependencies of online rumour evolution, and thus utilise contextual information based on conversational threads. A standard LSTM unit consists of a forget gate, input gate, output gate, and cell state. Its output is called a hidden state. Equations for LSTM networks (Hochreiter and Schmidhuber, 1997) are given by

$$\begin{aligned}
f_t &= \sigma(W_f x_t + W_f h_{t-1} + b_f), \\
i_t &= \sigma(W_i x_t + W_i h_{t-1} + b_i), \\
o_t &= \sigma(W_o x_t + W_o h_{t-1} + b_o), \\
c_t &= f_t \odot c_{t-1} + i_t \odot \sigma_c(W_c x_t + W_c h_{t-1} + b_c), \\
h_t &= o_t \odot \tanh(c_t),
\end{aligned}$$

where f_t, i_t, o_t, c_t and h_t denote a forget gate, input gate, output gate, cell state, and hidden state at time t, respectively. W and b are weight matrices and bias vectors which need to be learnt during training. σ is the sigmoid function which lets input values range between 0 and 1. The operator $\odot$ denotes the element-wise product. The *input gate* i decides the extent to which new information is added to cell state. The *forget gate* controls the extent to which an existing memory in the old cell state is kept in the cell new state. The *memory cell* c is updated with new information computed according to part of the existing memory and that of new values. The *output gate* decides the extent to which information stored in the new cell state is used to compute the hidden state of the LSTM unit.

2.6.3 *Soft Attention Mechanisms*

Attention mechanisms have recently become popular for sequence modelling and transduction models in a wide range of ML tasks. Attention offers two main benefits. One advantage is that it provides importance weights of elements in an input sequence for the prediction of a target (Yang et al., 2016a). Unlike hand-crafted features, input representations generated by deep learning models are not explicable. The visualisation of attention weights can provide a clear insight about which parts of input are useful to output a target. Another benefit is that attention helps a model to represent multi-dimensional contexts of an input sequence as one single compressed representation.

An input sequence is first encoded via LSTM networks. The hidden state h_j of LSTM represents an encoder state at time step j for $(j = 1, 2, \cdots, T)$. Equations for computing an attention-weighted vector for the i^{th} element in an input sequence are given by

$$
\begin{aligned}
e_{ij} &= tanh(W_h h_{ij} + b_h), \\
\alpha_{ij} &= \frac{\exp(e_{ij})}{\sum_{k=1}^{T} \exp(e_{ik})}, \\
c_i &= \sum_{j=1}^{T} \alpha_{ij} h_j
\end{aligned}
$$

where e_{ij}, α_{ij} and c_i denote a score function, attention weights and an attention-weighted sequence vector (Yang et al., 2016a), respectively. W_h and b_h are randomly initialised weights and bias. The context vector c_i is defined as a weighted sum of hidden states h of the input sequence, weighted by attention scores α.

2.7 LIMITATIONS OF CURRENT RUMOUR DETECTION

Based on the literature review, this thesis suggests several limitations of current rumour detection. One major limitation is *poor portability*, which means that it is not easy to generalise and transfer existing rumour detection frameworks to different contexts, setting, and/or domains. There exist several possible causes for this limitation. First of all, the predictive power of widely used hand-crafted features, which have been widely studied for the characterisation of online rumours and used in supervised learning approaches for rumour detection, is highly event-specific and/or rumour-specific (Kwon et al., 2017). These features also often involve careful and close observation, which makes them less generalisable to new settings (e.g. events and rumours). Next, labelled data scarcity and class imbalance lead to poor generalisation. Manual annotation is very limited because it is impossible for humans to read over millions of social media posts. Class imbalance is naturally inherent in the domain of rumour detection and can be commonly observed in most publicly available rumour data sets. Methods for handling class imbalance (e.g. data under- and over-sampling and cost-sensitive learning; Longadge and Dongre (2013)) can lead to over-fitting which makes ML models unable to generalise to new data and results in performance loss (Johnson and Khoshgoftaar, 2019). Despite a growing demand for deep learning-based solutions which are applicable to real-world large-scale data, little work has paid attention to the problem of limited labelled data and class imbalance in the context of rumour detection (Kochkina et al., 2018a).

Another limitation is that the importance of data reduction tends to be disregarded in the community of rumour detection. Analysing each individual social media post published during breaking news events is not viable due to the rapid speed, vast volume, and noise of data generated by users with ambiguous authorship and uncertain authenticity. Data reduction which automatically filters out less significant and/or invalid data in the initial stages of data analysis is necessary (Sharifi et al., 2013). The majority of existing studies on rumour detection focus on models and do not pay much attention to *settings*. Some studies claim that their models can perform ERD *in real time*, but do not provide evidence. This thesis finds out a gap between system design and the application of models in real-world scenarios. Specifically, it is impossible for a rumour detection model to classify every single social media message posted during a breaking news event. There is no standard approach for selecting social media posts which can be used as input to models in the context of rumour detection. Some studies select posts which have a large number of shares to generate data sets for rumour analysis (Zubiaga et al., 2016b; Ma et al., 2017). However, this approach is not optimum for generating input data for message-level *ERD* as it takes some time for tweets to receive a high number of reactions.

Finally, little work has explored combining advantages of deep learning and those of hand-crafted features. Existing supervised learning-based approaches exploit hand-crafted features which are assumed to be significant throughout entire event evolution. They are straightforward and useful because they can be used to characterise rumours and discriminate between rumours and non-rumours. However, patterns of rumour spreading change over time and this may affect the significance of features (Kwon et al., 2017). Some researchers propose temporal features to address this issue. Others present that the use of hand-crafted features is ineffective and observational. Furthermore, the growing popularity of deep learning in several tasks has motivated researchers in the community of rumour detection to exploit deep learning techniques which are capable of automatically learning meaningful features of rumour contents and evolutionary rumour propagation with little or no feature engineering. However, these features are not easily interpretable and deep learning architectures require vast amounts of training data. As mentioned before, limited labelled data is one of the major limitations of current rumour detection. Therefore, exclusively relying on either feature-based learning or deep learning requires careful consideration.

2.8 SUMMARY

This chapter has described background on rumour detection. Varying definitions of the term "rumour" and research efforts to characterise rumours on social media were introduced in Section 2.1. The task of rumour detection is often formed as a classification problem. A formal definition of message-level rumour detection was detailed in Section 2.2. Related studies on rumour detection were classified according to methodology in Section 2.3.3 and publicly available data sets for rumour detection were introduced in Section 2.4. Based on the information detailed in the previous sections, Section 2.7 discussed several limitations of the current rumour detection. The following chapters will detail the aim and objectives of this thesis as well as methods and experiments conducted to achieve the aim.

3 RESEARCH AIM AND OBJECTIVES

This chapter explains the aim, objectives and hypotheses of the research conducted in this thesis. It describes how different objectives and research methods developed in this thesis fit together to fulfil the ultimate research aim and advance the state of the art.

3.1 LABELLED RUMOUR DATA AUGMENTATION USING SEMANTIC RELATEDNESS

This research topic addresses RQs 1.1-1.3 stated in Section 1.2.

3.1.1 *Objective*

As discussed in Section 2.7, labelled data scarcity and class imbalance hinder achieving the full potential of deep learning techniques in a wide range of tasks including rumour detection. They also lead to the poor generalisation and transferability of rumour detection models to new data and contexts. Increasing training data size has a major impact on achieving new SOTA performance on ERD. However, humans cannot read millions of noisy social media messages to annotate them for the task of rumour detection (Zubiaga et al., 2016b) as it requires deeper domain knowledge and a more elaborate examination than common annotation tasks like image tagging or named entity annotations. To overcome these challenges and advance the SOTA methods for automatic ERD, this thesis aims to augment publicly available rumour data by minimising human supervision.

3.1.2 *Hypothesis*

Rumours spread via the distribution of sources (Maddock et al., 2015; Chen et al., 2018). Sources can quickly evolve into several new variations within the first few minutes. Variations will gradually be increased with more information such as URLs (links) and photos by Twitter users. Messages containing URLs and images are usually created as new messages without attribution. Although the majority of new variations of rumours do not usually have any link or acknowledgement of their sources, they can increase the credibility of sources with low credibility and the likelihood of rumour spreading (Tanaka et al., 2014; Castillo et al., 2011; Gupta and Kumaraguru, 2012; Friggeri et al., 2014).

Similarly, some studies (Maddock et al., 2015; Zhao et al., 2015; Chen et al., 2018) find out that new variations of rumours posted during the early stages of event diffusion are mostly textual variants. For example, Chen et al. (2018) report that 80% of a publicly available rumour tweet corpus consists of duplicated contents on average. Other studies (Liu et al., 2017; Kwon et al., 2017) present that variations of rumours share similar propagation patterns and propose methods for the identification of rumours based on temporal, structural, and linguistic properties of their propagation.

Based on the findings of related research, this thesis hypothesises that enriching existing labelled rumour data with duplicated tweets or corresponding variants is a promising attempt for ERD methods that rely on the structure of rumour propagation on social media. This can help to alleviate labelled data scarcity by increasing training data size and to balance class distributions, and to boost the performance of ML models for rumour-related tasks. Moreover, it is expected that data augmentation based on semantic relatedness can improve one of the limitations of public rumour data (i.e. a high proportion of duplicated tweets) by identifying rumours related to but not exactly identical to labelled rumours.

3.1.3 *Research Design Overview and Evaluation*

Input consists of *"References"* and *"Candidates"* sets. *"References"* are limited ground truth source tweets which are exploited to provide higher-level supervision for unlabelled candidate tweets (i.e. *"Candidates"*). Candidate tweets refer to any tweets that report an event of interest. The pairwise computation of semantic relatedness between embeddings of labelled reference tweets and those of unlabelled candidate tweets is performed. Using optimum thresholds fine-tuned by performing a paraphrase identification task, rumour and non-rumour source tweets are automatically selected. Retweets and replies of selected source tweets are also included in the final output (i.e. augmented data). Details about data sets and methodology are described in Section 4.2.

In order to evaluate the effectiveness of semantically augmented data in rumour detection, rumour detection experiments are carried out using data sets before and after data augmentation (see Section 4.7.2). To show the usefulness of data augmentation for ERD, temporal filtering is applied to data sets. Specifically, source tweets posted before the occurrence date of an event are filtered out, and then only contexts (i.e. replies) posted within the first seven days of the creation of their source tweets are considered. A SOTA deep learning-based model for rumour detection is employed in the experiments. F1-score, precision, and recall are used to evaluate performance.

The research brings performance enhancement to deep learning-based ERD by alleviating the problems of labelled data scarcity and class imbalance in existing rumour data sets without laborious human supervision. The resulting data is publicly available and will help the research community in rumour detection explore deeper NNs for ERD, which are expected to improve generalisability and classification performance.

3.2 IDENTIFICATION OF POTENTIAL RUMOURS VIA TEMPORAL SIGNALS

This research topic addresses RQs 2.1 and 2.2 stated in Section 1.2.

3.2.1 *Objective*

Analysing every social media post published during breaking news events is not viable due to the rapid speed, vast volume, and noise of data generated by users with unattributed authorship and uncertain authenticity. Data reduction which automatically filters out less significant and/or invalid data in the initial stages of data analysis is necessary (Sharifi et al., 2013). This can often be done by producing summaries which offer insights about further data exploration.

This thesis finds out a gap between system design and the application of ERD models in real-world scenarios. In particular, this gap is significant in the case of message-level ERD. It is inefficient and not viable to process every single social media message posted during a breaking news event in real time due to massive amounts. There is no standard approach for selecting social media posts which can be used as input to models in the context of rumour detection.

This thesis aims to partially fill this gap by proposing that a preliminary step for selecting candidates for rumours (i.e. *potential rumours*) without manually examining an enormous number of messages and with minimum time delay from the publication of messages. This is crucial to demonstrate the true value of rumour detection in real-world applications. The term "potential rumours" refers to claims which 1) are the centre of attention, and 2) should be further examined by human experts or analysed by automated rumour detection models to be confirmed as rumours. To this end, a framework involving key burst detection and summarisation is proposed (see Chapter 5).

3.2.2 *Hypothesis*

This thesis studies the following research question: "What are early signals for rumours?" Section 2.1.3 introduced and discussed different studies that aim to characterise rumours and their propagation on social media by exploring different factors such as users' reactions to rumours. Section 2.3.3.1 also reviewed existing supervised learning-based approaches for rumour detection which usually exploit hand-crafted contextual features.

Section 2.5 studies whether contextual features which have been claimed to be useful for rumour detection are actually valid. Based on the findings of related research and Section 2.5, this thesis hypothesises that *temporal signals* are the key to discovering potential rumours in the early stages of rumour evolution. Temporal signals refer to time series patterns (e.g. bursts, peaks, increasing/decreasing trends, etc.) of input data. Other signals, e.g. the popularity of a message (typically the number of retweets and/or replies to the post); user behavioural patterns; and the structure of information diffusion, become available when rumours have already become widespread. In contrast, temporal signals are instantaneous and require no intensive computation.

Existing studies find out that rumours posted during the early stages of event diffusion are mostly simple copies of sources (Maddock et al., 2015; Zhao et al., 2015). This finding indicates that a rumour will get many shares once it starts to get popular. This will eventually result in a high volume in event time series plots. Based on the insight gained from the related work, this thesis proposes that *bursts* identified using temporal signals are key to the early identification of potential rumours. Temporal signals are robust to short-term fluctuations exhibited in event evolution on social media and be portable to unseen events in different domains.

3.2.3 *Research Design Overview and Evaluation*

Input is a time series which consists of posts and their posting times. The first component, *key burst detection*, aims to identify key bursts exclusively based on evolutionary patterns of breaking news events over time. The main reason for using bursts is that they can indicate occurrences of popular sub-events in nature. Once a time window in a time series is identified as a key burst, the second component, *summarisation*, is performed. This task is to extract representative summary posts from a set of posts published during a key burst. Output (i.e. summaries of each key burst in the input time series) is referred to as *potential rumours*.

Most existing methods for burst detection and summarisation are evaluated using data sets related to sports events because ground truth (e.g. match highlights) can easily be obtained via sports news outlets (Hsieh et al., 2012; Meladianos et al., 2018b; Doman et al., 2014; Zubiaga et al., 2012; Nichols et al., 2012). However, due to inherent characteristics of online rumours, it is impossible to clearly identify *highlights* of rumour evolution. In the case of ERD, it is particularly difficult to identify temporal points of interest because "how early" can vary among decision-makers. Therefore, this thesis proposes a new approach to evaluate the proposed models in the context of ERD. This thesis first performs a comparative analysis of different burst detection methods based on patterns of detected bursts and the behaviour of parameters (Section 5.5.1). In Section 5.5.2, several frequency-based and graph-based summarisation methods with different settings are compared in the context of ERD based on weak labels obtained via data augmentation proposed in Chapter 4. Frequency-based methods assign weights to terms based on their frequency in a corpus. Graph-based methods do it based on attributes obtained from a graphical representation of an input corpus. Finally, Section 5.5.3 compares different burst detection methods in terms of ERD.

The research identifies and partially fills a gap between large-scale data collection and analysis in ERD, which has received less attention from existing research. The ultimate goal is to realise end-to-end rumour detection frameworks so that they can be accepted for use in research and practice. The proposed solution based on weak supervision (i.e. temporal signals and heuristic rules) is domain- and task-agnostic. It can be used to select appropriate input data cheaply and efficiently before performing ERD.

3.3 CONTEXT-AWARE EARLY RUMOUR DETECTION

This research topic addresses RQs 3.1 and 3.2 stated in Section 1.2.

3.3.1 *Objective*

This thesis addresses the task of *message-level* ERD. Social media messages, particularly tweets, are short and contain very limited context on their own.

Therefore, the analysis of each social media post in isolation has limited potential to advance SOTA performances on rumour detection. To tackle this issue, this thesis exploits *contextual information* for message-level rumour detection. As applying it to real-world cases is not viable without preliminary data reduction (Zubiaga et al., 2016b), this thesis proposed a preliminary step for identifying potential rumours to overcome this drawback.

This section focuses on message-level detection because current event-level rumour detection methods cannot be compatible with ERD and have limited application for the following reasons. Firstly, the identification of a specific rumour event is a challenging task which requires separate thorough research. Therefore, several studies proposing event-level rumour detection exploit data sets generated using external sources such as websites for rumour debunking and fact-checking. This makes models unable to generalise to newly emerging events as early as possible. Event-level rumour detection should be preceded by preliminary steps such as the automatic identification of specific rumours (Section 2.1.2). Secondly, even if a rumour event is identified, the selection of related posts requires further analysis such as clustering. This task is challenging due to inherent characteristics of social media data such as noise and vast volume. Several studies on event-level rumour detection do not deal with these preliminary tasks in practice. In particular, they do not study how to generate a test corpus for a pre-trained model as soon as an event occurs (i.e. before fact-checking websites identify and verify rumours). Without addressing this issue, the application of event-level detection models to real-life cases is limited.

As for contextual information, it typically refers to conversational threads (replies) of source tweets in the case of Twitter. Most early work on rumour detection dealt with each tweet in isolation as a unit of analysis rather than considering surrounding contexts. Recent findings have shown that the exploitation of contextual information improves the performance of rumour detection (Ma et al., 2017; Kochkina et al., 2018a; Zhou et al., 2019b). Unlike existing studies on context-sensitive rumour detection which focus on textual contents of source tweets and those of their contexts, this thesis aims to study whether a context-aware DNN framework for ERD, which is capable of learning not only rumour content but also social-temporal contexts of rumour diffusion in an unsupervised fashion, can improve SOTA performance.

3.3.2 *Hypothesis*

When humans find it difficult to make sense of a piece of information, it is natural for them to seek coherence in its surroundings (Mitra and Gilbert, 2015). For instance, contexts, in particular, conversational threads of rumour source tweets are used to manually annotate tweets for the analysis of rumours in social media (Zubiaga et al., 2016a).

Exploiting conversational threads of source tweets helps to understand contexts surrounding source tweets. For example, a study on rumour stance classification (Zubiaga et al., 2016c) notes that using similar patterns of users' reactions (e.g. support, deny, and question) displayed across different source tweets can help to improve the performance of stance classification. Although identifying types of replies is beyond the scope of this thesis, leveraging replies instead of exclusively relying on source tweets helps models learn more meaningful representations of rumours.

Rumours tend to proliferate easily through word-of-mouth. People tend to determine the credibility of unverified information based on how others view it (Lee and Oh, 2017). According to the authors, the fact that a piece of information receives a high number of retweets increases the likelihood that people will believe it is true and share it with more people because they will assume that many people have already checked their reliability. As to characteristics of users who participate in rumour spreading, previous studies found that influential Twitter users are less likely to repost and contribute to the diffusion of unverified information (Ma et al., 2017; Kwon et al., 2017).

This thesis therefore hypothesises that the use of contexts is beneficial for a fully automated and effective rumour detection architecture. Contexts can provide the understanding of propagation patterns which are different between rumour and non-rumours, and furthermore, users' reactions to rumours. Consequently, using contexts helps rumour detection models to better understand what distinguishes rumours from non-rumours (Zubiaga et al., 2018b).

3.3.3 *Research Design Overview and Evaluation*

Input consists of a sequence of source tweets and corresponding contexts (i.e. replies). The proposed architecture first learns textual representations and social-temporal propagation features separately. Specifically, textual contents of sources and contexts are embedded into a sequence of vectors. As for social-temporal features of contexts, this thesis exploits hand-crafted features which are simply based on metadata obtained from Twitter's API. They have been widely used to characterise rumours and their evolution for supervised learning-based approaches (see Section 2.1.3 and 2.3.3.1). The extraction of these features does not require any domain-specific feature engineering and intensive computation. Output representations of sentence embeddings and those of social-temporal features are concatenated and used to determine whether each source tweet in the input sequence is a rumour or non-rumour.

This thesis compares different context-sensitive rumour detection models based on F1-score, precision, and recall. Evaluation results show that the exploitation of shallow social-temporal features helps deep learning architectures provide better representations of rumours and their propagation in social media.

This dissertation investigates a novel hybrid and context-aware deep learning model for message-level ERD where limited information is available. The proposed solution leverages the joint learning of textual representations and social-temporal contexts of individual source tweets so that linguistic characteristics of rumours as well as their spreading patterns can be modelled. Extensive experiments on large weakly-labelled data and validation on real-world data demonstrate that its effectiveness in ERD and generalisability to new data.

3.4 summary

This chapter described how different contributions of this thesis fit together and can be integrated into a framework for ERD. Details of methods and experiments will be explained in the following four chapters. The main idea of this thesis is that leveraging weak supervision and contextual data resolves major challenges of current SOTA approaches for ERD, and consequently brings performance improvements. While a large body of SOTA research on rumour detection focuses on improving performance, this thesis sheds new light on preliminary tasks which are often overlooked and researches systematic approaches for an end-to-end model for ERD. By putting such a system into practice, practitioners such as emergency response team and journalists can detect rumour sources in the early stages of event diffusion, thereby taking proactive and remedial actions. A technical and scientific validation of the proposed solutions on real-world data is novel in that it is performed in the context of ERD. Evaluation results demonstrate that the context-aware ERD with weak supervision advances SOTA performance. Ultimately, the findings of this thesis are expected to reinforce the practicability and generalisability of an entire rumour resolution process.

Part III

4 LABELLED RUMOUR DATA AUGMENTATION USING NEURAL LANGUAGE MODELS

4.1 INTRODUCTION

The introduction of this thesis, background on rumour detection as well as the aim and objectives of this thesis were introduced so far. Data sets are a fundamental, vital ingredient in ML tasks. In particular, richer and larger data sets play a key role in improvement in the performance of ML models. Therefore, this thesis starts with confronting challenges in current data sets for rumour detection. This chapter will now address the RQs 1.1-1.3 introduced in Section 1.2. Previous research related to this chapter was introduced in Section 2.3.1.

Research areas that have recently been receiving much attention in ML and NLP include automated rumour and fake news detection (Helmstetter and Paulheim, 2018; Kwon et al., 2017; Shu et al., 2017; Shu et al., 2018; Ma et al., 2018b) and fact-checking (Boididou et al., 2018; Vosoughi et al., 2017; Kochkina et al., 2018a). In particular, deep learning architectures have been increasingly popular by providing significant improvements to SOTA performances. Despite their success, several challenges have yet to be tackled.

One major bottleneck of SOTA ML methods for rumour studies is that they require a vast amount of labelled data to be trained. However, the manual annotation of large-scale and noisy social media data for rumours is highly labour-intensive and time-consuming (Zubiaga et al., 2016b) as it requires deeper domain knowledge and a more elaborate examination than common annotations tasks like image tagging or named entity annotations. Due to limited labelled training data, existing NNs for rumour detection usually have shallow architecture (Chen et al., 2018; Ma et al., 2016). This restricts a further exploration of NNs through many layers of non-linear processing units and different levels of abstractions (Zhong et al., 2016), which results in over-fitting and generalisation concerns. The scarcity of labelled data is a major challenge faced by the research of rumours in social media (Aker et al., 2017).

Another problem is that publicly available data sets for rumour-related tasks such as PHEME data (Kochkina et al., 2018a) suffer from imbalanced class distributions (Liu et al., 2017). Existing methods for handling the class imbalance problem (e.g. oversampling and the use of synthetic data (Xu and Chen, 2015)) may cause overfitting and poor generalisation performance. Methodology for rumour data augmentation with the minimum of human supervision is necessary.

Data augmentation has the potential to be key to learning with DNNs as modern DNNs require a large amount of data for training. The artificial augmentation of training data helps to alleviate data sparsity and class imbalance, reduce overfitting, and improve the generalisation performance of ML models, thereby sustaining deeper networks and improving their performance.

Previous research on online rumours has reported that new variants of rumours in the early stages of event evolution are mostly textual variations (Maddock et al., 2015; Chen et al., 2018; Zhao et al., 2015) and they exhibit similar spreading patterns (Liu et al., 2017; Kwon et al., 2017). Therefore, enriching existing labelled rumour data with duplicated tweets or corresponding variants is a promising attempt for ERD methods that rely on the structure of rumour propagation on social media. This can help to alleviate labelled data scarcity by increasing training data size and to balance class distributions, and to boost the performance of ML models for rumour-related tasks.

Data sets for rumour detection are slightly different from those for other text classification tasks such as polarity and topic classification. It is natural that certain forms of rumours are predominantly popular than others during a real-world event, hence there can be dominant types of rumours in manually annotated benchmark data. Such a bias is realistic and natural in the task of rumour detection, while it causes a serious problem of biasing a model towards specific data instances. To alleviate this, my data augmentation employs semantic relatedness as weak supervision rather than relying on static synonyms. In addition, using contextualised sentence embeddings allows identifying rumours which are not exactly identical to manually annotated rumours but have contextual meaning similar to them.

This chapter is based on my publications in the proceedings of 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (Han et al., 2019b) and at the workshop on Learning from Limited Labelled Data (ICLR 2019) at the 7th International Conference on Learning Representations (Han et al., 2019a). This chapter proposes a novel data augmentation method based on semantic relatedness in order to improve current ERD. The method is based on a publicly available paraphrase identification corpus, context-sensitive embeddings of labelled reference tweets and unlabelled candidate source tweets. Pairwise similarity is used to guide the assignment of pseudo-labels to unlabelled tweets. ELMo (Peters et al., 2018), a SOTA context-sensitive NLM, is fine-tuned on a large credibility-focused social media corpus and used to encode tweets. ELMo is introduced in Section 2.6.1. Results show that data augmentation can contribute to rumour detection via deep learning with increased training data size and a reasonable level of quality. This has the potential for further performance improvements using deeper NNs. Data augmentation results for six real-world events and the performance of a SOTA DNN model for rumour detection with augmented data are presented. The augmented data sets have been made publicly available for further research purposes.

4.2 DATA

The goal of this chapter is to augment manually annotated data with tweets annotated using their semantic similarity with manually labelled rumour stories. To achieve the aim, this section introduces the data sets used for the *two source tasks* (i.e. semantic relatedness threshold fine-tuning and NLM fine-tuning) and the *target task* (i.e. rumour data augmentation): 6 publicly available data sets including a Twitter paraphrase corpus, two large-scale

Twitter corpora, and three data sets covering a wide range of real-world events on social media are used. This section also describes an overview of the proposed methodology.

In the first source task, two semantic similarity thresholds for annotating rumour and non-rumour tweets respectively should be determined. To this end, an experimental study of semantic equivalence between tweet pairs is conducted using a Twitter corpus built for such a task. In the second source task, a Twitter corpus with associated credibility ratings are employed to fine-tune a pre-trained NLM specifically for the task of rumour detection. To show the effectiveness of a task-specific NLM, a general-purpose Twitter corpus is also used for NLM fine-tuning. In the target task, the input corpus of the proposed data augmentation method consists of *References* and *Candidates* sets. *References* are limited ground truth source tweets which are exploited to provide higher-level supervision for unlabelled candidate tweets (i.e. *Candidates*). *Candidates* refer to any tweets that report an event of interest.

4.2.1 *Data for Semantic Relatedness Fine-Tuning*

- **SemEval-2015 task 1 data (Xu et al., 2014)** This data is built for paraphrase identification and semantic similarity measurement. It is employed to determine semantic relatedness thresholds for selecting rumour and non-rumour source tweets given embeddings of labelled references and those of unlabelled candidate tweets. It consists of tweet pairs and binary labels indicating whether two sentences in each pair express the same or very similar meaning or not. In more specific, it consists of training (13, 063 sentence pairs), development (4, 727 pairs), and test (972 pairs) sets. Each data set contains three types of labels: "paraphrase", "non-paraphrase", and "debatable". This chapter excludes sentence pairs with the "debatable" label. In addition, this chapter concerns with identifying semantic similarity scores rather than developing a ML model, which typically requires training, validation, and testing, for identifying paraphrases. Therefore, the three types of data sets (i.e. training, development, and test) were merged. Consequently, a single data set comprising of 16, 510 sentence pairs with binary labels (i.e. paraphrase and non-paraphrase) is used for the proposed task. All sentences were lower-cased and tokenised.

4.2.2 *Data for LM Fine-Tuning*

- **CREDBANK (Mitra and Gilbert, 2015)** This data comprises more than 80 million tweets grouped into 1, 049 real-world events, each of which was manually annotated with credibility ratings. This large corpus is leveraged to fine-tune ELMo model in order to provide meaningful representation of tweets for rumour-related tasks.

- **Twitter7 (Yang and Leskovec, 2011)** The SNAP Stanford Twitter data set *Twitter7* is used as a general-purpose Twitter corpus in data augmentation. This is a collection of 476 million tweets collected between June-Dec 2009. The authors estimate that it contains 20-30% of all public tweets published during the time period. This data is used to conduct a comparative analysis of the effectiveness of *CREDBANK* as a rumour task-specific dataset for

LM training. It was downloaded from https://snap.stanford.edu/data/twitter7.html (accessed on March 2019).

Table 4.1: Statistics of two corpora for fine-tuning ELMo.

Corpus	Item	Training	Hold-out
CREDBANK	tweets	6, 155, 948	1, 232
	tokens	146, 313, 349	27, 298
	vocabs	2, 234, 861	6, 517
SNAP	tweets	13, 928, 924	6, 000
	tokens	193, 192, 322	99, 758
	vocabs	11, 696, 602	24, 585

For the data augmentation task, two types of data sets are generated. One is a task-specific corpus generated using the *CREDBANK* and the other is a general-purpose corpus generated using the *SNAP Twitter7*. Table 4.1 shows the number of tweets, tokens, and vocabularies in the training and hold-out sets of the *CREDBANK* and *SNAP Twitter7* after language filtering (i.e. considering only English tweets) and deduplication (i.e. removing duplicated tweets). For the *CREDBANK*, sentences in the original corpus are shuffled and split into training and hold-out sets. About 0.02% of the entire corpus is used as the hold-out set. As for the *SNAP Twitter7*, "June" tweets are used as a training set to fine-tune the pre-trained ELMo model. 6,000 tweets are sampled from "November" tweets and used as a hold-out set. A test set is built using the *PHEME* (Kochkina et al., 2018a) containing 6, 162 tweets related to 9 events in the hope that it will offer an independent and robust evaluation.

4.2.3 *Data for Rumour Data Augmentation*

4.2.3.1 *Reference data*

The data sets listed in this section are used to generate references for rumour and non-rumour source tweets for data augmentation.

"Germanwings plane crash", "Sydney siege", "Ferguson unrest", "Ottawa shooting", and "Charlie Hebdo shooting"

- **PHEME (6392078; Kochkina et al. (2018a))** This data contains manually labelled rumour and non-rumour source tweets and their replies for 9 events. Refer to Section 2.4.1 for the details. The five breaking news events which potentially produce several rumours will be augmented.
- **CrisisLexT26 (Olteanu et al., 2015)** This data set comprises tweets associated with 26 hazardous events happened between 2012 and 2013. A subset of data is manually labelled based on informativeness, information types, and information sources. It should be noted that this data might not contain rumours. How references are generated using this data is detailed in Section 4.2.4.

4.2.4 *Reference Generation*

Some examples of references for the six events are shown in Appendix a.1. This section details how references are generated using publicly available labelled data.

Example 4.1:

(1) "CORRECTION: We reported earlier Sydney air space was shut down. That is not correct. No Sydney air space has been shut down. #SydneySiege"

(2) "DEVELOPING: Airspace shutdown over Sydney amid chocolate shop hostage situation; Islamic flag shown in shop's window."

For the *PHEME5*, annotated rumour categories in the *PHEME (6392078; Kochkina et al. (2018a))* are used. In specific, rumour source tweets were categorised by their topics and the authors created clean texts for each rumour category. For example, a rumour category for the Sydney siege event, "The Sydney airspace has been closed", includes several rumour source tweets related to airspace over Sydney. Example 4.1 shows some examples.

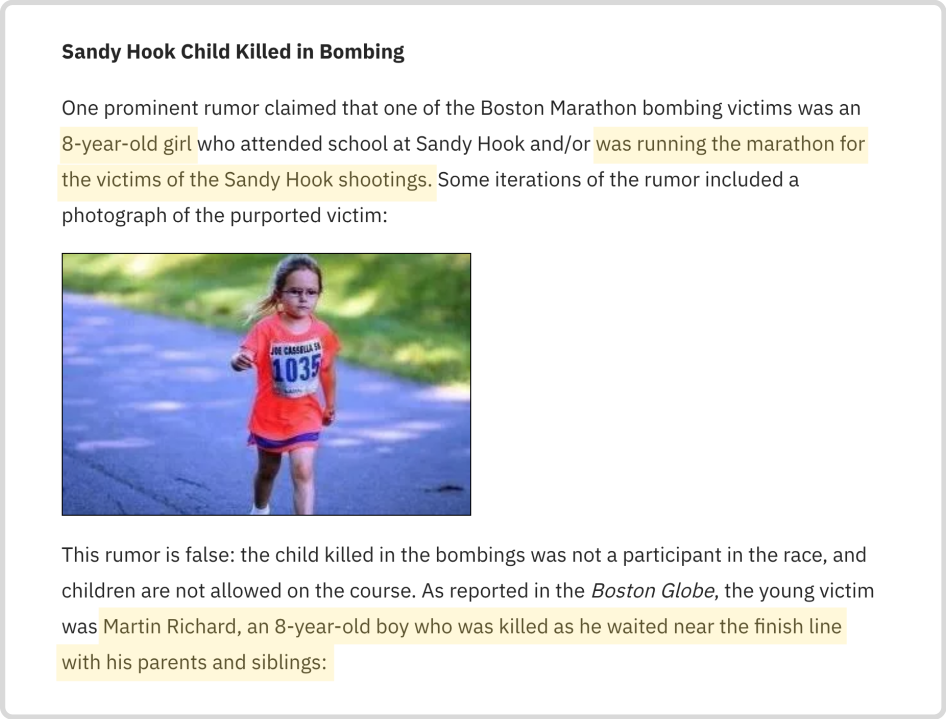
Using raw tweets as references may help to capture more various patterns of rumour variations. However, tweets are very noisy and contain a large amount of non-standard spelling. To ensure high-quality references and reduce the computation time of pairwise similarity between candidates and references, clean texts, e.g. "The Sydney airspace has been closed." in the example above rather than tweets in Example 4.1, are used as rumour references.

As the "bostonbombings" event is not available in the *PHEME (6392078)*, the Boston Marathon bombings rumour archive created by Snopes.com[1] and *CrisisLexT26* are used to build references.

Firstly, any rumours investigated by Snopes.com are included in the reference set for "bostonbombings" regardless of their veracity. Figure 4.1 shows an example of rumours listed in the archive. From this example, several references can be generated such as "8-year-old girl died", "8-year-old boy killed has been identified as Martin Richard", "8-year-old girl is not one of the dead, a 8-year-old boy is dead."

Secondly, in the *CrisisLexT26*, tweets are categorised by their *informativeness* (i.e. related to the crisis and informative; related but not informative; and unrelated), *information type* (i.e. affected individuals; affected infrastructure; donations & volunteers; caution & advice; emotions; and other useful information), and *information sources* (e.g. eyewitness and media). The original data includes 1,000 annotated tweets for the Boston marathon bombings. As the *CrisisLexT26* is not annotated under an annotation scheme for social media rumours, labels should be mapped to binary labels (i.e. rumours/non-rumours). To this end, tweets annotated as "related and informative" (*informativeness* category) are first selected. Among related and informative tweets, tweets labelled as "affected individuals", "infrastructure and utilities", and "other useful information" (*information type* category) are selected. After selecting tweets based on annotations, 335 annotated tweets remain. They are manually inspected and categorised into rumours and non-rumours according to a rumour tweet annotation scheme proposed by (Procter et al., 2013). Specifically,

[1] available via https://www.snopes.com/fact-check/boston-marathon-bombing-rumors/



Sandy Hook Child Killed in Bombing

One prominent rumor claimed that one of the Boston Marathon bombing victims was an 8-year-old girl who attended school at Sandy Hook and/or was running the marathon for the victims of the Sandy Hook shootings. Some iterations of the rumor included a photograph of the purported victim:

This rumor is false: the child killed in the bombings was not a participant in the race, and children are not allowed on the course. As reported in the *Boston Globe*, the young victim was Martin Richard, an 8-year-old boy who was killed as he waited near the finish line with his parents and siblings:

Figure 4.1: An example of rumours listed in the Boston Marathon bombings rumour archived built by Snopes.com.

it suggests that the "rumour" label is assigned to tweets in which authors emphasise they "heard" something, but lacks evidence (e.g. a URL); those rebutting claims made by other people; those appealing for more details; and those expressing reactions to rumours. Example 4.2 shows some example tweets which remain after filtering according to the *CrisisLexT26*'s annotations and are coded as "rumour" under the Procter's scheme. User mentions are replaced with "[username]" to ensure the confidentiality of personally identifiable information. To match the format of references generated using the *PHEME (6392078)*, tweets are preprocessed as described in Section 4.3, e.g. removing retweet symbols, hashtags, and URLs. For instance, from the third example tweet in Example 4.2, a reference "Despite reports to the contrary there has not been an arrest in the Marathon attack." is obtained.

Example 4.2:

(1) "RT @[username]: ABC News is reporting that air quality experts have been brought into Boston to ensure explosions were not a chemical attack."

(2) "RT @[username]: Doctors: bombs contained pellets, shrapnel and nails that hit victims #BostonMarathon"

(3) "RT @[username]: Despite reports to the contrary there has not been an arrest in the Marathon attack."

4.2.4.1 *Candidate Data*

The data set listed in this section is used to generate candidates for rumour and non-rumour source tweets.

- **Twitter event datasets (2012-2016; Zubiaga (2018))** This data consists of over 147 million tweets associated with 30 real-world events unfolded between February 2012 and May 2016. Tweets were collected using Twitter's streaming API with predefined keywords and hashtags shown in Table 4.2. This data is used as a pool of candidate source tweets. Among the 30 events, six events for which references can be generated using publicly available data sets and resources are selected: "Ferguson unrest", "Sydney siege", "Ottawa shooting", "Charlie Hebdo attacks", "Germanwings plane crash", and "Boston marathon bombings". Five events except for the "Boston marathon bombings" are separately referred to as ***PHEME5*** since their references can be generated from the *PHEME (6392078)*. As for the "Boston marathon bombings" event, a reference set is built from *CrisisLexT26* and publicly identified rumour sources from fact-checking website Snopes.com [2] since it is not available from *PHEME (6392078)*.

Table 4.2: Keywords used by Zubiaga (2018) and time periods to collect the original Twitter event data sets (2012-2016).

Event	Year	Start	End	Keywords
germanwings	2015	24 March	30 March	'germanwings', 'a320', '4u9525', 'absturz', 'flughafen duesseldorf'
sydneysiege	2014	14 December	17 December	'#sydneysiege', 'sydney', 'gunman', 'lindt', 'martin place'
fergusonunrest	2014	9 August	26 August	'#ferguson'
ottawashooting	2014	22 October	24 October	'ottawa', 'shooting', '#ottawashooting'
bostonbombings	2013	15 April	16 April	'boston', 'marathon', '#prayforboston'
charlihebdo	2015	7 January	14 January	'#charliehebdo', '#jesuischarlie', 'charlie hebdo', 'paris'

4.2.5 *Data Collection*

The tweets in the *PHEME (6392078)*[3] and the *CrisisLexT26*[4] are publicly available. Source tweets for six selected events in the *Twitter events 2012-2016* and *CREDBANK* are downloaded using an open source tweet collector called *Hydrator* [5]. Table 4.3 shows the number of tweet ids in the original *Twitter events 2012-2016* data, that of downloaded tweets, that of candidate source tweets which remained after language-based filtering (i.e. considering only English tweets) and linguistic preprocessing (see Section 4.3), and that of references. For the *CREDBANK*, 77, 954, 446 out of 80, 277, 783 tweets (i.e. 97.1% of the original data) are downloaded. After deduplication, the train

[2] A collection of rumours source tweets associated with the Boston Marathon bombing are available via https://www.snopes.com/fact-check/boston-marathon-bombing-rumors/, last access in April, 2019

[3] available via https://figshare.com/articles/PHEME_dataset_for_Rumour_Detection_and_Veracity_Classification/6392078

[4] available via https://crisislex.org/data-collections.html#CrisisLexT26

[5] available via http://github.com/DocNow/hydrator

corpus contains 6,157,180 tweets with 146,340,647 tokens and 2,235,075 vocabularies. Retweets are collected using a Python library *tweepy* [6].

As to replies, Twitter does not provide an API for retrieving them. This chapter collects replies following the practice introduced in existing research on the veracity of social media information supported by the European Union's Seventh Framework Programme (FP7) under the PHEME project [7] (Zubiaga et al., 2016a). Tweet IDs of replies are collected via a HTML parsing technique implemented using Python libraries *Selenium* [8] and *BeautifulSoup* [9]. For each source tweet id and its author's *screen name*, a Python script visits the source tweet's page (i.e. https://twitter.com/[screen name]/status/[source tweet id]). The script then parses the page and returns only tweet IDs of tweets replying to the source tweet. To collect Twitter objects for those replies, Twitter API is used; specifically, the 'statuses/lookup'[10] endpoint providing full Twitter objects for up to 100 tweets per request.

A "screen name" in Twitter is an alias that a user identifies themselves with.

Table 4.3: Number of tweet ids provided in the original data, downloaded tweets, tweets after preprocessing, and reference tweets for Twitter event data sets (2012-2016).

Event	Original tweets	Downloaded tweets	After preprocessing	# of references
bostonbombings	3,430,387	1,886,632	1,259,857	88
charlihebdo	1,894,0619	12,253,734	4,276,112	60
fergusonunrest	8,782,071	5,743,959	5,504,692	41
germanwings	2,648,983	1,726,981	702,864	19
ottawashooting	1,075,864	737,136	669,734	51
sydneysiege	2,157,879	1,376,218	1,211,295	61

4.3 METHODOLOGY OVERVIEW

An overview of data augmentation method is presented in Figure 4.2. The input consists of *"References"* and *"Candidates"* sets. *"References"* are limited ground truth source tweets which are exploited to provide higher-level supervision for unlabelled candidate tweets (i.e. *"Candidates"*). Candidate tweets refer to any tweets that report an event of interest. Schemes for constructing references vary between data sets. For the *PHEME5*, manually labelled rumours in the *PHEME (6392078)* are used. References for the "Boston marathon bombings" event are generated separately. Detailed reference generation procedure is described in Section 4.2.4. The red box in the centre shows that a deep BiLM is first trained with domain-specific corpora in order to learn the representation of rumours. The ELMo BiLM model (Peters et al., 2018) is used in the proposed method.

[6] available via https://www.tweepy.org/

[7] The PHEME project's official website is https://www.pheme.eu/

[8] available via http://selenium-python.readthedocs.io/

[9] available via http://www.crummy.com/software/BeautifulSoup/bs4/doc/

[10] https://developer.twitter.com/en/docs/tweets/post-and-engage/api-reference/get-statuses-lookup

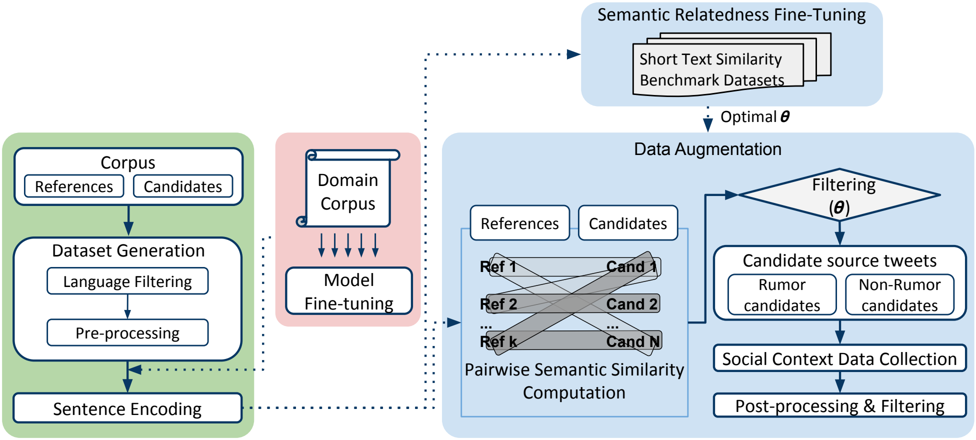


Figure 4.2: Overview of the proposed methodology for data augmentation.

The leftmost (green) box illustrates data preprocessing and sentence encoding. Given a corpus that contains pairs of a reference and candidate tweets, language-based filtering and linguistic preprocessing are performed. The preprocessing includes lower-casing, the removal of retweet symbols ("rt @"), user mentions ("@[username]"), URLs, and non-alphabetic characters, and tokenisation. Embedded links can provide useful information regarding the virality and trustworthiness of tweets (Tanaka et al., 2014; Gupta and Kumaraguru, 2012; Castillo et al., 2011; Friggeri et al., 2014), but they tend not to appear at the very early stages of event diffusion (e.g. during the first burst; Maddock et al. (2015)). Rather, people are more likely to repost what they observed without additional information in the early stages. Relevant findings were described in Section 3.1.2. To recap, the proposed solution is based on a hypothesis that exploiting textual variants of hand-labelled data as a supervision signal is an efficient and promising attempt for augmenting data sets for ERD. Tweets with a minimum of 4 tokens are considered because tweets which lack enough textual features are generally unremarkable and add noise to data (Ifrim et al., 2014). However, it would be worth studying whether considering very short tweets (i.e. tweets with less than 4 tokens) can improve data augmentation results in future work. Next, contextual embeddings of tweets are computed using the fine-tuned BiLM model.

The blue boxes on the right side illustrate a semantic relatedness-based method for the identification of rumour variants. Cosine similarity between embeddings of references and those of unlabelled candidate tweets is used as a measurement of semantic similarity. Cosine similarity between vector representations of two sentences is a common metric for measuring semantic similarity (Lu et al., 2006; Vijayaraghavan et al., 2016; Shin et al., 2018b). Two semantically equivalent embeddings have a cosine similarity of 1 and two vectors with no relation have that of 0.

To fine-tune relatedness thresholds that determine whether a reference-candidate pair bears strong semantic relation, a standard short text similarity benchmark data set (*SemEval-2015 task 1 data*) is used. Two thresholds learnt from the fine-tuning process include a rumour candidate threshold (θ_1) and non-rumour candidate threshold (θ_2).

Having optimum thresholds, the pairwise semantic similarity of reference-candidate pairs from the *References* and *Candidates* sets is computed. The next step is to select rumours and non-rumours from candidate tweets based on the optimum relatedness thresholds. In the final step, data collection is performed to retrieve social-temporal context data (typically retweets and replies) for selected candidate tweets. Source tweets without replies are filtered out.

4.4 FINE-TUNING ELMO FOR THE TASK OF RUMOUR DETECTION

This section demonstrates how to fine-tune an SOTA ELMo with domain-specific corpora and evaluates whether the fine-tuned model can provide more meaningful and effective representations of tweets related to rumours. ELMo is used in this thesis because it can model contextual meaning of words unlike previous embedding models and there had been no publicly available ELMo fine-tuned on a large-scale, credibility-oriented social media corpus when this research was designed and conducted. ELMo is described in Section 2.6.1. This section addresses the RQ 1.2.

RQ 1.2: Does fine-tuning a SOTA Neural Language Model (NLM) using a domain-specific corpus improve representations of rumours?

4.4.1 *Methodology for Fine-Tuning ELMo*

Previous research shows that fine-tuning NLMs with domain-specific data allows them to learn more meaningful word representation and provides a performance gain (Kim, 2014; Peters et al., 2018).

While ELMo has shown its effectiveness in several downstream tasks (Jiang et al., 2019; Zeng and Gifford, 2019; El Boukkouri et al., 2019; Perone et al., 2018), little work (Veyseh et al., 2019b) has explored the impact of the ELMo on rumour detection.

Fine-tuning a publicly available pre-trained ELMo with a new corpus involves the following stages described on https://github.com/allenai/bilm-tf: 1) input data and a vocabulary preparation, 2) training a pre-trained ELMo, 3) testing on a hold-out set, and 4) saving newly trained ELMo weights. The rest of this section details each stage of ELMo fine-tuning.

Firstly, all sentences in an input corpus are tokenised with whitespace delimiters. Each sentence is padded with special tokens <S> and </S> denoting the start and end of a sentence. The processed input data is randomly shuffled and split into training and hold-out sets. The hold-out set is used to evaluate the fine-tuned ELMo. A vocabulary comprising of unique tokens (i.e. words) in the training data is built.

Secondly, a pre-trained model is retrained on the new data using the same hyperparameters used for pre-training and the new vocabulary. All hyperparameters of the pre-trained model were stored in checkpoint [11] files. The pre-trained bidirectional LSTM-based LM takes input text represented using character embeddings. Due to the statefulness of LSTM, its internal states (i.e. memory) are carried over from batch to batch, and thus, the LM can be contextualised. The model is trained with a window of 20 tokens. For data containing less than 10 million tokens, it is recommended to fine-tune for a small number of epochs and monitor model performance on the hold-out set to avoid overfitting.

Thirdly, a fine-tuned model is evaluated on the hold-out set. "Perplexity" is a popular metric for evaluating LMs and measures how well a LM predicts

[11] https://www.tensorflow.org/guide/checkpoint

an unseen test set. Given a test set denoted by $\mathcal{D} = \{w_1, w_2, \cdots, w_N\}$, where w_i $(\forall i)$ denotes each token, perplexity $PP(W)$ is the inverse probability of the test set. Formally,

$$PP(W) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i|w_1 \cdots w_{i-1})}}$$

The lower perplexity is, the better the model predicts an unseen test set.

Finally, all hyperparameters and weights of the fine-tuned ELMo are saved and used in different downstream tasks. Specifically, ELMo representations of input text are computed using character embeddings.

To fine-tune a pre-trained ELMo, the model and pre-training checkpoints are obtained from the Tensorflow implementation of ELMo[12]. This chapter employs a model trained on "1 Billion Word Language Model Benchmark" corpus (Chelba et al., 2013) with the vocabulary of 793,471 tokens. The fine-tuned model on test data built for rumour detection with *perplexity* to investigate the impact of the fine-tuned ELMo.

Since the *CREDBANK* training set is still relatively small for NLMs, the pre-trained ELMo is fine-tuned for one epoch to avoid overfitting as suggested on https://github.com/allenai/bilm-tf. The BiLM weights of the fine-tuned model are fixed and used for computing sentence embeddings of tweets in the experiments.

The model fine-tuned on the *CREDBANK* was trained for more than 800 hours on the Intel E5-2630-v3 CPU. The model fine-tuned on the *SNAP Twitter7* was trained for more than 500 hours on a NVIDIA Kepler K40M GPU.

4.4.2 *Results of Fine-Tuned ELMo Performance*

Table 4.4 shows great improvements in perplexity on both hold-out and test sets with the ELMo fine-tuned on CREDBANK (i.e. *ELMo+CREDBANK*) in comparison to the ELMo fine-tuned on Twitter7 (i.e. *ELMo+Twitter7*). Reported values are the average of forward and backward perplexity. In particular, it is worth noting that the model fine-tuned on a task-specific corpus shows significant performance gains. Specifically, the *ELMo+CREDBANK* reduces perplexity by 443.04 on the task-specific test set (i.e. *PHEME*), while the *ELMo+Twitter7* merely reduces it by 163.42. The biLM weights of the *ELMo+CREDBANK* are fixed and used for computing ELMo embeddings of tweets in data augmentation and experiments on ERD.

4.5 semantic relatedness threshold selection

For semantic relatedness-based rumour data augmentation, two thresholds for selecting rumour and non-rumour source tweets should be set. To this end, this section adopts the task of paraphrase identification as a benchmark task. Cosine similarity between pairs of two sentence embeddings is used as a measurement of semantic relatedness. It is a common metric for measuring

[12] available via https://github.com/allenai/bilm-tf.

Table 4.4: Improvements in perplexity after fine-tuning a pre-trained ELMo on two different corpora.

Data	Before fine-tuning ELMo	After fine-tuning ELMo on CREDBANK	After fine-tuning ELMo on Twitter7
Hold-out (CREDBANK)	883.06	18.24	389.14
Hold-out (Twitter7)	476.42	N/A	68.22
Test (PHEME)	475.06	32.02	311.64

semantic similarity (Lu et al., 2006; Vijayaraghavan et al., 2016; Shin et al., 2018b).

In the first part of this section, several embedding models are compared to examine whether the fine-tuned ELMo described in the previous section provides more effective tweet representations in the benchmark task. Subsequently, semantic relatedness thresholds for rumour data augmentation are determined.

4.5.1 *Experiments on Embedding Model Selection*

In order to show the effectiveness of the fine-tuned ELMo and select an embedding model for encoding tweets for data augmentation, different word embedding models are compared based on their performance in the task of paraphrase identification. As for data, the *SemEval-2015 data*, which consists of pairs of tweets with binary labels indicating whether two tweets in a pair imply the same meaning or not, is employed (see Section 4.2.1). As for baseline models, two models which have been widely used to encode tweets in the task of rumours detection, i.e. word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014), and a pre-trained ELMo model, i.e. ELMo Original 5.5B, are used. Sentence embeddings are computed using a word embedding model for all the 16, 510 pairs. Cosine similarity between two sentence embeddings of each pair is computed. F1-score, precision, and recall are computed for different threshold values.

- **Glove (Pennington et al., 2014)**: It is a regression model trained on a word-word co-occurrence matrix using an input corpus with a weighted least-squares objective. For the experiments, pre-trained word vectors trained on 2B tweets consisting of 27B tokens and 1.2M vocabularies are obtained from https://nlp.stanford.edu/projects/glove/. This model includes 200-dimensional vectors.

- **Word2Vec (Mikolov et al., 2013)**: It first builds a vocabulary consisting of tokens in an unlabelled training corpus. It then learns high-dimensional vector representations of words and captures semantic relations between words. For the experiments, pre-trained word vectors trained on Google News data consisting of 100M words are obtained

Table 4.5: Comparison of the paraphrase identification performance of different models for sentence representation.

Model	F-measure	Precision	Recall	Threshold
ELMo+CREDBANK (average)	**0.6507**	**0.6088**	0.6986	0.6526
ELMo+CREDBANK (top)	0.6270	0.5660	0.7027	0.6470
ELMo Original 5.5B (average)	0.6281	0.5872	0.6752	0.6305
ELMo Original 5.5B (top)	0.6047	0.5554	0.6635	0.6875
GloVe (twitter.27B.200d)	0.5079	0.3417	**0.9890**	0.5017
Word2Vec (Google News)	0.4223	0.4796	0.3772	0.5003
ELMo Original 5.5B (top)*	0.5868	0.5112	0.6887	0.6752
GloVe (twitter.27B.200d)*	0.5117	0.3565	0.9062	0.5070
Word2Vec (Google News)*	0.4715	0.4473	0.4985	0.5000

*Models are applied to normalised tweets.

from https://code.google.com/p/word2vec/. This model contains 300-dimensional vectors for 3M words and phrases.

- **ELMo Original 5.5B (Peters et al., 2018)**: This pre-trained ELMo model was trained on a data set of 5.5B tokens obtained from Wikipedia[13] (1.9B) and all of the monolingual online news articles from WMT 2008 – 2012[14] (3.6B). The weights and hyperparameters are obtained from https://allennlp.org/elmo.

For the Glove and Word2Vec, each sentence is represented as an average of the embeddings of its constituent words. For tokens that are not found in the vocabularies of the pre-trained models (i.e. out-of-vocabulary problem), embedding vectors are zero vectors. For the ELMo models (i.e. ELMo Original 5.5B and fine-tuned ELMo), the number of LSTM layers in the BiLM (i.e. L) is set to 2. As the first layer is a token layer, three vectors in total are returned for each word. Each layer outputs a 1024-dimensional embedding vector. There are three ways to output ELMo embeddings: "**all** (i.e. all three vectors)", "**top** (i.e. the output of the last LSTM layer)", and "**average** (i.e. the average of the all the three vectors)". As a single vector representation for each tweet is needed, "average" and "top" representations are used in the experiments. Results are presented in Section 4.5.1.1.

4.5.1.1 *Results of Embedding Model Selection*

This section presents the results of comparisons of different word embedding models for the task of paraphrase identification. Table 4.5 compares different word embedding models for word representation on the paraphrase identification corpus, i.e. *SemEval-2015 data*. The reported evaluation results are based on the maximum F-score each model achieves. The experimental results show the effectiveness of the *ELMo+CREDBANK* over a publicly available, pre-trained model (i.e. "ELMo Original 5.5B") and SOTA word embedding

[13] https://en.wikipedia.org/wiki/Main_Page
[14] availablehttps://www.statmt.org/wmt18/translation-task.html

models. Different models are also applied to normalised texts. Normalisation methods used in the experiments include removing English stopwords and punctuations, and lemmatisation using "WordNetLemmatizer" in a Python library NLTK [15]. As shown in Table 4.5, text normalisation degenerates the performance of the ELMo in terms of F-score, while it improves the performance of the other word embeddings. In fact, SOTA NLMs like ELMo do not need much text normalisation[16]. A pre-trained ELMo model only needs tokenisation. As for the output of ELMo models, using the average of representations from all layers outperforms using only the top layer representation. This finding is consistent with results presented in (Perone et al., 2018). The best-performing model, i.e. *ELMo+CREDBANK*, will be employed to obtain sentence embeddings in further experiments on semantic relatedness threshold selection and data augmentation.

4.5.2 *Domain-Specific Semantic Relatedness Threshold Selection*

In the case of rumour data augmentation, higher precision is required to ensure the higher quality of resulting data (i.e. less false positives in selected samples; Vijayaraghavan et al. (2016) and Resnick et al. (2014)). Although the ideal retrieval of posts related to rumours is required to achieve both high precision and high recall, most retrieval methods aim to achieve high precision. Typically, high precision (100% or close to it) is achieved when retrieval results mostly contain relevant posts and high recall (100% or close to it) is achieved when almost all relevant posts are retrieved. Therefore, relatedness thresholds should be fine-tuned based on precision achieved by the best-performing model.

However, the size of a resulting data set is also important (Zubiaga et al., 2016b). For the generation of data sets for rumour-related tasks, Zubiaga et al. (2016b) chose source tweets, the number of retweets of which is above 100, and let journalists manually annotate them. The authors report that the threshold (i.e. 100 retweets) is selected based on the size of the final data set, however, they do not suggest the standard of data size. In fact, no publicly available data for rumour studies (see Section 2.4) specifies a rule for data size. Therefore, a heuristic approach for semantic relatedness selection is employed in the experiment. In specific, augmentation results obtained using different values for the two thresholds are examined. Results are described in Section 4.5.2.1.

This chapter claims that augmented data generated using the proposed heuristic can improve the performance of SOTA deep learning-based ERD model based on experiments in Section 4.7.2.

In the experiment described in Section 4.5.1, the best-performing embedding model in terms of F1-score is used to encode all the tweets in the *SemEval-2015 task 1 data*. The cosine similarity of sentence pairs is computed. Precision, recall, and F1-score are computed for different similarity thresholds. Due to a trade-off between high precision and data size, optimal values for the thresholds θ_1 and θ_2 for data augmentation cannot be formally defined (see discussions in Section 4.5.2). To address this issue, a few thresholds

[15] available via https://www.nltk.org
[16] https://github.com/allenai/bilm-tf

Table 4.6: The performance of the *ELMo+CREDBANK* on paraphrase identification for different thresholds.

F1-score	Precision	Recall	Threshold
0.5093	0.3417	1.0000	0.248
0.5093	0.3417	1.0000	0.266
0.5094	0.3417	1.0000	0.283
0.6507	0.6088	0.6986	0.653
0.6176	0.7000	0.5526	0.691
0.5907	0.7500	0.4871	0.708
0.4421	0.8502	0.2987	0.760
0.2832	0.9003	0.1681	0.802
0.1961	0.9104	0.1099	0.831
0.1731	0.9214	0.0956	0.840
0.1595	0.9301	0.0872	0.846
0.1340	0.9400	0.0722	0.856
0.1240	0.9517	0.0663	0.862
0.0255	0.9865	0.0129	0.920

resulting in various precision values are selected and applied to pairs of references and candidates used for data augmentation. The results are examined and the final θ_1 and θ_2 are chosen by considering the trade-off. Although this approach may not provide optimum results, it is extensible and fulfil the aims stated in Section 3.1. Furthermore, this chapter will investigate its effectiveness in improving deep learning-based ERD in the experiments (see Section 4.7.1).

4.5.2.1 *Results of Semantic Relatedness Threshold Selection*

The *ELMo+CREDBANK* is used to encode sentence pairs in the *SemEval-2015 task 1 data* and cosine similarity betweem two sentences in each pair is computed. Table 4.6 compares a few selected F1-score, precision, and recall computed for different semantic similarity thresholds. A F1-score of 0.5093 and precision of 0.3417 are the lowest performance achieved by the model.

In order to choose two thresholds θ_1 and θ_2 for augmenting rumour and non-rumour source tweets, the data augmentation procedure described in Section 4.6.1 is adopted. The results are shown in Table 4.7. It shows the number of rumour and non-rumour source tweets obtained using different thresholds for four events. "Pr.", "R", and "NR" are abbreviations for "precision", "rumour", and "non-rumour".

To determine a threshold θ_1 for augmenting rumour source tweets, the other threshold θ_2 is fixed to 0.266, which achieves the lowest performance on the benchmark task (i.e. the task of paraphrase identification). Figure 4.3 visualises the number of augmented rumour source tweets for different values for θ_1. The most significant increase in the number of rumours is observed when $\theta_1 = 0.802$, which achieves a precision of 0.9003. This phenomenon is particularly significant for the event "germanwings", which has the lowest number of augmented source tweets across different thresholds. For the

Table 4.7: The number of rumour and non-rumour source tweets augmented using different thresholds and the method described in Section 4.6.1 for different events. (Pr.: precision, R: rumour, NR: non-rumour).

	Threshold		bostonbombings		germanwings		ottawashooting		sydneysiege	
Pr.	R	NR	R	NR	R	NR	R	NR	R	NR
0.9865	0.920	0.266	1,058	3,173	1	3	182	546	753	2,238
0.9517	0.862	0.266	2,196	6,588	558	1,674	3,282	9,799	2,961	8,744
0.9400	0.856	0.266	2,285	6,854	575	1,725	4,642	13,674	3,896	11,516
0.9301	0.846	0.266	2,725	8,171	743	2,229	7,809	22,586	6,088	18,010
0.9214	0.840	0.266	2,896	8,684	794	2,382	9,882	28,296	7,039	20,811
0.9104	0.831	0.266	3,338	10,008	988	2,964	11,210	31,979	8,858	26,183
0.9003	0.802	0.266	8,533	25,569	5,419	4,454	21,227	59,016	21,936	54,165
0.8502	0.760	0.266	17,625	52,707	17,459	4,454	44,907	82,232	66,249	53,488
0.9003	0.802	0.248	8,533	25,579	5,419	2,042	21227	35,748	21,936	27,194
0.9003	0.802	0.266	8,533	25,569	5,419	4,454	21,227	59,016	21,936	54,165
0.9003	0.802	0.283	8,533	25,529	5,419	8,593	21,227	60,410	21,936	65,246

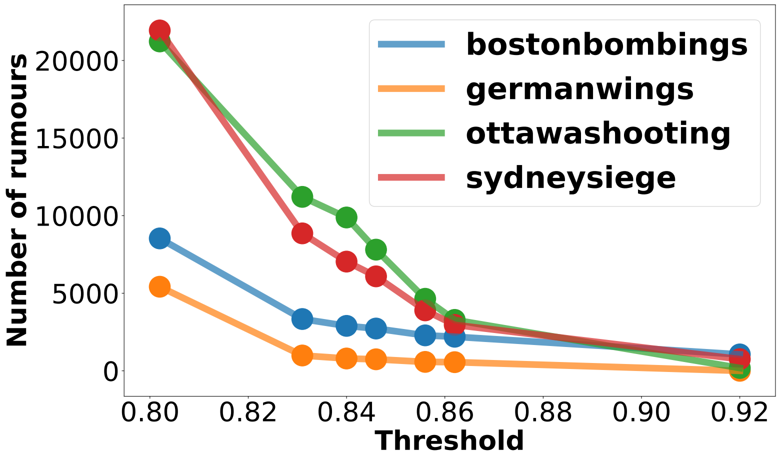


Figure 4.3: Visualisation of the number of augmented rumour source tweets for different thresholds.

thresholds higher than 0.802, the number of augmented rumour sources for that event is less than 1,000. As described in Section 4.3, source tweets without replies are excluded from resulting data and a considerable reduction is expected at this stage of data augmentation process. The size of resulting data is an important factor to be considered in the generation of rumour data sets (Zubiaga et al., 2016b). It is natural that a threshold lower than 0.802 selects more rumour source tweets. In Table 4.7, for instance, $\theta_1 = 0.760$ selects two to three times the number of rumour source tweets selected by $\theta_2 = 0.802$. However, $\theta_1 = 0.760$ reduces precision by 5%. Based on the findings and considering a trade-off between augmented data size and precision, this chapter sets θ_1 to 0.802.

To choose a threshold θ_2 for augmenting non-rumour source tweets, θ_1 is fixed to 0.802. Thresholds $\theta_2 = 0.248$ and $\theta_2 = 0.266$ achieve the same performance in the benchmark task. With these values, the number of rumour sources is greater than that of non-rumour sources for the "germanwings", while the former is generally smaller than the latter for the other events.

Therefore, the third smallest value, $\theta_2 = 0.283$, which shows a very marginal improvement in F1-score is also tested. With this value, the number of non-rumour sources is larger than that of rumour sources for the "germanwings". Therefore, it seems that $\theta_2 = 0.283$ is a reasonable choice. However, this chapter set θ_2 to 0.266 in the experiments to ensure the minimum level of noise in resulting data. Future work may investigate the effects of higher values for θ_2 on the quality of augmented data.

4.6 DATA AUGMENTATION

Given the fine-tuned ELMo for tweet embeddings and two semantic relatedness thresholds, this section describes how to annotate unlabelled tweets using weak supervision. The RQ 1.1 is addressed in this section.

RQ 1.1: To what extent could the size of existing training data for rumour detection be extended with approaches based on semantic relatedness?

4.6.1 *Methodology for Data Augmentation*

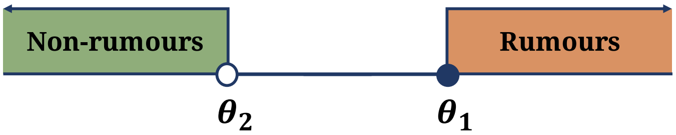


Figure 4.4: Visualisation of selecting rumour and non-rumour source tweets using two semantic relatedness thresholds.

For each event, all references and candidate tweets are encoded using an embedding model (see Section 4.5.1 for details about the selection of an embedding model). Given all pairs of embeddings of references and candidates, pairwise cosine similarity is computed. If a semantic similarity score between a candidate and *one or more* references is greater than or equal to the first threshold θ_1, the candidate is included in a rumour source collection (see Figure 4.4). If a candidate is identified as a rumour for any of rumour references, this candidate is included in the rumour collection. For non-rumour sources, it is assumed that low semantic relatedness to rumour references indicates the high likelihood of being a non-rumour. If and only if a semantic similarity between a candidate and *every* rumour reference is less than the second threshold θ_2, the candidate is included in the non-rumour source collection (see Figure 4.4). This approach may result in noisy training labels which contain overlaps and lack coverage. For example, rumour source tweets which are not covered by references may be identified as non-rumour sources. Despite these limitations, weak supervision can increase the efficiency, reduce the cost of manual annotations, and improve the usability of massive amounts of unlabelled data (Ratner et al., 2017). Ultimately, high-performance rumour detection models leveraging weak labels can be robust to unseen noisy inputs, which will also improve their flexibility in practice.

Based on the experimental results presented in Section 4.5.2.1, θ_1 and θ_2 are determined. For sampling rumour sources, a threshold $\theta_1 = 0.802$, which achieves a precision of 0.9 in the paraphrase identification task illustrated in Section 4.5.2, is used. For sampling non-rumour sources, a threshold $\theta_2 = 0.266$, which achieves the lowest precision in the same task (see Table 4.6), is used. The details of augmenting source tweets using the two thresholds are described in Section 4.6.1. For each event, if a candidate is identified as a rumour for any of rumour references, it is annotated as a rumour.

One of the research aims stated in Section 3.1 is addressing the class imbalance problem of publicly available rumour data sets (Kochkina et al., 2018a). To achieve this goal, class distributions in the augmented data will be balanced. Data augmentation results after applying thresholds show high class imbalance for all events except the "germanwings", which will be discussed in detail in Section 4.6.2 (see Table 4.9 for a brief insight). Specifically, random sampling is applied to augmented non-rumour source tweets. Before sampling, replies and retweets for all rumour and non-rumour source tweet in the augmented data are collected as described in Section 4.2.5 and source tweets without replies are removed from the augmented data. A considerable reduction in augmented data size is observed because a large number of source tweets do not have replies. Subsequently, $(\mathbf{2} * (\texttt{the number of rumour source tweets}))$ non-rumour source tweets are randomly sampled in each event data set. In order to keep source tweets which are rich in conversational threads, all source tweets that have more than 10 replies are included. The remainder is randomly chosen. Finally, augmented rumour and non-rumour source tweets with replies and retweets are merged with the *PHEME5*.

Fundamentally, a data set has class imbalance if the ratio of the majority class and the minority one is not one to one. However, defining the degree of class imbalance (e.g. low, moderate, high, severe, etc.) is dependent on the goal of a task. For example, a study (Leevy et al., 2018) on class imbalance in big data states that a large-scale data set suffers from "high class imbalance" if the ratio is between $100:1$ and $10,000:1$. On the other hand, a study (Prusa et al., 2015) on class imbalance in tweet sentiment analysis states that a data set is "highly imbalanced" if minority instances constitute less than 10% of the entire data. In this chapter's experiments on ERD using the augmented data, class distributions will be equally balanced (i.e. random sampling while keeping source tweets with rich context). The main reason to keep the ratio of rumour and non-rumour sources one to two in the final augmented data is to provide richer information to other researchers employing it for different downstream tasks related to online rumours such the characterisation of rumours and their spreading different from non-rumours.

4.6.2 *Results of Data Augmentation*

The augmented data and temporally filtered augmented data will hereafter be referred to as *Aug-PHEME* and *Aug-PHEME-filtered* throughout the thesis.

4.6.2.1 *Statistics of Aug-PHEME*

Rumour and non-rumour source tweets are augmented for the six selected events in the *Twitter events 2012-2016* data. Table 4.8 shows the number of source tweets for rumours and non-rumours in the Aug-PHEME obtained by simply applying the two thresholds θ_1 and θ_2 (i.e. no filtering or sampling is applied). It also shows the number of references available in the Aug-PHEME. Comparing Table 4.8 with Table 4.3 shows that not all references exist in the

Table 4.8: The number of rumour and non-rumour source tweets in the augmented data.

Event	Rumour	Non-rumour	Number of ref.
charliehebdo	23,073	48,770	53
fergusonunrest	9,359	25,301	24
germanwings	5,419	4,454	13
ottawashooting	21,227	58,998	40
sydneysiege	21,936	54,165	49
bostonbombings	8,533	25,574	45
Total	89,547	217,262	

Table 4.9: Number of rumour and non-rumour source tweets and replies in the Aug-PHEME. Values in parentheses are statistics for the original PHEME5.

	Aug-PHEME				**After merging and balancing**			
	Rumour		**Non-rumour**		**Rumour**		**Non-rumour**	
Event	**source**	**threads**	**source**	**threads**	**source**	**threads**	**source**	**threads**
bostonbombings	392	2,084	3,231	31,290	392 (N/A)	2,084 (N/A)	784 (N/A)	24,536 (N/A)
charliehebdo	802	3,565	4,437	22,969	1,225 (458)	10,152 (6,888)	2,450 (1,621)	45,765 (29,302)
fergusonunrest	475	2,222	2,934	11,168	737 (284)	8,184 (6,196)	1,476 (859)	24,639 (16,837)
germanwings	272	1,028	373	1,099	502 (238)	3,231 (2,256)	604 (231)	2,863 (1,764)
ottawashooting	625	3,335	3,607	18,340	1,047 (470)	8,860 (5,966)	2,072 (420)	20,933 (5,428)
sydneysiege	1,289	4,632	3,955	14,673	1,764 (522)	12,330 (8,155)	3,530 (699)	27,797 (14,621)
Total	3,855	16,866	18,537	100,439	5,667 (1,972)	44,805 (29,461)	10,916 (3,830)	146,533 (67,952)

Aug-PHEME. This indicates that semantic similarity between some references and *all* candidate tweets is below θ_1. Appendix a.2 shows references and the number of rumour source tweets relevant to each individual reference in the Aug-PHEME. After downloading contexts as described in Section 4.2.5, the augmented tweets for the *PHEME5* events are merged with the original *PHEME5*. Table 4.9 shows the number of source tweets and replies (i.e. threads) obtained via the proposed data augmentation method. It also shows those after merging the Aug-PHEME with the original *PHEME5* and balancing the merged data. Note that all source tweets in the original *PHEME5* are kept in the merged data. The values in the parentheses are the number of tweets in the original *PHEME5*. Overall, the number of source tweets for rumours and non-rumours increased by 187% and 185%, respectively. There are 52% and 110% increases in the number of replies for rumour sources and that for non-rumour sources, respectively. The standard deviation of imbalance ratios of non-rumour sources to rumour source improved from 1.24% to 0.35%, respectively. In particular, high class imbalance in two largest events–"fergusonunrest" and "charliehebdo"–has been mitigated as a result of data augmentation. Specifically, the ratio of rumour source tweets to non-rumour source tweets were 1 : 3.5 and 1 : 3 for the former and latter, respectively. In the Aug-PHEME sets, the ratios have improved to 1 : 2 for both events.

4.6.2.2 *Statistics of Aug-PHEME-filtered*

Since the aim of these experiments is to show the usefulness of data augmentation for ERD, temporal filtering is applied to every individual event data set in the experiments. Specifically, source tweets posted before the occurrence date of an event are filtered out. The UTC time zone is used in the experiments. Next, contexts posted in the first seven days of the creation of their source tweets are kept in the Aug-PHEME-filtered. Table 4.10 shows the number of source tweets and replies and the dates used for temporal filtering. Table 4.11 and 4.12 show detailed statistics of the Aug-PHEME-filtered. The statistics of original PHEME data are given in parentheses for comparison.

Table 4.10: Data statistics after temporal filtering and deduplication.

		Rumour		**Non-rumour**	
Event	**Date**	**source**	**thread**	**source**	**thread**
germanwings	30/24/2015	375	2,801	402	2,202
sydneysiege	12/14/2014	1,134	10,271	2,262	19,547
ottawashooting	10/22/2014	713	7,117	1,420	10,522
fergusonunrest	08/09/2014	471	7,103	949	19,545
charliehebdo	01/07/2015	812	8,356	1,673	34,435
bostonbombings	04/15/2013	323	1,973	645	21,871
Total		3,828	37,621	7,351	108,122

Table 4.11: Statistics of *replies* in the Aug-PHEME-filtered. Values in parentheses are statistics for the original PHEME5.

	Rumour					**Non-rumour**				
Event	**Mean**	**Mdn**	**Std**	**Min**	**Max**	**Mean**	**Mdn**	**Std**	**Min**	**Max**
germanwings	8	4	11	1	79	6	3	7	1	35
	(11)	(7)	(11)	(1)	(76)	(9)	(6)	(8)	(1)	(35)
sydneysiege	9	4	13	1	209	9	3	17	1	341
	(16)	(15)	(14)	(1)	(173)	(22)	(18)	(24)	(1)	(341)
ottawashooting	10	6	14	1	208	8	4	11	1	114
	(13)	(11)	(11)	(1)	(107)	(14)	(10)	(13)	(1)	(103)
fergusonunrest	16	9	22	1	200	23	17	28	1	288
	(24)	(18)	(25)	(2)	(200)	(22)	(16)	(27)	(1)	(288)
charliehebdo	11	5	18	1	224	21	17	26	1	341
	(15)	(12)	(18)	(1)	(177)	(19)	(16)	(20)	(1)	(341)
bostonbombings	6	2	15	1	207	34	20	39	1	197
	(–)	(–)	(–)	(–)	(–)	(–)	(–)	(–)	(–)	(–)

4.6.2.3 *An Empirical Study of Augmented Data*

The manual inspection of sampled source tweets shows that augmented data contains tweets identical to references and several variations of references. It is worth noting that the proposed data augmentation with weak supervision

Table 4.12: Statistics of *retweets* in the Aug-PHEME-filtered.

	Rumour					Non-rumour				
Event	**Mean**	**Mdn**	**Std**	**Min**	**Max**	**Mean**	**Mdn**	**Std**	**Min**	**Max**
germanwings	52	49	26	14	96	63	75	26	5	98
sydneysiege	40	24	38	0	96	27	3	36	0	97
ottawashooting	59	79	35	0	98	34	15	36	0	95
fergusonunrest	53	78	38	0	95	81	85	16	0	97
charliehebdo	54	77	37	0	97	83	86	13	0	99
bostonbombings	25	11	29	0	94	69	77	22	0	94

can even capture rumours which are related but not technically identical to reference tweets. Example 4.3 shows some examples of rumour tweets in the augmented data.

Example 4.3:

(1) "**A 20-year-old student** is among the hostages at the kosher shop in Paris http://t.co/orBfH8MK1J"

(2) "Uber Promises **Free Rides** in Sydney after Surge Pricing Kicks in During Hostage Crisis http://t.co/7NAO9HSxEA"

Example 4.3 (1) is almost identical to a reference tweet, "**A baby** is among the hostages in the Kosher market", for the Charlie Hebdo attack, except for subjects of sentences. The semantic similarity score between two sentences is 0.8123. Example 4.3 (2) is a variation of a reference tweet, "Uber introduced **surge pricing** in down town Sydney during hostage crisis.". Two sentences report contradictory sub-events related to a taxi booking company called Uber, but their semantic similarity score is 0.8238.

Using raw annotated tweets as references rather than refined categories of rumours may help to retrieve more positive examples. In the *PHEME (6392078)*, for example, a tweet, "Ray Hadley says he spoke with hostage, and could hear the gunman in the background barking orders and demanding to go live on air", is annotated as a rumour category, "The gunman and/or hostages have made contact with Sydney media outlet(s) (radio station, etc.)". Without a background knowledge that Ray Hadley is an Australian radio broadcaster, data augmentation methods based on semantic relatedness fail to identify such rumours.

4.7 EARLY RUMOUR DETECTION

RQ 1.3: Does data augmentation improve the performance of deep learning-based ERD architectures? How can this be assessed?

Remember that the main purpose of rumour data augmentation in this thesis is to improve the performance of deep learning models for rumour detection by increasing the size of existing rumour data sets. Now that augmented rumour data is now available, this section addresses the RQ 1.3.

4.7.1 *Experiments on Early Rumour Detection*

4.7.1.1 *Data*

The *Aug-PHEME* consists of data sets for the 5 events in the *PHEME (6392078)* and one new event "Boston marathon bombings". The detailed statistics for each event data set is presented in Section 4.6.2. Rumour detection experiments are conducted using the *PHEME5* and two versions of augmented data sets, i.e. *Aug-PHEME-filtered* and *Aug-PHEME-filtered -boston*. *Aug-PHEME-filtered* is the augmented data for the five events in the *PHEME5* (see Section 4.6.2.1). *Aug-PHEME-filtered -boston* is the *Aug-PHEME-filtered* excluding the "bostonbombings" event. Remember that the references for this event were not included in the original PHEME. The augmented data for this may have different characteristics from the other five events in the PHEME, and hence it is worth experimenting with and without this event's corpus.

4.7.1.2 *Model*

This chapter exploits a SOTA baseline model for rumour detection (Kochkina et al., 2018a) because it uses the benchmark data (i.e. *PHEME (6392078)*) for this chapter's data augmentation, uses conversational threads as context, and is designed for tweet-level rumour detection rather than event-level classification. The proposed model performs multi-task learning which combines 1) rumour detection, 2) stance classification, and 3) rumour verification. It consists of a shared LSTM layer followed by task-specific layers. First of all, the task of stance classification requires labels for every reply in a conversational thread. Therefore, the hidden states at every time step of the shared LSTM are fed into multiple dense ReLU layers followed by a Softmax layer that predicts a probability distribution over classes for each input source tweet. For the other two tasks, the hidden state at the last time step in the shared LSTM is fed into another LSTM layer followed by multiple dense ReLU layers and a Softmax layer. In their model, source tweets and replies are represented as 300-dimensional word2vec (Mikolov et al., 2013) word embeddings pre-trained on the Google News data set [17].

This model is employed with three modifications in this chapter's experiments. Firstly, as this chapter only concerns with rumour detection, only rumour detection module (i.e. a shared LSTM + a task-specific LSTM + a dense layer with the softmax function) is implemented in the experiments. Source code is obtained from http://github.com/kochkinaelena/Multitask4Veracity and the modified implementation is available via https://github.com/soojihan/Multitask4Veracity.

Secondly, the structure of input sequences is modified. In the original model, each conversation consists of a source tweet and replies to it. Conversational threads are decomposed into several branches based on stances and user mentions (i.e. "@[username]") appearing in tweets. In the experiments, an entire conversation thread of each source tweet is considered as a single branch. As this thesis aims at ERD, only replies published in the early stages of rumour spreading should be considered. Kochkina et al. (2018a) set the maximum length of each branch to 25. Considering these observations, each

[17] https://code.google.com/archive/p/word2vec/

input sequence of this chapter's experiments consists of embeddings of each source tweet and its top (i.e. most recent) 24 replies. One may argue that considering only 24 replies for each source tweet is not sufficient or using a fixed length is not optimal. However, it should be noted that rumour detection is not the main task of this chapter. Here, the aim is to show augmented data can improve the effectiveness of SOTA NN model for rumour detection. In the Chapter 6 which focuses on rumour detection, the maximum number of replies for each source tweet is set to 200 to overcome such a limitation. The original model requires input with shape: (the number of branches in each event dataset, the maximum length of a branch, 300). Therefore, the modified model requires input with shape: (the number of source tweets in each event data, 25, 300). Zero-padding and masks are applied to handle varying lengths of input conversational threads.

Finally, training and hold-out sets are generated completely independently from test sets (see details in Section 4.7.1.3). In the original implementation, the "charliehebdo" data set is used as a validation set for hyperparameter optimisation for all LOOCV iterations. This means that there exists an LOOCV iteration in which the validation set is the same as the test set, which results in a biased evaluation. Note that a test set should never been used in training. As for hyperparameter optimisation, a grid search with the parameter space defined by Kochkina et al. (2018a) is implemented. Parameter combinations are optimised based on accuracy on the validation set over 30 trials.

4.7.1.3 *Evaluation*

In order to evaluate the performance and ability of the model using the augmented data in a realistic scenario, LOOCV is performed. Simply speaking, one event is used as a test set and the remaining events are used as a training set on each iteration. This setting makes it possible to evaluate rumour detection in real-world scenarios in which detection models are required to identify unseen rumours. For the *PHEME5*, four out of five original PHEME5 events are shuffled and split into training and hold-out sets. The remaining one event is used as a test set for evaluation. Thus, a 5-fold LOOCV is applied. The *Aug-PHEME-filtered -boston* and *Aug-PHEME-filtered* are also evaluated in a 5-fold LOOCV setting as the *PHEME5*. The only difference is that training and hold-out sets are generated using the augmented data. In other words, test sets generated from the *PHEME5* are used for all three settings. Class distributions of training, validation, and test sets are equally balanced. This helps to evaluate the contribution of data augmentation on rumour detection by mitigating class imbalance.

4.7.2 *Results of Early Rumour Detection*

Rumour detection experiments are conducted on three data sets: (1) *PHEME5*, (2) *Aug-PHEME-filtered -boston*, and (3) *Aug-PHEME-filtered*. Kochkina et al. (2018a)'s method is employed as a SOTA baseline model of rumour detection with slight modifications (see Section 4.7.1). Table 4.13 shows the overall performance of rumour detection with three different data sets. The values of four evaluation metrics are the mean scores of all LOOCV iterations. The overall results show that data augmentation helps to boost performance on rumour

Table 4.13: Rumour detection results for different data sets.

Data	F	P	R	Acc.
PHEME5	0.535	0.650	0.484	0.622
Aug-PHEME-filtered -boston	0.625	0.688	0.585	0.664
Aug-PHEME-filtered	**0.656**	**0.716**	**0.614**	**0.685**

Table 4.14: LOOCV results for the PHEME5 and augmented data sets.

Event	Data	F	P	R	Acc.
germanwings	PHEME5	0.577	0.619	0.541	0.604
	Aug-PHEME-filtered -boston	**0.601**	**0.652**	**0.558**	**0.630**
	Aug-PHEME-filtered	0.575	0.650	0.515	0.619
sydneysiege	PHEME5	0.583	0.714	0.492	0.648
	Aug-PHEME-filtered -boston	**0.695**	**0.755**	**0.644**	**0.717**
	Aug-PHEME-filtered	0.632	0.759	0.542	0.685
fergusonunrest	PHEME5	0.242	0.550	0.155	0.514
	Aug-PHEME-filtered -boston	0.416	0.618	0.313	0.560
	Aug-PHEME-filtered	**0.609**	**0.707**	**0.535**	**0.657**
ottawashooting	PHEME5	0.516	0.653	0.426	0.600
	Aug-PHEME-filtered -boston	0.671	0.680	**0.662**	0.675
	Aug-PHEME-filtered	**0.697**	**0.739**	0.660	**0.713**
charliehebdo	PHEME5	0.758	0.714	0.808	0.742
	Aug-PHEME-filtered -boston	0.742	**0.734**	0.749	0.739
	Aug-PHEME-filtered	**0.767**	0.723	**0.817**	**0.752**

detection in terms of F-score (F), precision (P), recall (R), and accuracy (Acc.) The model performance in terms of F-score increases by 9% and 12% with *Aug-PHEME-filtered -boston* and *Aug-PHEME-filtered*, respectively. Table 4.14 shows the details of LOOCV results described in Section 4.7.1. The "Event" column in Table 4.14 shows 5 different events used as a test set on each iteration of LOOCV. It is worth noting that the "fergusonunrest" was the most difficult event in the *PHEME5* for a rumour detection model as it has a unique class distribution distinguished from all the other events (Kochkina et al., 2018a). With the *Aug-PHEME-filtered*, the F-measure on this event increases by 36.7%.

4.8 CONCLUSION AND FUTURE WORK

This chapter proposed a new paradigm of data augmentation for effectively enlarging existing rumour data sets using publicly available, large-scale, and unlabelled data for real-world events on social media. Semantic relatedness was exploited to apply weak supervision to unlabelled data based on limited labelled rumour source tweets. The experiments have shown that the potential efficiency and effectiveness of semantically augmented data for combating the scarcity of labelled data and class imbalance of existing publicly available rumour data sets. Augmented data is highly realistic and can potentially increase the diversity of existing labelled data and improve its quality. Preliminary results achieved using a SOTA DNN model indicate that augmented data is helpful to train DNNs. More extensive experiments using the data will be described and discussed in Section 6. This augmented data was released in the hope that it will be useful for further research in the field of rumour detection and general studies of rumour propagation on social networks. Future work will extend the proposed method to other events and training tasks in order to build more comprehensive data for rumour detection. Further research will also look into more advanced techniques for rumour variation identification. In addition, it is arguable that different types of rumour events may expose different propagation patterns. In addition, whether data augmentation creates a bias towards detecting the same sort of rumours will be explored. Increasing diversity and reducing bias in training data will be a future research direction.

5 IDENTIFICATION OF POTENTIAL RUMOURS

5.1 INTRODUCTION

The two main outcomes of the previous chapter are weakly labelled, large-scale rumour data and a task-specific, fined-tuned NLM. In particular, the main contribution of the former to the research community is that it provides larger training data for rumour detection models, which leads to performance gains. In this chapter, the augmented data will be used to evaluate potential rumours (i.e. input to a rumour detection model). This makes the proposed evaluation approaches for burst detection and text summarisation unique and novel. This chapter addresses the RQ 2.2. Previous research related to this chapter was introduced in Section 2.3.2.

RQ 2.2: How can candidates for rumours (i.e. potential rumours) be selected with minimal human supervision and time delay?

Due to intrinsic characteristics of social media data (e.g. rapid speed, vast volume, etc.), data reduction is necessary and this can be done by producing summaries which offer insights about further data exploration (Sharifi et al., 2013). Based on the analysis results and discussion in Section 2.5, this chapter addresses the problem of identifying *potential rumours* that emerge during breaking news events on social media via *key burst detection* and *summarisation*. The aim of the task of identifying potential rumours is to detect dubious and ambiguous claims rather than detecting false statements or classifying whether a post is related to a rumour or non-rumour. It should be noted that the verification of the truthfulness of rumours (i.e. determining whether a rumour is false or true) is beyond the scope of this chapter and thesis (see Section 1.1).

***Data reduction**: the process of automatically filtering out less significant and/or invalid data in the initial stages of data analysis*

Within current SOTA frameworks for automated message-level rumour detection, a classifier is designed to determine whether every single input tweet is a rumour or non-rumour based on a set of features. However, a tremendous number of noisy messages are generated on social media during a breaking news event in real-world scenarios. Tweets during breaking news events are usually collected using a set of keywords related to an event or geolocation (Olteanu et al., 2015; Zubiaga, 2018). For example, relevant keywords such as “bostonexplosion”, “bostonbombing”, “bostonblast”, and “prayforboston” can be identified in the early stages of the Boston marathon bombings event. In particular, hashtags starting with “prayfor” are very popular worldwide and usually appear when tragic events take place. Due to this data collection procedure, there can be a considerable number of irrelevant or uninformative posts in a stream. Hence, it is inefficient and infeasible for a classifier to analyse all messages generated during an event in real time. Existing studies select candidate rumours based on their popularity which is often represented by the number of retweets (Zubiaga et al., 2016b; Ma et al., 2017). However, this is not suitable for *ERD* in a real-world setting because it takes some time for tweets to receive a high number of reactions (Zhao et al., 2015).

This chapter aims to partially fill this gap by studying how to identify potential rumours appearing during breaking news events without manually examining a large number of messages and while minimising the delay in identification. The term "potential rumours" refers to claims which should be further examined by human experts or analysed by automated rumour detection models to be confirmed as rumours (Zhao et al., 2015). To this end, early signals for rumours were investigated and identified in Section 2.5. Its results show that *temporal signals (bursts)* are a promising key to the early identification of potential rumours. This chapter addresses the proposed task via a two-step framework which comprises of key burst detection and summarisation. The proposed framework relies on both peaks and bursts of user activity. Bursts and peaks identified using the framework are collectively referred to as *key bursts*.

There exists a downside of using bursts as early signals for potential rumours. Bursts can be considered as anomalies in time series (Zubiaga et al., 2012). As existing research on outlier detection suggests, the distinction between outliers and normal instances can be regulated based on the interest of analysts or decision-makers (Aggarwal, 2013). Therefore, bursts are employed as *weak supervision* for the task of early identification of potential rumours. As the interest of this chapter is to fulfil the task without analysing every tweet available while an event unfolds, this chapter proposes and evaluates burst detection and summarisation in the context of ERD. Although the proposed method produces noisy outputs as all weak supervised learning approaches do, this chapter shows that it is simple yet efficient and effective for the proposed task. The method can benefit not only researchers in the field of rumour studies but also practitioners in a broad range of domains including emergency responders and journalists by providing insights about events of interest and what the public is interested in. This chapter also proposes a novel evaluation approach suitable for ERD. This chapter is based on my publication in the proceedings of the 16th International Conference on Information Systems for Crisis Response And Management (Han and Ciravegna, 2019).

5.2 SYSTEM DESIGN

5.2.1 *Definition of Burst*

In this thesis, a "burst" is defined as a sudden increase in the number of social media posts for a short period following the definition of a major dictionary[1] and existing work leveraging bursts in Twitter (Lee et al., 2011; Myers and Leskovec, 2014).

This section discusses why peaks *by themselves* are not appropriate early signals for key bursts. Suppose it should be determined whether the time window marked with a square in Figure 5.1a is a burst or not. This task may become less challenging as soon as the next time window is visible. In Figure 5.1b, for example, the time window marked with a square is a key burst without a doubt because it is a "peak". However, identifying a peak cannot avoid time delays (e.g. time elapsed between the time windows marked with a square and star in Figure 5.1b and c), which does not comply with "early detection" and "time delay minimisation" requirements. To avoid

[1] https://dictionary.cambridge.org/dictionary/english/burst

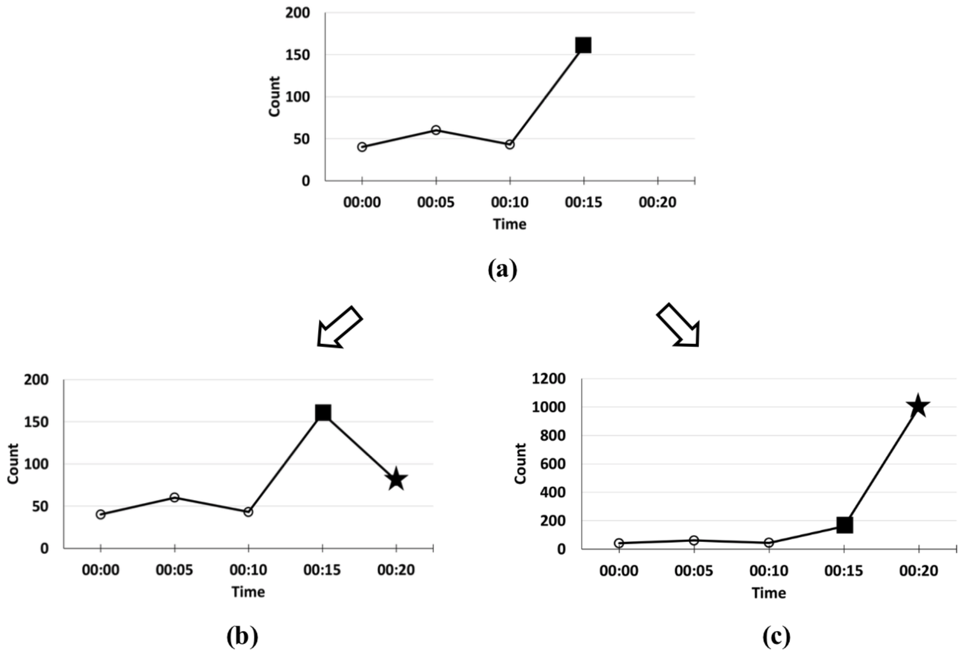


Figure 5.1: Examples of event time series plots. The x-axis is time and the y-axis is the number of messages about a specific event.

such time delays, key bursts should be detected without observing future instances. In other words, a method for burst detection should rely exclusively on information drawn from the left side of each time window in time series.

The following explanation will help to further understand how such method should be designed. In Figure 5.1c, it is difficult to decide whether the window marked with a square (i.e. 00:15) is a key burst or not even after seeing the time window marked with a star (i.e. 00:20). This is because an increase in the number of posts from the previous window (i.e. 00:10) is not significant compared to a variation between the time windows marked with a square and star. However, at the time when the window marked with a square appears for the first time in Figure 5.1a, it is probable that a human identifies it as a key burst because it lies on the increasing line of the time series plot and the increase is significant compared to preceding variations.

Even if a peak is not an early signal on its own, it is a useful cue for potential rumours. Therefore, bursts and peaks used to identify potential rumours are collectively referred to as *key bursts* in this thesis.

5.2.2 *Architectural Design*

This section describes the design of desirable architecture for burst detection by investigating two limitations which several existing methods have. This section also introduces other possible methods for the identification of potential rumours other than temporal signal-based burst detection and explains why they are not an optimum solution for the task.

As burst detection is a subjective task, several existing studies rely on thresholds which can be adjusted according to stakeholders' interests. In particular, the use of thresholds is common practice for research on burst detection based on *temporal signals* (Hsieh et al., 2012; Nichols et al., 2012;

Gillani et al., 2017; Doman et al., 2014; Zubiaga et al., 2012; Peng et al., 2018; Shamma et al., 2009; Kong et al., 2015). Their methods tend to how two limitations.

The first limitation is that they tend to capture a large number of spurious bursts and miss relatively small bursts (Meladianos et al., 2015). This observation is clearer with methods which simply rely on basic statistics of the number of messages (e.g. mean and standard deviation) and focus on peaks (Zubiaga et al., 2012; Shamma et al., 2009; Hsieh et al., 2012; Nichols et al., 2012; Gillani et al., 2017). Patterns of event evolution on social media exhibit bursts of intense activity separated by long periods of inactivity or low-frequency periods (Barabási, 2005). Due to this intrinsic characteristic, there can be significantly large peaks easily distinguished from other parts in event diffusion plots. Some examples of the visualisation of event evolution on Twitter can be seen in Figure 2.4. The main issue of using combinations of the basic statistics of the number of messages as thresholds for identifying bursts is that they tend to stay high after a huge peak. Consequently, relatively small bursts appearing after the peak are ignored even if they might contain emerging unseen rumours or developing rumours (i.e. correcting, debunking, and/or verifying previously seen rumours; Meladianos et al. (2015)). With regards to spurious bursts, this chapter further investigates periods of persistent decline in time series plots of events on social media. Most of the studies introduced above consider all time windows, at which the number of messages is above a threshold, as bursts. This means that *instances located in a persistently falling line (decay)* are identified as bursts due to their frequency. Note that they are not bursts according to the definition employed in this thesis. However, it is worth investigating whether decay can be considered as key bursts in the task of potential rumour identification. To do so, this chapter proposes a hypothesis that the same topic can be persistently discussed even after its popularity reaches a peak for several reasons such as differences in time zones of users. Chapter 5 performs a case study on if major emerging rumours appear in bursts or decaying lines for the first time (see Section 5.4.2).

In this thesis, a burst is defined as a sudden increase in the number of social media posts for a short period.

The other limitation is that thresholds used in existing methods for burst detection based on temporal signals are task-specific and their effectiveness depends heavily on the characteristics of data (e.g. sizes and burstiness) (Shamma et al., 2009). In particular, parameters associated with thresholds behave arbitrarily and/or their effectiveness varies between events (Doman et al., 2014). Most work does not explain the effects of their parameters on results. This makes it difficult to employ existing methods to detect bursts for new events and domains. Chapter 5 proposes a simple yet effective burst detection algorithm based on temporal signals by addressing this limitation. The proposed method is compared with SOTA methods (Peng et al., 2018; Gillani et al., 2017) on burst detection based on the behaviour of parameters (Section 5.5.1). As Chapter 5 conducts a thorough analysis of parameters (Section 5.4.3), end users can optimise the proposed method according to their interests and objectives.

Other possible approaches for identifying potential rumours include 1) bursty topic detection and 2) clustering.

As for the bursty topic detection, Zhao et al. (2015) report that identifying rumours based on bursty topics from an entire corpus is computationally expensive and not suitable approaches for ERD. The focus of this chapter should be distinguished from burst/event detection via topic modelling. Although bursty topic detection can provide insights about general trends or overall semantic representations of sub-events (Xing et al., 2016), its output (typically topic words or phrases) is not appropriate to be used as potential rumours for message-level ERD. Actual tweets, which are worth being examined by a rumour detection model to be confirmed as rumours, should be identified for the task of Chapter 5. Once topic words are identified, it is possible to collect tweets containing them using other algorithms. The simplest way to do this is to retrieve all tweets containing any of the identified topic words. However, this method is likely to identify several irrelevant and uninformative tweets. Therefore, other techniques such as noise filtering algorithms should be employed.

As for clustering, it is often used in event detection (Srijith et al., 2017; Weng and Lee, 2011; Xing et al., 2016; Li et al., 2012; Zhang et al., 2015b; Becker et al., 2011; Hasan et al., 2016; Phuvipadawat and Murata, 2010; Unankard et al., 2015). Conventional clustering algorithms usually fixes the total number of clusters, which makes it infeasible to adopt them to Twitter data containing massive amounts of messages covering a wide range of topics (Hasan et al., 2018). Incremental clustering algorithms avoid this issue by processing one data sample at a time and incrementally assigning it to a corresponding cluster (Ackerman and Dasgupta, 2014). However, further analysis of output clusters is required to extract potential rumours. For instance, depending on algorithms and events, a large number of clusters can be identified. For example, (Srijith et al., 2017) report that $2,500$, $5,000$, and $45,000$ clusters were identified by three different clustering algorithms for England riots in 2011. They are large compared with the number of manually annotated rumour events. For example, the average number of rumour stories manually annotated by journalists for five breaking news events in the PHEME (6392078) data is 46 (Table 4.3). This observation indicates that not all clusters are related to potential rumours, and thus, further analysis to select clusters of potential rumours is required for clustering-based approaches. For example, Hasan et al. (2016) filter out insignificant events (i.e. clusters) based on entropy, user diversity, the number of tweets in a cluster, and links to news media outlets. Becker et al. (2011) employ a SVM classifier trained with cluster-level features to distinguish event clusters from non-event clusters. An example of the latter is a cluster which mostly comprises of retweets of a source tweet posted by a popular user.

Putting it all together, this thesis proposes a framework combining burst detection and summarisation as a preliminary step (i.e. data reduction) for tweet-level ERD. Its details are described in Chapter 5. The proposed framework is capable of generating candidate tweets considering both temporal dynamics (via *burst detection*) and linguistic significance (via *summarisation*). The fine-grained detection, tracking, stance classification, and verification of rumours are beyond the scope of this chapter.

5.3 METHODOLOGY

5.3.1 *Overview of the Proposed Framework*

When a breaking news event occurs in the real world, event-specific keywords selected by human experts are used to collect related social media posts (Zubiaga, 2018; Olteanu et al., 2015). These keywords are general but related to events. For the Boston Marathon bombings, for instance, "boston", "marathon", and "#prayforboston" can be used (Zubiaga, 2018). More examples are given in Table 4.2.

Given a collection of tweets related to an event of interest, a time series is generated using tweets' posting times. Taking the time series as input, key burst detection is performed on a newly emerging time window. If the window is identified as a key burst, representative tweets are extracted from the collection of tweets posted during the burst.

Key bursts are identified based on temporal features extracted using preceding time windows observed up to the current time in order to provide summaries and identify potential rumours from them. To extract summaries from identified key bursts, a collection of tweets of each burst is represented as an undirected weighted graph of words. Vertices of the graph are words and edges indicate co-occurrence of two words in every tweet in the collection. Next, word scores (i.e. weights of vertices of the graph) are computed. Given word scores, weights of individual tweets are computed. Finally, tweets are ranked in order of their scores and the top N tweets are selected as representative summaries for the burst.

The output of the proposed framework is referred to as potential rumours which draw people's attention during a specific time period and should be further examined by human experts or automated rumour detection models to be affirmed as rumours. Remember that this framework is proposed as a preliminary step for automatic rumour detection. The main purpose is to automatically generate candidate tweets which are likely to be related to rumours as early as possible.

5.3.2 *Key Burst Detection*

A rule-based algorithm based on temporal signals is proposed in this section. The method detects key bursts by solely using the left-hand side of any instances (i.e. time window) of a time series.

5.3.2.1 *Temporal Features*

The proposed method relies exclusively on temporal features that are computed using the number of tweets observed up to the current time window. Patterns of event evolution on social media exhibit bursts of intense activity separated by long periods of inactivity or low-frequency periods (Barabási, 2005). Temporal features should be able to well characterise bursty patterns of information diffusion on social media. To this end, this section proposes several temporal features classified into three categories: *pre-filtering features*, *frequency-based features*, and *gradient-based features*. Considering not only the absolute values but also differences in frequency between consecutive intervals help to better capture temporal fluctuations in time series (Ma et al., 2015). Most features are obtained after smoothing out random variations in time series as smoothing out time series help to remove irregular noise and reveal underlying trends in time series.

Proposed features are selected according to results of preliminary experiments on the Boston marathon bombings data in Twitter event data (2012-2016; Zubiaga (2018)). To find features that 1) effectively detect key bursts, and 2) behave in the same manner across different data sets (portability), key bursts detected using several combinations of statistics are visualised and manually examined. The features proposed by existing studies on burst detection based on temporal patterns (Hsieh et al., 2012; Nichols et al., 2012; Gillani et al., 2017; Doman et al., 2014; Zubiaga et al., 2012; Peng et al., 2018; Shamma et al., 2009) are used as a starting point for the experiments. Major findings from the initial experiments can be summarised as follows:

This data consists of over 147 million tweets associated with 30 real-world events unfolded between February 2012 and May 2016.

Given the number of tweets at time i, denoted by c_i, three groups of features according to their usage are used in the proposed key burst detection.

Pre-filtering

- CumSum (a_i): Sum of c with a window size of 10 (e.g. 10 minutes). A fraction of this value is used to pre-filter time windows with low frequency.

$$a_i = \begin{cases} 0, & i = 1 \\ \sum_{k=1}^{\min\{10, i-1\}} c_{i-k}, & i = 2, \cdots, N \end{cases}$$

Frequency-based features

- EWMMean (s_i): Exponentially weighted average of c.

$$s_i = \begin{cases} c_1, & i = 1 \\ (1-\alpha) * s_{i-1} + \alpha * c_i, & i = 2, \cdots N \end{cases}$$

,where smoothing factor $0 < \alpha \leq 1$.

- EWMRMean (p_i): Mean of s with a window size of 30

$$p_i = \begin{cases} 0, & i = 1 \\ \frac{\sum_{k=1}^{\min\{30, i-1\}} s_{i-k}}{\min\{30, i-1\}}, & i = 2, \cdots, N \end{cases}$$

- θ_i: Sum of s_{i-1} and the sample standard deviation of $\{s_{i-3}, s_{i-2}, s_{i-1}, s_i\}$.

$$\theta_i = \begin{cases} 0, & i = 1 \\ s_{i-1} + std_i, & i = 2, \cdots, N \end{cases}$$

Difference-based features

- EWMDiff (d_i): Difference between s_i and s_{i-1}. It can be negative.

$$d_i = \begin{cases} 0, & i = 1 \\ s_i - s_{i-1}, & i = 2, \cdots, N \end{cases}$$

- EWMDMean (z_i): Rolling mean of d. It exhibits very subtle variations. In other words, it tends to stay constant throughout event evolution except when there are extreme increases or decreases.

$$z_i = \frac{1}{i} \cdot \sum_{k=0}^{i-1} d_{i-k}, i = 1, \cdots, N$$

- ψ_i: Sum of the weighted mean of difference at time $i-1$ and the weighted difference at time i. It can be negative.

  $$\psi_i = \begin{cases} 0, & i = 1 \\ \lambda * z_{i-1} + (1-\lambda) * d_i, & i = 2, \cdots, N \end{cases}$$

  It smooths a time series, but exaggerates rising and falling patterns. EWMDiff and ψ behave in a similar way except that the amplitude of EWMDiff is wider than that of ψ. In other words, ψ is equivalent to d offset by z.

5.3.2.2 *Key Burst Detection Method*

Algorithm 1 Identification of key bursts.

Input: Time series $\mathcal{D} = \{x_i\}$ $(|\mathcal{D}| = N, x_i = (c_i, s_i, \theta_i, \psi_i))$
Pre-filtering factor γ

Output: $y_k \in \{0, 1\}$

```
for $i = 1$ to $N$ do
    H = constant
    while $i < H$ do
        AVG = mean$\{c_1, c_2, \cdots, c_i\}$          ▷ mean of the number of tweets up to time $i$
        THRESHOLD = min $\{AVG, 100\}$
        DIFF_THRESHOLD = max $\{\theta_1, \theta_2, \cdots, \theta_i\}$   ▷ the maximum $\theta$ up to the current time
    $y_i \leftarrow 0$
    if ($c_i$ >TRESHOLD) and ($c_i > 10$) then
        if $y_{i-1} == 1$ then
            if $s_i \geq \theta_i$ or $c_{i-1} \leq c_i$ then
                $y_i \leftarrow 1$
        else if $y_{i-1} == 0$ and $s_i > \theta_i$ then
            if $\psi_i$ >DIFF_THRESHOLD then
                $y_i \leftarrow 1$
    else
        $y_i \leftarrow 1$
```

In the preliminary work of this thesis, a method for detecting key bursts based on temporal signals is proposed (see Algorithm 1). Algorithm 1 describes a prototype for the updated model described in Algorithm 2. It describes a rule-based procedure of identifying key bursts using some of the features given in Section 5.3.2.1. Given a time series consisting of N windows and associated features, the algorithm assigns binary labels y_i to time windows based on a set of conditions. If the i^{th} time window is a key burst, y_i is 1, and 0 otherwise. The algorithm is experimentally fine-tuned using the "bostonbombings" data set in the *Twitter event data sets (2012-2016)* (see Section 4.2.4.1 for data description and Table 4.3 for statistics). Ideally, the goal of fine-tuning is to identify bursts lying in increasing lines and peaks in event evolution graphs. Detected bursts are visualised and rules in the algorithm are adjusted to fulfil the objective. As different events exhibit different spreading patterns, this chapter tests fine-tuned algorithms on the "sydneysiege" and "charliehebdo" data sets and examine if they can capture bursts in these events as well.

Some existing studies define thresholds for the absolute number of messages of each time window (Gillani et al., 2017; Shamma et al., 2009; Zubiaga et al., 2012). Others define thresholds for differences in the number of messages over several time windows (Nichols et al., 2012; Peng et al., 2018). Algorithm 1 overcomes several limitations that existing methods for key burst detection have. Firstly, results obtained using SOTA methods are dependent on spreading patterns of events (e.g. the total number of tweets posted during event evolution, burstiness, etc.) (Doman et al., 2014; Meladianos et al., 2015). This poses a challenge because the performance of a detection method can be poor if an inappropriate threshold is chosen. To overcome this, thresholds are updated in the early stages of event evolution (LINE $3-6$; Zubiaga et al. (2012)). The parameter H controls the number of time windows used for determining thresholds. A larger value of H is required for events which gradually develop. In general, it takes longer to draw people's attention in the case of progressive events such as "Ferguson unrest" as events develop more slowly than they do in instantaneous and rapidly evolving events such as "Boston marathon bombings" (see Section 2.4 and Table 4.2). Secondly, the proposed algorithm considers both the absolute volume (see LINE $8-11$) and differences in the number of messages (see LINE $12-14$) to detect key bursts. The main advantages of the proposed method over existing ones are as follows: 1) detected bursts have a significant number of messages to be considered anomalous, and 2) instances lying in decreasing lines in time series plots are filtered out. Despite these advantages, Algorithm 1 some technical limitations.

- **Limitation 1:** Even if thresholds are adaptive to evolution patterns of events, the selection of window size H still requires background knowledge of events (e.g. instantaneous versus progressive) and relies on decision-makers' intuition.

- **Limitation 2:** In Line 13 in Algorithm 1, a difference-based feature ψ is compared with a threshold based on a frequency-based threshold (i.e. DIFF_THRESHOLD). It is not able to clearly explain the correlation between them.

To overcome these limitations, this thesis proposes another method for key burst detection using temporal signals by addressing the limitations of the initial approach. Similar to Algorithm 1, Algorithm 2 aims to assign binary labels y_i to time windows in a time series. If the i^{th} time window is a key burst, y_i is 1, and 0 otherwise. It is fine-tuned using the same fine-tuning approach as Algorithm 1 is fine-tuned.

Firstly, if the number of tweets at time i is less than or equal to a cumulative sum a_i multiplied by a pre-filtering parameter γ (see Line 4), y_i is equal to 0. Otherwise, the next step of the algorithm is performed. γ is a constant between 0 and 1. This procedure ensures that a pre-filtering is based on a flexible threshold which is constantly updated throughout event evolution. γ does not require domain-specific knowledge, and it can be determined via hyperparameter optimisation such as a grid search and random search.

In the following step, whether the label of the previous time window, y_{i-1}, is 0 or 1 is checked (see Line 5 and Line 8). If $y_{i-1} = 1$, conditions

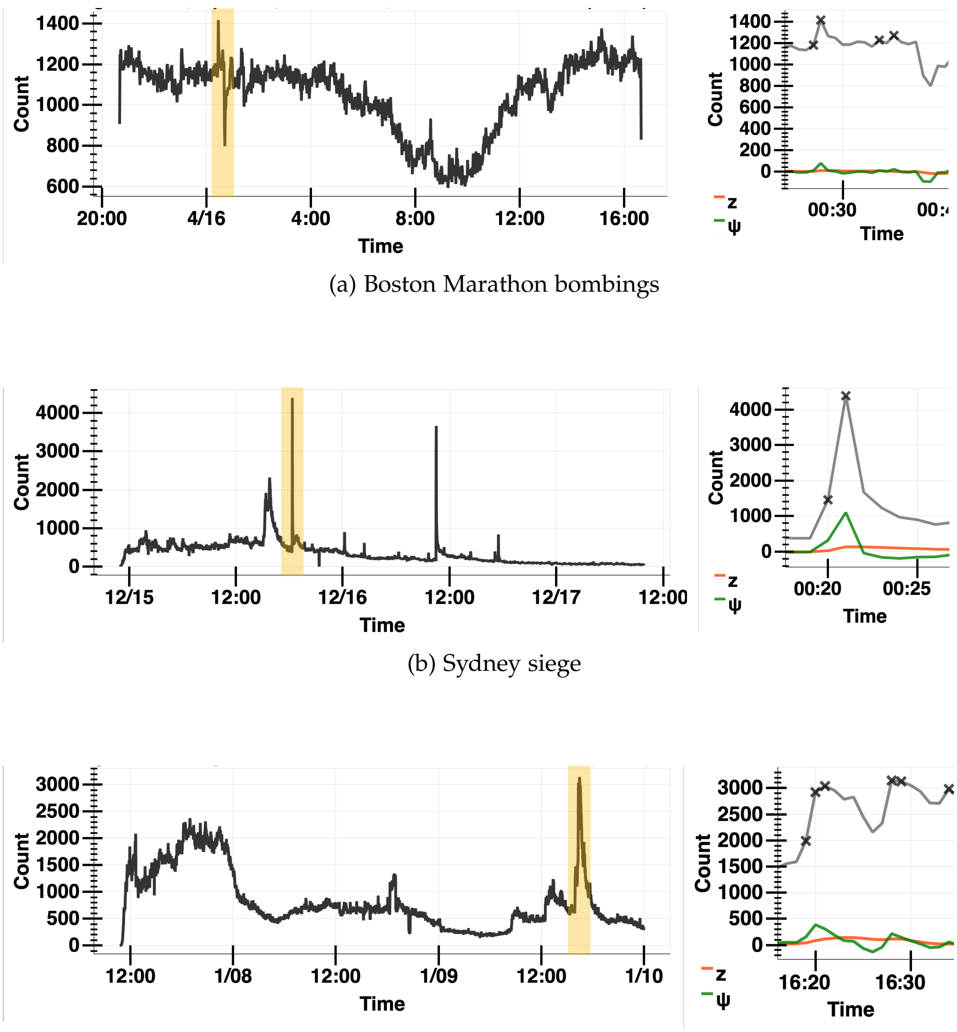
Algorithm 2 Identification of key bursts.

```
Input: Time series $\mathcal{D} = \{\boldsymbol{x}_i\}$ ($|\mathcal{D}| = N, \boldsymbol{x}_i = (c_i, s_i, a_i, p_i, \theta_i, \psi_i)$)
       Pre-filtering factor $\gamma$
Output: $y_k \in \{0, 1\}$
1:  $y_1 \leftarrow 0$
2:  for $i = 2$ to $N$ do
3:      $y_i \leftarrow 0$
4:      if $c_i > \gamma * a_i$ then
5:          if $y_{i-1} == 1$ then
6:              if $s_i \geq \theta_i$ or $c_i \geq c_{i-1}$ then
7:                  $y_i \leftarrow 1$
8:          else if $y_{i-1} == 0$ and $s_i > \theta_i$ then
9:              if $\psi_i > z_i$ then
10:                 $y_i \leftarrow 1$
```

related to smoothed frequency and the raw number of tweets (see Line 6) are examined. Specifically, given $y_{i-1} = 1$, if smoothed frequency at time i is greater than or equal to a threshold (i.e. $s_i > \theta_i$) or the number of tweets has increased at time i (i.e. $c_i \geq c_{i-1}$), time i is classified as a key burst. When $y_{i-1} = 0$, stricter constraints are required as minor fluctuations should not be detected as key bursts. If $y_{i-1} = 0$, whether the smoothed frequency at time i is above a threshold (i.e. $s_i > \theta_i$; Line 8) is checked.

In the next step, two difference-based features are compared unlike Algorithm 1. There is an interesting relationship between ψ and z. The difference between two values is small at minor fluctuations in time series plots. At peaks and in rising patterns in time series plots, however, ψ is greater than z. In contrast, z tends to be greater than ψ in falling lines. Figure 5.2 illustrates the relationship observed in three events showing different spreading patterns. Firstly, the "bostonbombings" event used to fine-tune the algorithm exhibits constantly bursty patterns. As can be seen in the close-up figure (right), no significant spike is observed. At two bursts around $00:30$ a.m., ψ is greater than z. The opposite is observed at a trough appearing after four bursts. Secondly, the "sydneysiege" displays a few large spikes separated by low-frequency periods. In the close-up figure, the difference between z and ψ is more significant. Finally, the "charliehebdo" tend show periods of *gradual* bursts compared with the "sydneysiege". The spike marked in yellow is gradual (i.e. smaller gradients and taking more time to reach the highest point) compared with the one in the "sydneysiege" graph. However, it can be observed that the aforementioned relationship between z and ψ is shown. During a small dip around $16:26$ p.m. in the close-up figure, z is greater than ψ. Based on these observations, $y_i = 1$ if ψ is greater than z.

Algorithm 2 is the final method for burst detection and used in the experiments of this chapter. In the experiments, the proposed method aims to detect about 25% of the total number of time windows in event time series as key bursts. Experimental results show that the majority of the most popular 10 rumours of 5 real-world events in the *PHEME (6392078)* (Kochkina et al., 2018a; Zubiaga et al., 2016a) can be identified by only analysing part of an entire event diffusion (Section 5.5.2). As this chapter conducts a thorough



(a) Boston Marathon bombings

(b) Sydney siege

(c) Charlie Hebdo shooting

Figure 5.2: Visualisation of the relationship between z (orange) and ψ (green) for three events. The figures on the left show the entire event evolution, and those on the rights show the periods coloured in yellow on the left-hand figures. × indicates bursts.

analysis of parameters (Section 5.4.3), end users can optimise the proposed method (parameters) according to their interests and objectives.

5.3.3 *Burst Summarisation*

Given detected key bursts, a methodology for automatically extracting the top N representative tweets that can summarise unfolding events within each burst is proposed. One simple approach to extract summaries is to assign scores to words in each tweet using a frequency or term frequency-inverse document frequency. However, this approach generally produces poor results. This section proposes three summarisation methods based on graph-based keyword extraction algorithms (*TextRank* (Mihalcea and Tarau, 2004), *K-core* (Seidman, 1983b), and *Dens* (Tixier et al., 2016)). There are two main reasons for deciding to use these graph-based methods: 1) they are fully unsupervised.

The performance of supervised text ranking models is highly dependent on training data. In contrast, graph-based methods are capable of extracting summaries solely based on the text itself (Meladianos et al., 2015). This is particularly important for breaking news events as they are usually unseen and evolve unexpectedly. 2) they can be adapted to short texts, which are required for summarisation methods for tweets.

One common limitation of graph-based methods is they tend to favour long tweets which contain commonly used words or hashtags rather than informative tweets (Alsaedi et al., 2016b). It is highly likely that tweets irrelevant to a target event and less important are ranked in the top N summaries. Techniques used in the proposed framework to avoid this issue include: 1) using inverse document frequency (IDF; Robertson (2004); see Section 5.3.3.4), and 2) using normalised edge weights in graphs of words (Meladianos et al. (2015) and Meladianos et al. (2018b); Section 5.3.3.2).

5.3.3.1 *Data Preprocessing*

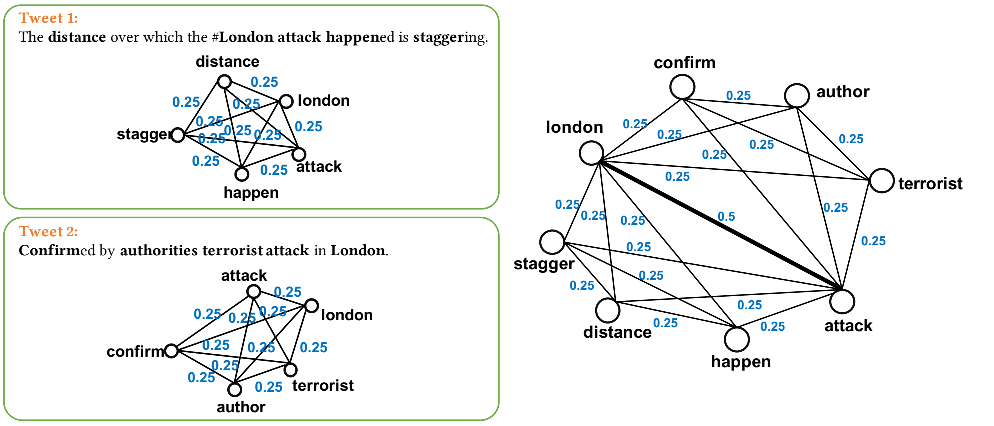
All tweets are lowercased. Retweet symbols (i.e. "rt @"), URLs, user mentions, and special characters (e.g. !, ?, # etc.) are removed. Embedded links can provide useful information regarding the virality and trustworthiness of tweets (Tanaka et al., 2014; Gupta and Kumaraguru, 2012; Castillo et al., 2011; Friggeri et al., 2014), but they tend not to appear at the very early stages of event diffusion (e.g. during the first burst; Maddock et al. (2015)). Rather, people are more likely to repost what they observed without additional information in the early stages (Maddock et al., 2015; Zhao et al., 2015; Chen et al., 2018). As this chapter aims to identify potential rumour tweets in the early stages of event diffusion, semantic and syntactic information is exploited. All tweets are normalised. Normalisation methods include removing English stopwords and lemmatisation using "WordNetLemmatizer" in a Python library NLTK [2]. Tweets with a minimum of 4 tokens are considered as tweets which lack enough textual features are generally unremarkable and add noise to data (Ifrim et al., 2014).

5.3.3.2 *Graphical Representation of Tweets*

In order to employ graphical keyword ranking algorithms for event summarisation, an input tweet corpus at each time window should be represented as graphs. In the proposed framework, the graphical representation of tweets (i.e. *graphs-of-words*) proposed by Meladianos et al. (2015) is adopted.

Let $G = (V, E)$ be a weighted undirected graph with the set of vertices V and that of edges E. Let D be a document which is equivalent to a set of preprocessed tweets at a resampled time window. Compared to other traditional documents such as news articles, tweets contain multiple languages, ungrammartical phrases and sentences, and unofficial abbreviations. Therefore, only uni-grams are used as vertices to minimise the growth of the graph size (Mihalcea and Tarau, 2004). Two vertices are connected if they co-occur within a tweet. Each co-occurrence is equally important. If two vertices V_i and V_j are connected, a weight $w_{ij} = \frac{1}{p-1}$ is added to the edge between V_i and V_j, where p is the number of unique terms in the tweet. This

[2] available via https://www.nltk.org



Tweet 1:
The **distance** over which the #**London attack happen**ed is **stagger**ing.

Tweet 2:
Confirmed by **authorities terrorist attack** in **London**.

Figure 5.3: Visualisation of the graph-of-words representations (Meladianos et al., 2015) of two example tweets (left) and a single graph generated from them (right).

guarantees that each node in the tweet has degree 1. This allows avoiding assigning higher degrees to nodes in long tweets. The entire procedure is recursively performed over a collection of tweets posted at each time interval. Edge weights are accumulated as iterations continue.

In Figure 5.3, for example, consider the first tweet "The distance over which the #London attack happened is staggering." When representing it as a subgraph where each node has degree 1, each edge has a weight of $\frac{1}{4} = 0.25$ because there are 5 unique tokens after normalisation (see Section 5.3.3.1). Given the first subgraph, consider another tweet "Confirmed by authorities terrorist attack in London." Each edge in the subgraph for the second tweet has a weight of 0.25. When the subgraph for each tweet is merged with the graph-of-words (i.e. the output graph representing every input tweet), the weight of an edge in the latter is incrementally increased by the weight in the former if the edge exists in the latter. In the example, the *graph-of-words* for the two tweets on the right shows that the weight of the edge between "london" and "attack" is increased from 0.25 to $0.25 + 0.25 = 0.5$ because this pair appears in both tweets.

5.3.3.3 *Term Weighting Approaches*

This section describes three graph-based keyword extraction algorithms: *TextRank* (Mihalcea and Tarau, 2004), *K-core* (Seidman, 1983b), and *Dens* (Tixier et al., 2016). They are used to compute weights for individual words, which will be used to compute rank tweets in Section 5.3.3.4.

- **TextRank (Mihalcea and Tarau, 2004)** TextRank is a graph-based method for ranking keywords in texts. The TextRank score of a vertex V_i is defined by

$$WS(V_i) = (1-d) + d * \sum_{V_j \in In(V_i)} \frac{w_{ji}}{\sum\limits_{V_k \in Out(V_j)} w_{jk}} WS(V_j),$$

where d is a damping factor that ranges between 0 and 1, which enables the algorithm to pick a vertex in the graph at random. It prevents the algorithm from reducing the influence of certain vertices in the graphs excessively. The parameter d is set to 0.85 as proposed in the original work. The obtained scores of vertices are sorted in descending order. The TextRank is implemented using source code released by Barrios et al. (2016).

- **K-cores (Seidman, 1983a)** K-cores of a graph G refer to the maximal subgraph where the degrees of all vertices is at least k (Seidman, 1983a). In other words, this method returns weights for keywords in a subgraph rather than all keywords appearing in input. K-cores with the largest core number represent the most cohesive subregions of a graph. Therefore, the use of k-cores for ranking keywords allows identifying *influential keywords* from a collection of noisy tweets (Tixier et al., 2016). K-cores of a graph can be obtained by recursively removing all vertices of degree equal to or less than k and their adjacent edges from the graph until the degrees of all vertices in the remaining graph is larger than k (Batagelj and Zaversnik, 2003). The k is incrementally increased until no node remains in the graph. Note that the removal of vertices and edges is conceptual to describe that a visited (i.e. removed) node and its adjacent edges are not considered in computation at the next step. They are not physically removed from a graph. This process is referred to as *k-core decomposition*. This chapter follows the implementation proposed in (Saríyüce et al., 2013). Algorithm 3 is a pseudo-code of the k-core decomposition for unweighted and undirected graphs. Therefore, the degree of a vertex is defined as the number of its adjacent edges and $w_{u,v}$ in Line 6 is always 1 (i.e. the degrees of each visited vertex's neighbours are decreased by one). In weighted undirected graphs-of-words, $w_{u,v}$ in Line 6 in Algorithm 3 is the edge weight between nodes u and v. The benchmark method (Meladianos et al., 2015) does not perform k-core decomposition, which requires a recursive pruning process.

- **Dens (Tixier et al., 2016)** This method is based on K-cores described above. Given the k-core decomposition output of a graph-of-words $G = (V, E)$, this method outputs the best core-number for selecting a sub-graph of G. Algorithm 4 describes the computation of the method. The ***density*** function is defined by

$$density(G) = \frac{|E|}{|V|(|V|-1)}.$$

  The *levels* is a set of unique core numbers of G sorted in descending order. The ***elbow*** finds the farthest point from the line connecting the first and last point of a curve in which the x-axis is core number and y-axis is density. If all points are on the same line, the main core (i.e. the largest core number) is returned. When a graph has only two core numbers, one with the highest density score is returned. Figure 5.4 shows example visualisation of the Dens. Given the best core number k_{best} obtained via the Dens, keywords in the k_{best} subgraph are selected.

As source codes for keyword detection algorithms are not publicly available, these methods are reproduced based on descriptions and implemented using Python.

5.3.3.4 *Summarisation Method*

The input of the proposed summarisation methods is a collection of tweets that were posted during a key burst. The first step is to generate a graph-of-words (see Section 5.3.3.2). Note that only uni-grams are considered when

Algorithm 3 k-core decomposition (Saríyüce et al., 2013)

```
Input: $G$: weighted undirected graph-of-words,
       $\delta(v)$: the degree of vertex $v \in V$, $e \in E$: edge,
       $w_{u,v}$: the weight of edge connecting vertices $u$ and $v$,
Output: $K(v)$: $k$-cores of vertices in $G$
Order the set of vertices $v \in V$ in ascending order of $\delta(v)$
for each $v \in V$ do
    $K(v) \leftarrow \delta(v)$
    for each $(u, v) \in E$ do
        if $\delta(u) > \delta(v)$ then
            $\delta(u) \leftarrow \delta(u) - w_{u,v}$
    Reorder unvisited vertices($v'$) in ascending order of $\delta(v')$
```

Algorithm 4 *Dens* (Tixier et al., 2016)

```
Input: core decomposition of $G$
Output: set of keywords (i.e. nodes of $k_{\text{best}}$-core of $G$)
$D \leftarrow$ empty vector of length $n_{\text{levels}}$
for each $n \in n_{\text{levels}}$ do
    $D[n] \leftarrow$ density(levels[$n$]-core)
$k_{\text{best}} \leftarrow$ levels[elbow($n, D[n]$)]
```

generating graphs-of-words to minimise the growth of the size of graphs-of-words and data sparsity (as the length of n-grams increases, the frequency of any given n-gram decreases; Mihalcea and Tarau (2004)). Next, weights for all words (i.e. uni-grams) appearing in the corpus are computed using a term weighting method (see Section 5.3.3.3). Given weights of uni-grams, weights for n-grams can be defined as the average score of all uni-grams appearing in an n-gram. This step is optional. This chapter conducts an experiment to study whether the use of multi-word keywords helps to extract more meaningful summary tweets by capturing more context in short tweets (see details in Section 5.5.2). Another option is the use of inverse document frequency (IDF). Specifically, the score of each n-gram is multiplied by its IDF value. In general, IDF is employed in frequency-based term weighting methods to diminish the impact of terms that appear very frequently in a

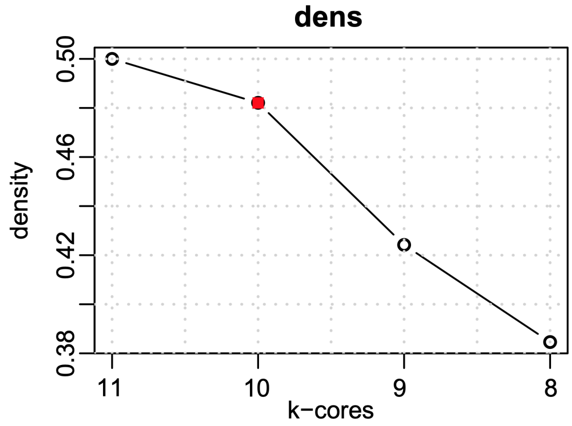


Figure 5.4: Visualisation of the computation of the Dens (Tixier et al., 2016). The red point is the elbow of the curve.

corpus. Even after removing stopwords, some domain- and event-specific words are often observed. For example, hashtags starting with "prayfor" are very popular worldwide and appear when tragic events (e.g. terrorist attacks, natural distastes, etc.) take place. Such words are usually not related to potential rumours. This chapter conducts experiments to study whether the incorporation of IDF into graph-based term weighting methods helps to identify tweets more related to rumours. N-gram scores are sorted in descending order, and the top $P\%$ terms are selected and used to compute tweet scores. This chapter presents summarisation results with different P values in Section 5.5.2.

Given scores of n-grams, each tweet is represented as a vector of scores of n-grams constituting the tweet. Subsequently, l^2-normalisation is applied. Let the vector representation of a tweet be $v = [w_1, w_2, \cdots, w_n]$, where w_i denotes a word weight. A normalised representation v' is defined by

$$v' = \frac{v}{\sqrt{w_1^2 + w_2^2 + \cdots + w_n^2}} = [w'_1, w'_2, \cdots, w'_n],$$

The final score of each tweet $S(v')$ is defined as the sum of normalised scores of n-grams. Formally,

$$S(v') = w'_1 + w'_2 + \cdots + w'_n.$$

This approach overcomes a common limitation of graph-based summarisation: preference for longer tweets. Some work claims that longer messages tend to be more descriptive than shorter messages do (Meladianos et al., 2015). However, some linguistic signals such as "reportedly" and "really?" can be useful to characterise rumours (Zhao et al., 2015). This chapter argues that short tweets can be as important as long tweets in the context of rumour detection because these linguistic signals can appear in short tweets. In addition, long tweets may contain hashtags which make them pointless and irrelevant to topics of interest (Duan et al., 2012). Tweets in a corpus of a key burst are sorted in descending order, and the top N tweets are selected as summaries of the burst.

5.4 EXPERIMENTS

5.4.1 *Data*

Data sets for six events in the *Twitter event data (2012-2016*; Section 4.2.4.1) and *Aug-PHEME* (Section 4.6.2.1) are used in the experiments. A time series of each event is plotted using the *Twitter event data (2012-2016)* and a burst detection model takes it as input. Given identified key bursts, a summarisation method is applied to a collection of tweets at each burst. The top N summary tweets are evaluated using weakly annotated tweets in the *Aug-PHEME*. Section 5.4.5 describes details about evaluation approaches.

As the task of identifying potential rumours is proposed as a preliminary task for "early" rumour detection which will be detailed in Chapter 6, this chapter investigates model performance during the initial stages of the evolution of each event. In specific, this chapter only considers the first seven days for the "fergusonunrest" and the first three days of the other four events. The

Table 5.1: Statistics of the Twitter events (2012-2016) data after temporal filtering and resampling.

Event	Start	End	Number of time windows	Number of tweets
charliehebdo	10:48 7/1/2015	00:00 10/1/2015	3,673	3,037,338
fergusonunrest	23:45 9/8/2014	23:45 16/8/2014	10,081	2,415,222
germanwings	10:34 24/3/2015	00:00 27/3/2015	3,687	493,003
ottawashooting	13:55 22/10/2014	8:22 24/10/2014	2,548	669,572
sydneysiege	23:00 14/12/2014	9:51 17/12/2014	3,532	1,210,252
bostonbombings	20:39 15/04/2013	16:39 16/04/2013	1,201	1,259,857

reason for using varying time windows is that the "fergusonunrest" lasted for a few weeks compared to the others which lasted for a couple of days (see Section 2.4 and Table 4.2). A time series of each event is resampled using a 1-minute window following the practice of previous studies (Zubiaga et al., 2012; Nichols et al., 2012). Table 5.1 shows statistics of the data after temporal filtering and resampling. *Start* and *end* refer to the first and last timestamps of filtered time series.

5.4.2 *Preliminary Experiment and Results*

A hypothesis of the research in this chapter is that it is likely that the same topic can be persistently discussed even after its popularity reaches a peak, and therefore, major emerging newsworthy stories and rumours can be detected by analysing messages generated during time windows lying in increasing lines in event evolution graphs. This section describes a preliminary experiment undertaken to prove the hypothesis and its results. Specifically, when rumours are detected *for the first time* is analysed. Table 5.2 shows the number of rumours detected for the first time and whether they are detected from bursts or decaying lines. Here, rumour source tweets in the augmented data (see Section 4.6.2) are used as the ground truth. Note that it does not mean that rumours appearing in decaying lines cannot be detected from bursts, and vice versa. Most rumours in the augmented data (see Table 4.8) appear in both bursts and decaying lines. This section's experiment focuses on the very first detection of rumours. For all but the "fergusonunrest", rumours are first seen in key bursts rather than persistently decreasing lines in event evolution. This confirms the proposed hypothesis.

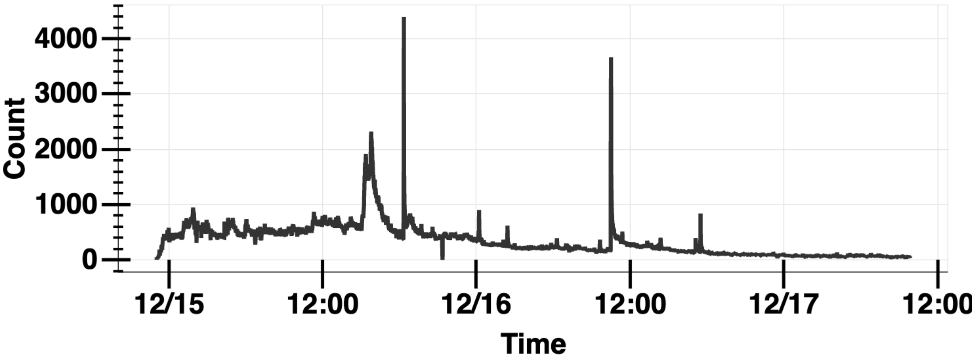
5.4.3 *Parameter Selection for Key Burst Detection*

A grid search is performed to select optimum sets of parameters of the proposed key burst detection algorithm. The main purpose of parameter selection introduced in this chapter is to analyse correlations between each individual parameter and the number and quality of detected bursts, rather than perfectly optimising hyperparameters of the proposed method. Future work should incorporate automatic hyperparameter optimisation. The parameter space is defined as follows: {0.001, 0.0025, 0.005, 0.01, 0.02, 0.025, 0.05, 0.1} for a pre-filtering parameter *gamma* (γ) (see Line 4 in Algorithm

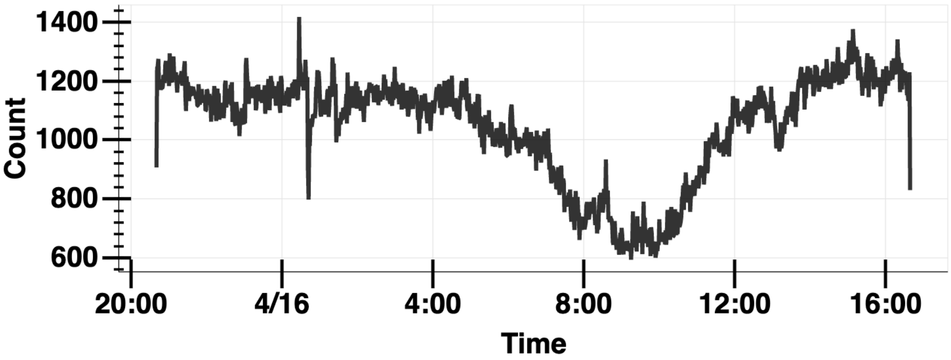
Table 5.2: Results of a preliminary experiment for confirming the hypothesis. The number of rumours first seen in bursts and that first seen decaying lines are shown for six events.

Event	Number of rumours in *bursts*	Number of rumours in *decay*
bostonbombings	**16**	7
charliehebdo	**16**	13
fergusonunrest	6	**7**
germanwings	**6**	2
ottawashooting	**14**	5
sydneysiege	**18**	7
Total	**76**	41

2); {0.01, 0.1, 0.3, 0.5, 0.9, 0.95} for a smoothing factor *alpha (α)* for computing EWMMean s_i, and {0.0001, 0.001, 0.01, 0.1, 0.5, 0.9} for a weight factor *lambda (λ)* for computing ψ (see Section 5.3.2.1).



(a) Time series plot of the “sydneysiege” event.



(b) Time series plot of the “bostonbombings” event.

Figure 5.5: Time series plots of the “sydneysiege” and “bostonbombings” events.

Two data sets–“sydneysiege” and “bostonbombings”– are used to select parameters. It is worth noting that propagation patterns of the two events are very different (Figure 5.5). The time series plot for the “sydneysiege” exhibits a few huge spikes between periods of low frequency, while that for the “bostonbombings” tend to constantly exhibit high volume.

First of all, detected bursts for the two events for different combinations of parameters are visualised. For each data set, parameter sets that detect 20-30% of the total number of time windows are selected. As described in

Section 5.2, parameter sets that 1) identify instances located in a persistently falling line, and 2) miss significant peaks are manually excluded.

As for the effect of each parameter on the number of detected bursts, The γ values between 0.001 and 0.05 detect the same number of bursts. For the γ values greater than 0.05, the number of bursts decreased as γ increases. The λ values between 0.0001 and 0.1 detect the same number of bursts for both events. For the λ greater than 0.1, the larger λ is, the fewer time windows are detected as key bursts. When λ increases to 0.9, the number of detected key bursts significantly decreases (around by 40% compared with $\lambda = 0.5$ in the case of the "sydneysiege" event). On the other hand, the larger α is, the more time windows are identified.

Table 5.3 shows 6 combinations of parameters after manual inspection and removing parameter sets producing duplicated results. For deduplication, sets of the smallest parameters are left. For example, $\{\gamma = 0.1, \alpha = 0.3, \lambda = 0.0001\}$ and $\{\gamma = 0.1, \alpha = 0.3, \lambda = 0.001\}$ generate the same results. In that case, only the first set of parameters is shown in Table 5.3 as λ of the former is smaller than that of the latter. Percentages in parentheses are the proportion of the detected bursts to the total number of time windows in the time series of each event. It should be noted that differences in the number of key bursts do not have a great impact on patterns of key bursts with the parameters shown in the table. In other words, significant key bursts (e.g. large spikes and bursts followed by them) are already detected by using the parameters detecting the smallest number of bursts in the Table 5.3, i.e. $\{\gamma = 0.1, \alpha = 0.3$, and $\lambda = 0.0001\}$. For instance, for the "sydneysiege", the set $\{\gamma = 0.001, \alpha = 0.5, \lambda = 0.0001\}$ detects 157 (i.e. $1001 - 844$) more bursts than the set $\{\gamma = 0.1, \alpha = 0.3, \lambda = 0.5\}$ does. However, these additional bursts are relatively small bursts. Therefore, end users can start with the parameter set $\{\gamma = 0.1, \alpha = 0.3, \lambda = 0.5\}$ or $\{\gamma = 0.1, \alpha = 0.5, \lambda = 0.0001\}$ and tweak the parameters according to their objectives.

gamma(γ)	**alpha(α)**	**lambda(λ)**	**Number of bursts**	
			sydneysiege	**bostonbombings**
0.1	0.3	0.0001	847(24%)	329(27%)
0.1	0.3	0.5	844(24%)	330(28%)
0.001	0.3	0.0001	867(25%)	340(28%)
0.1	0.5	0.0001	899(25%)	340(28%)
0.1	0.9	0.5	970(27%)	366(30%)
0.001	0.5	0.0001	1001(28%)	363(30%)

Table 5.3: Results of parameter selection. Percentages in parentheses are the proportion of the detected bursts to the total number of time windows in the time series of each event.

5.4.4 *Baselines*

5.4.4.1 *Baselines for Burst Detection*

Two SOTA models for key burst detection in event evolution on social media are used to evaluate the method proposed in this thesis. Let t_i be the i^{th} time window in a time series $S = \{(t_i, c_i)\}$, where $i = 1, 2, \cdots, N$ and c_i is the number of tweets at t_i.

- **Peng et al. (2018)**: *Emerging Score* at t_i, denoted by $ES(t_i, c_i)$, is defined by
$$ES(t_i, c_i) = \frac{c_i - EWMA(c_1, c_2, \cdots, c_{i-1})}{1 - EWMStd(c_1, c_2, \cdots, c_{i-1})},$$
where $EWMA$ is exponentially weighted moving average and $EWMStd$ is exponentially weighted standard deviation. Time window t_i is classified as a key burst if $ES(t_i, c_i) > \textit{threshold}$. The authors did not specify how to set two parameters used in the methods. In the experiments of this thesis, smoothing factor α_{peng} for computing $EWMA$ is randomly sampled. A threshold θ_{peng} is randomly sampled from a uniform distribution. In the experiments, parameters that produce results similar to the proposed method are chosen for comparison. To this end, more than $1,000$ iterations for parameter sampling for each input event data set were performed.

- **Gillani et al. (2017)**: Given a local maximum at t_i which satisfies $c_{i-1} < c_i > c_{i+1}$, time window t_i is identified as a key burst if $c_i > \mu + \sigma$. μ and σ are the mean and standard deviation of the number of tweets of all time windows in input time series S, respectively. Unlike the proposed method and Peng et al. (2018), this method does have no parameter. As described in Section 5.2, methods for detecting peaks are not suitable for identifying rumours in the early stages of their diffusion as they require future observations to annotate the current time window. Despite incompatibility, the incorporation of this method into evaluation will help to further understand and confirm why *peaks* by themselves are not appropriate signal for key bursts, especially in the context of ERD.

As source code and data sets for the studies above are not publicly available, these methods are reproduced based on descriptions and implemented using Python.

5.4.4.2 *Baselines for Summarisation*

This section introduces one graph-based and two frequency-based baselines.

- **Meladianos et al. (2018a)**: This method is a graph-based method for extractive summarisation. Given a graph-of-words G_t described in Section 5.3.3.2 and a set of tweets $\mathcal{D}_t$ at time window t, a non-decreasing submodular function $f(\mathcal{S})$, where $\mathcal{S} \subseteq \mathcal{D}_t$, is defined as the sum of the weights of edges of G_t which connect all pairs of words of tweets in $\mathcal{S}$. The method extracts representative tweets from $\mathcal{D}_t$ by maximising f given a cardinality constraint using a greedy algorithm. Source code is publicly available via https://bitbucket.org/ksipos/optimization-sub-event-detection/src.

- **Hybrid TF-IDF (Sharifi et al., 2013)**: Hybrid TF-IDF is a frequency-based method for generative summarisation. Given a set of tweets at time window t, a weight of a tweet $W(S)$is defined by

  $$W(S) = \frac{\sum_{i=0}^{NumWords} W(w_i)}{nf(S)},$$

  where

  $$\begin{aligned} W(w_i) &= tf(w_i) * \log_2(idf(w_i)), \\ tf(w_i) &= \frac{NumOccurrencesOfWordInSet}{NumWordsInSet}, \\ idf(w_i) &= \frac{NumTweetsInSet}{NumTweetsInWhichWordOccurs}, \\ nf(S) &= \max\{MinimumThreshold, NumWordsInTweet\}. \end{aligned}$$

  The *MinimumThreshold* is the desired number of words in a summary. Following the original work, this is set to 10. After computing $W(S)$ for all tweets posted during the time window t, tweets are sorted in descending order of weights.

- **SumBasic (Nenkova and Vanderwende, 2005)** SumBasic is a frequency-based method for extractive summarisation which uses probability distributions of words. Given a set of tweets at time window t, a weight of a tweet $W(S)$ is defined by

  $$W(S) = \sum_{w_i \in S} \frac{p(w_i)}{|\{w_i | w_i \in S\}|}$$

  where

  $$p(w_i) = \frac{NumOccurrencesOfWordInTweet}{NumWordsInSet}.$$

The tweet which contains the word with the highest word probability $p(w_i)$ and has the best tweet weight (i.e. $W(S)$) is included in a summary. For each word in the selected tweet, its probability is updated with a new probability defined by $p(w_i) * p(w_i)$. $W(S)$ for all tweets are computed based on the updated word probability distributions. Subsequently, the same procedure is repeated until the desired length (e.g. 10 sentences) of the summary is reached.

5.4.5 *Evaluation Approaches*

5.4.5.1 *Key Burst Detection*

Identifying highlights of rumour evolution is infeasible. Therefore, there is no standard approach for evaluating the proposed method in the context of ERD. This thesis first performs a comparative analysis of different burst detection methods based on patterns of detected bursts and the behaviour of parameters. It also proposes a novel evaluation approach suitable for the early identification of potential rumours. It uses weak labels obtained via data augmentation proposed in Chapter 4. They cannot be used as ground truth for the task of ERD which requires high precision and recall. However, they are enough to identify tweets which bear newsworthy stories or rumours. Details are described in the next section.

Table 5.4: The number of available references and that of references which have greater than or equal to 50 related tweets in the augmented data.

Event	Number of references	Number of references ($>$ 50)
charliehebdo	53	33
fergusonunrest	24	10
germanwings	13	7
ottawashooting	40	23
sydneysiege	49	26

5.4.5.2 *Burst Summarisation*

No standard method is available for the evaluation of automatic summarisation methods (Sharifi et al., 2013). In general, two approaches are used to evaluate summarisation methods (Sharifi et al., 2013): 1) evaluating results in terms of predefined metrics such as grammaticality, content, and readability. As can be seen in the literature review in Section 2.3.2.2, ROUGE (Lin, 2004) is commonly used when manually built ground truth is available. 2) measuring the applicability of results to different tasks. This chapter employs the second evaluation approach. Specifically, it evaluates summarisation methods based on how many potential rumours can be captured using the top N summaries. The augmented data before balancing and merging with the benchmark data is used as ground truth in evaluation (see Table 4.9 for detailed statistics). Table 5.4 shows the total number of available reference rumours and the number of references which have more than 50 related tweets in the augmented data (see Section 4.6.2).

For each key burst, the top N summary tweets obtained via a summarisation method are compared with rumour tweets in the ground truth. Note that detectable references vary between different key bursts. To consider textual variations (e.g. the addition of URLs and pictures, mentions, and retweets) of ground truth rumours, pairs of a summary tweet and rumour with and without preprocessing (see Section 5.3.3.1) are compared. If the tweet id, raw text, or processed text of a summary tweet matches with that of any ground truth rumours, the summary is a *potential rumour*. Example 5.1 illustrates an evaluation approach in more detail. User mentions are replaced with "[username]" to ensure the confidentiality of personally identifiable information. Similarly, URLs are replaced with "[LINK]". This example burst contains tweets relevant to reference #0, 3, and 43. "Seed tweets" are unique tweets weakly labelled as a corresponding reference (i.e. the output of Chapter 4). For instance, there are three seed tweets annotated as the Ref. 0 via data augmentation. Their textual variations in the burst are identified (i.e. "Total umber of relevant tweets"). All summary tweets which can be found in the set of relevant tweets are returned. As there are 7 summary tweets related to the Ref. 0, it is considered that this burst identifies a potential rumour related to the Ref. 0. For a reference which is identified over several time windows, only the first timestamp is recorded. This timestamp will be used to evaluate different burst detection methods in the context of the *early* detection of potential rumours.

As illustrated in Section 5.3.3, representative tweets are extracted from each key burst for the five selected events in the Twitter events 2012-2016 data. Bursts detected using the parameters $\gamma = 0.1, \alpha = 0.3$, and $\lambda = 0.0001$, which detect 24.4% of the total number of time windows on average, are used for summarisation. End users can control the parameters according to their interests and objectives based on the analysis in Section 5.4.3.

Example 5.1:

References detectable in this burst: 0, 3, 43

Ref. 0

- **Seed tweets**
  - ARMED MAN TAKES HOSTAGE IN KOSHER GROCERY IN PARIS – AFP
  - AFP reports shooting in eastern #Paris where an armed man has taken hostage in a kosher shop.
  - @YahooNewsUK: BREAKING: 'Armed hostage crisis' in a Kosher grocery store in Vincennes, east of Paris: [LINK]
- **Total umber of relevant tweets:** 254
- **Matched tweets in the top N summaries:** 7
  - RT @[username]: #BREAKING Armed man takes hostage in kosher grocery in Paris: source
  - "@[username]: #BREAKING Armed man takes hostage in kosher grocery in #Paris: source"
  - RT @[username]: ARMED MAN TAKES HOSTAGE IN KOSHER GROCERY IN PARIS – AFP
  - ARMED MAN TAKES HOSTAGE IN KOSHER GROCERY IN PARIS - AFP
  - RT @[username]: BREAKING :Armed man takes hostage in kosher grocery in Paris
  - *ARMED MAN TAKES HOSTAGE IN KOSHER GROCERY IN PARIS: AFP
  - RT @[username]: *ARMED MAN TAKES HOSTAGE IN KOSHER GROCERY IN PARIS: AFP

Ref. 3

- **Seed tweets**
  - New post: Charlie Hebdo killers seize HOSTAGE and are holed up in business premises near Paris airport as they tell [LINK]
- **Total umber of relevant tweets:**1
- **Matched tweets in the top N summaries:** 0

Ref. 43

- **Seed tweets**
  - RT @[username]: UPDATE: #CharlieHebdo Suspects in contact with police, say they want to die as martyrs; have at least 1 hostage. #wcvb
  - RT @[username]: CNN zegt: Charlie Hebdo suspects tell police they want to die as martyrs [LINK]
- **Total umber of relevant tweets:** 3
- **Matched tweets in the top N summaries:** 0

5.5 RESULTS AND DISCUSSIONS

5.5.1 *Evaluation of Patterns of Detected Bursts*

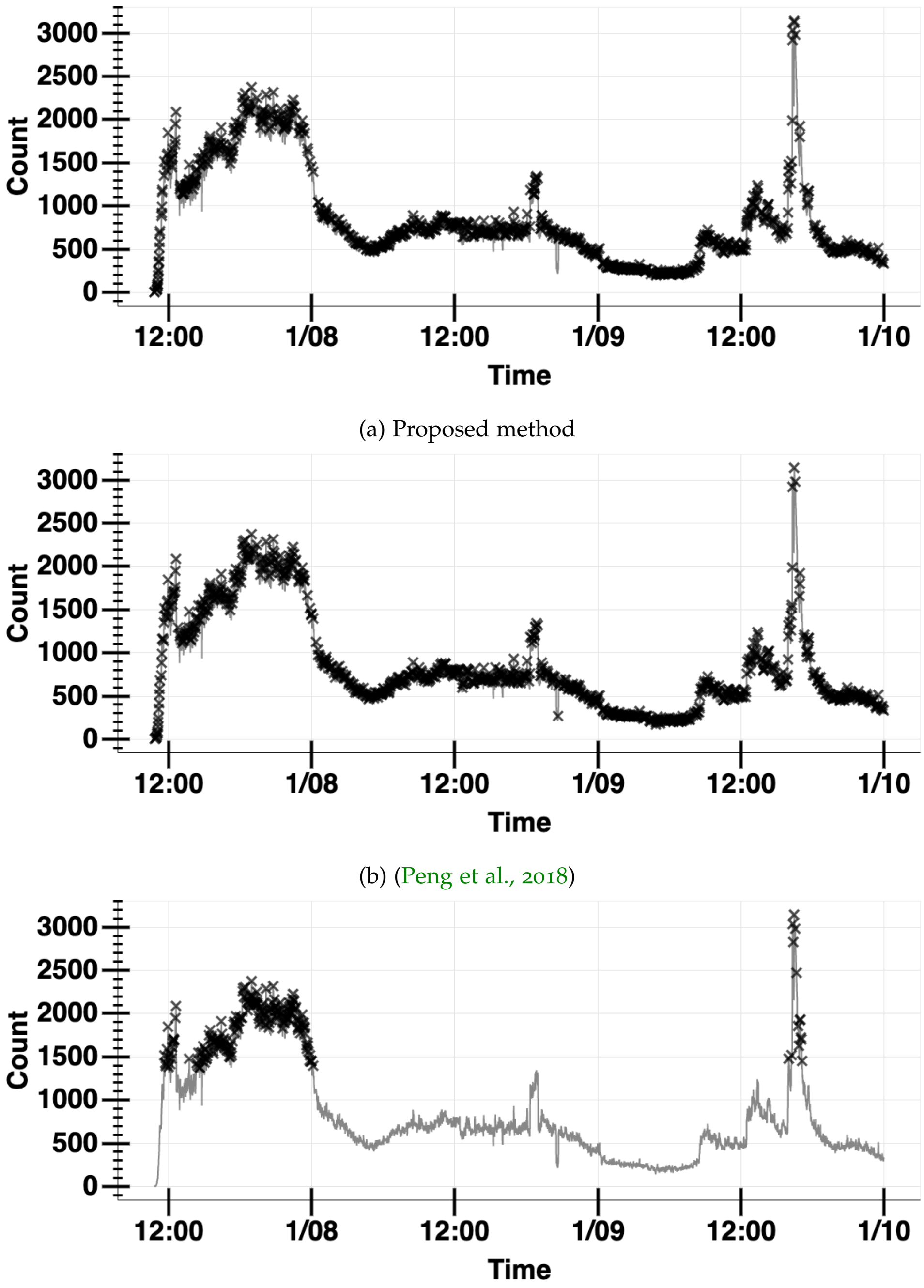
This section compares the proposed methods with the two baselines based on patterns of detected bursts. Table 5.5 shows the number of detected bursts and parameters. For (Peng et al., 2018), the number of bursts detected using parameters, which produce results similar to the proposed method, is shown. Note that Peng's parameters were randomly sampled over 1,000 iterations for this purpose (see Section 5.4.4.1) As for (Gillani et al., 2017), their method does not have controllable parameter as it aims to detect significant maxima rather than emerging trends. Therefore, it detects much fewer time windows compared with the proposed method. The visualisation of the results are shown in Figure 5.6, 5.7, 5.8, 5.9, and 5.10. It seems that all methods detect time windows located in persistently falling lines. This is because the original graphs had to be scaled down. Relatively small peaks are not clearly visible in the figures. For example, Figure 5.13a shows a close-up of the largest spike at 12 : 00 p.m. on 26 March in Figure 5.8 for the "germanwings" event. It looks like the number of tweets persistently decreases since 12 : 00 p.m. in Figure 5.8. It can be seen that there are small fluctuations after the large peak in the close-up figure.

The proposed method and (Peng et al., 2018) can exhibit similar patterns by adjusting parameters accordingly. This is understandable because both methods aim to identify bursts associated with emerging trends. An important advantage of the proposed method over (Peng et al., 2018) is that the former is robust to very different propagation patterns of different events. This chapter enabled this via a thorough analysis of the method's parameters (see Section 5.4.3). On the contrary, Peng's parameters are arbitrary and their impact on different events' diffusion patterns is not clearly explicable. In specific, with the proposed method, the number of key bursts does not have a great impact on the distribution of key bursts (see Section 5.4.3). When more bursts are identified by tweaking parameters, small bursts or peaks are additionally identified. This enables users to tweak the parameters without worrying about a drastic change in distributions of key bursts. However, the impact of (Peng et al., 2018)'s parameters on the distribution of key bursts tends to be inconsistent. Distributions of identified bursts vary according to parameters.

The experimental results show that (Gillani et al., 2017) is not appropriate for the task of the identification of potential rumours based on key bursts. Firstly, it ignores small but potentially significant bursts over a long period. In other words, it detects bursts at the local level and fails to explain the entire evolution of events. Another reason is that most detected time windows are lying on decreasing lines of time series plots from a global perspective. As shown in Section 5.4.2, emerging newsworthy stories and rumours tend to be detected from bursts lying on increasing trends.

Table 5.5: Burst detection results for the proposed method and SOTA methods. The number of bursts detected using the parameters $\gamma = 0.1, \alpha = 0.3$, and $\lambda = 0.0001$ (Proposed method). For (Peng et al., 2018), the number of bursts detected using parameters, which produce results similar to the proposed method, is shown. No controllable parameter is available for (Gillani et al., 2017).

Event		Proposed method	Peng et al. (2018)	Gillani et al. (2017)
charliehebdo	**count**	932 (25%)	930 (25%)	209(6%)
	params	$\gamma = 0.1$ $\alpha = 0.3$ $\lambda = 0.0001$	$\alpha_{\text{peng}} = 0.626$ $\theta_{peng} = 1.154$	—
fergusonunrest	**count**	$2,655$ (26%)	$2,655$ (26%)	304(3%)
	params	$\gamma = 0.1$ $\alpha = 0.3$ $\lambda = 0.0001$	$\alpha_{\text{peng}} = 0.113$ $\theta_{peng} = 0.571$	—
germanwings	**count**	847 (23%)	847 (23%)	126(3%)
	params	$\gamma = 0.1$ $\alpha = 0.3$ $\lambda = 0.0001$	$\alpha_{\text{peng}} = 0.128$ $\theta_{peng} = 0.542$	—
ottawashooting	**count**	623 (24%)	622 (24%)	77(3%)
	params	$\gamma = 0.1$ $\alpha = 0.3$ $\lambda = 0.0001$	$\alpha_{\text{peng}} = 0.479$ $\theta_{peng} = 0.827$	—
sydneysiege	**count**	847 (24%)	845 (24%)	136(4%)
	params	$\gamma = 0.1$ $\alpha = 0.3$ $\lambda = 0.0001$	$\alpha_{\text{peng}} = 0.627$ $\theta_{peng} = 1.148$	—

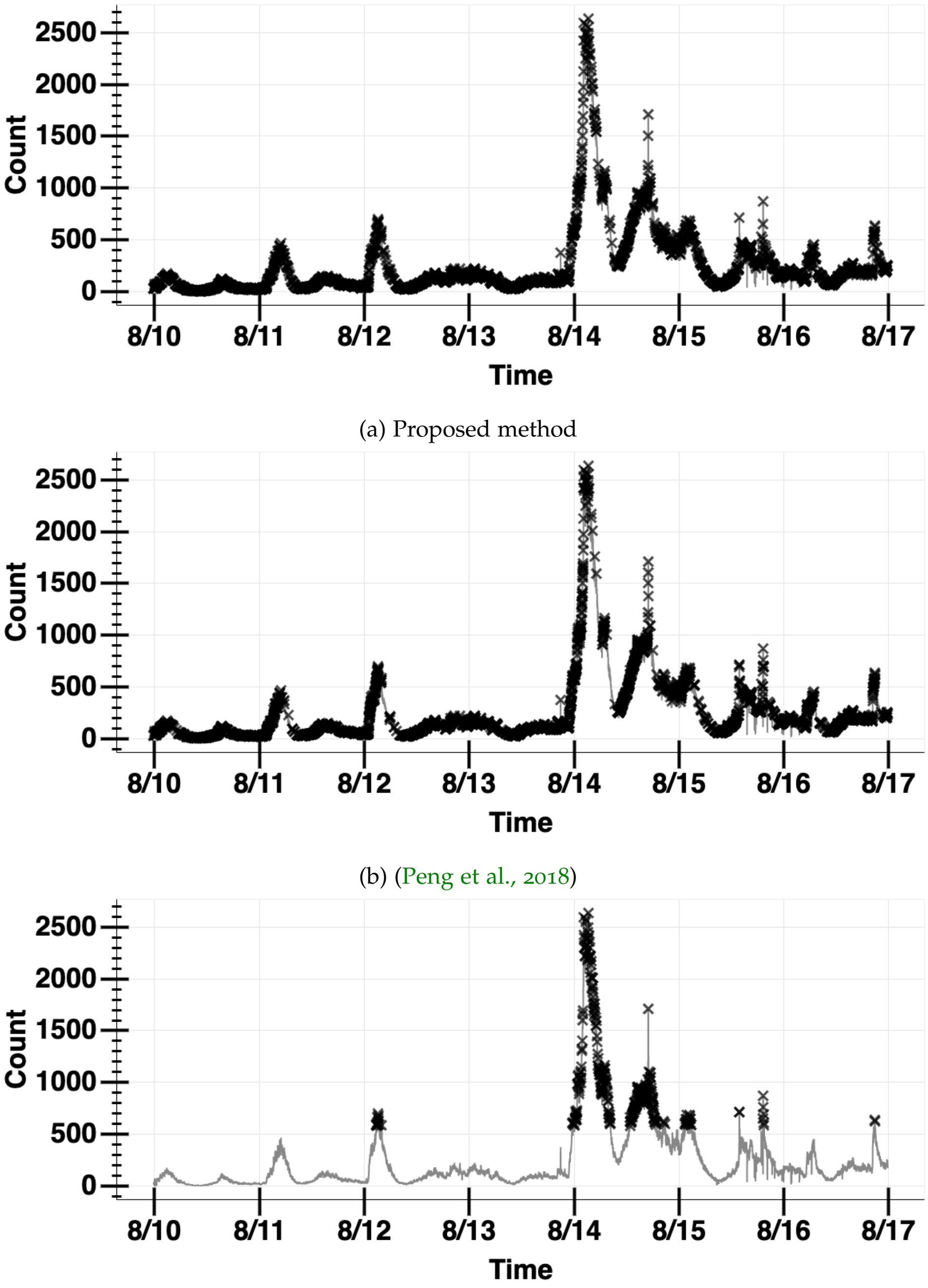


(a) Proposed method

(b) (Peng et al., 2018)

(c) (Gillani et al., 2017)

Figure 5.6: Burst detection results of three methods for the "charliehebdo".

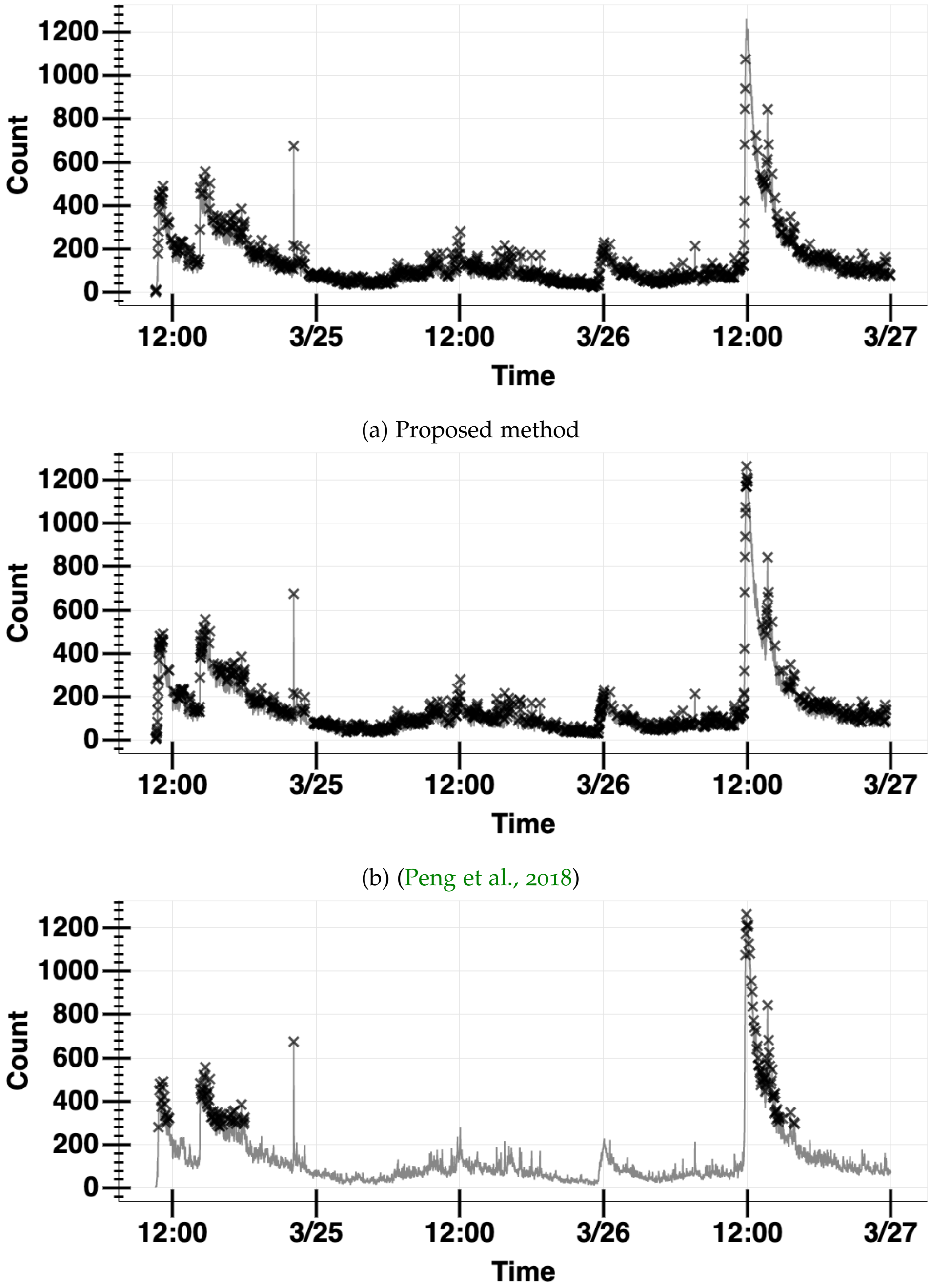


(a) Proposed method

(b) (Peng et al., 2018)

(c) (Gillani et al., 2017)

Figure 5.7: Burst detection results of three methods for the "fergusonunrest".

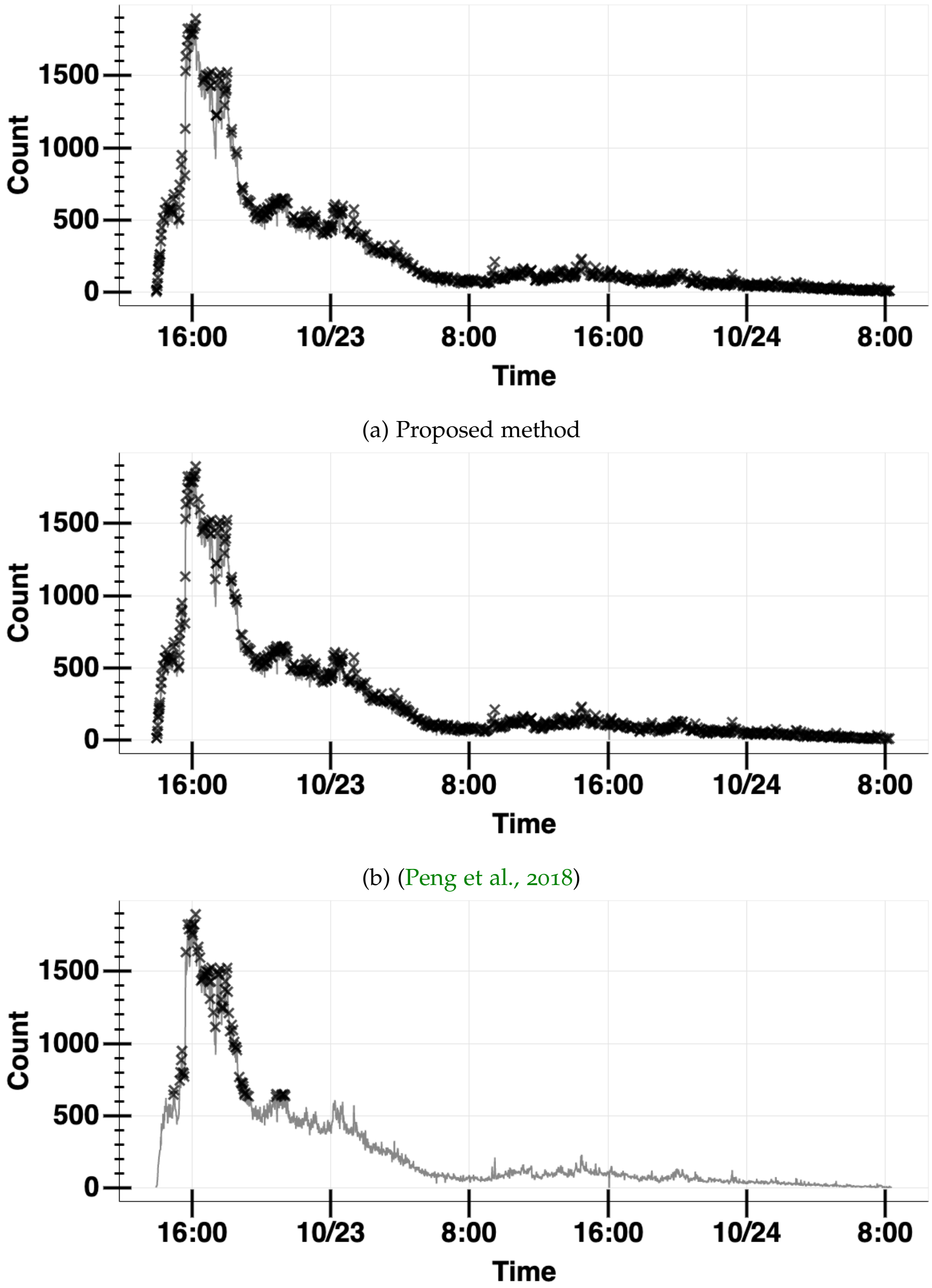


(a) Proposed method

(b) (Peng et al., 2018)

(c) (Gillani et al., 2017)

Figure 5.8: Burst detection results of three methods for the "germanwings".

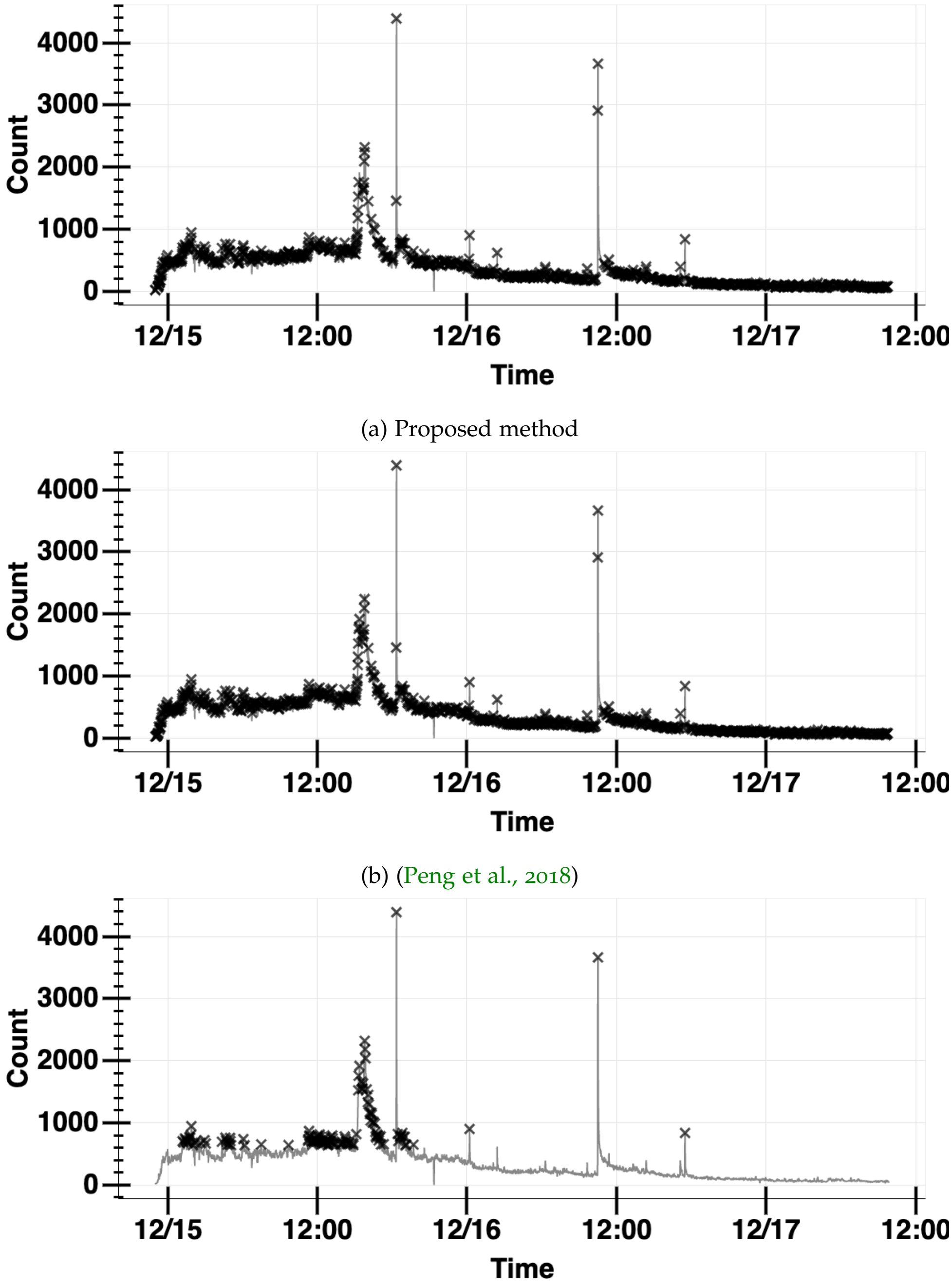


(a) Proposed method

(b) (Peng et al., 2018)

(c) (Gillani et al., 2017)

Figure 5.9: Burst detection results of three methods for the "ottawashooting".



(a) Proposed method

(b) (Peng et al., 2018)

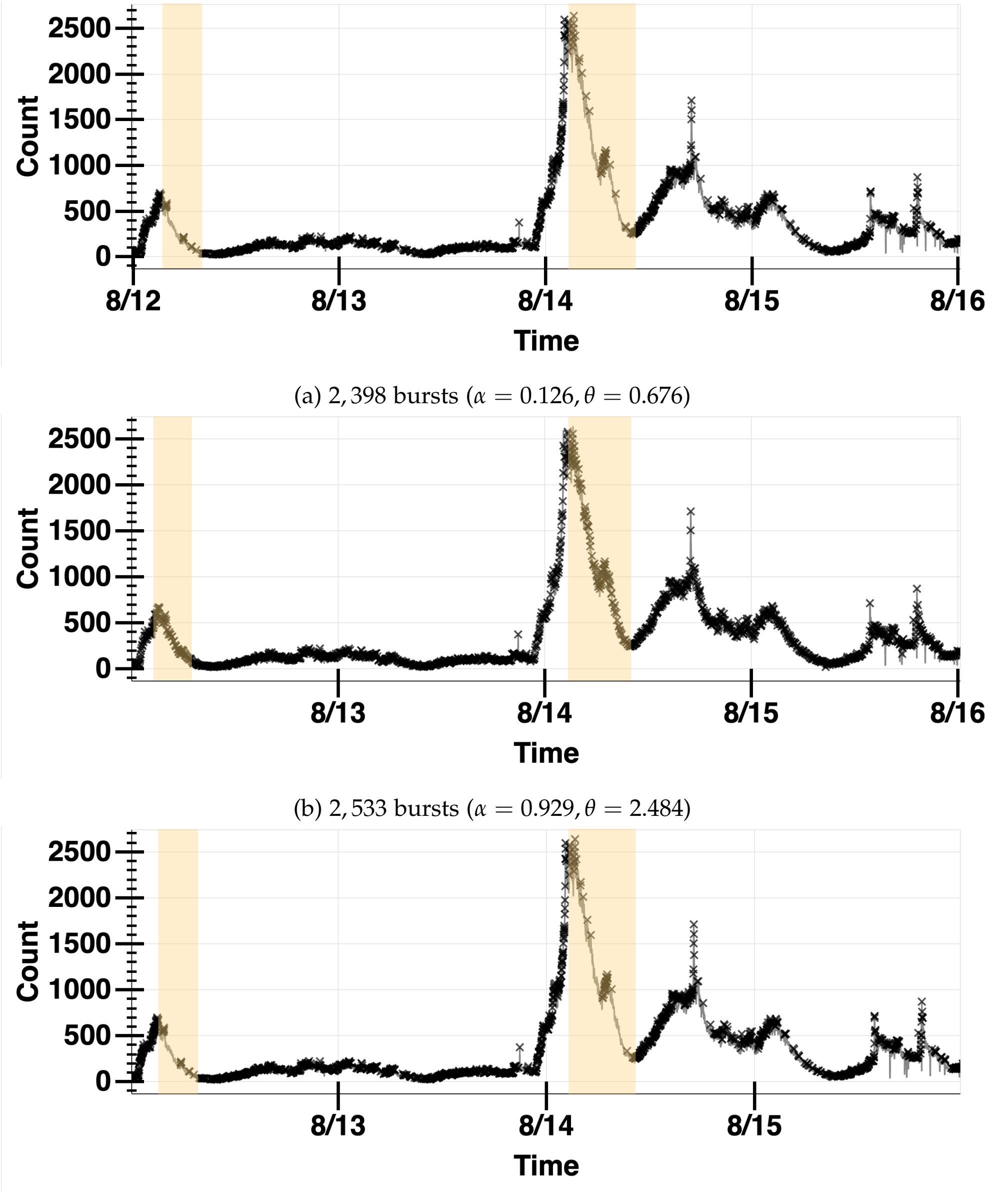
(c) (Gillani et al., 2017)

Figure 5.10: Burst detection results of three methods for the "sydneysiege".

Figures 5.11 and 5.12 show these observations using key bursts detected via (Peng et al., 2018) and the proposed method with different parameter values for the "fergusonunrest" event, respectively. Figures 5.11b and 5.11c respectively show 135 and 257 more bursts compared with Figure 5.11a. Peng's method detects few relatively small bursts after the highest point in Figures 5.11a and 5.11c. This indicates that the method identifies bursts before the highest point rather than those preceded by it. In Figure 5.11b, on the contrary, the method detects a large number of instances after the highest point. These observations suggest that Peng's parameters cannot be tweaked according to the number of identified bursts and a further analysis of the relationship between Peng's parameters and distributions of key bursts may be necessary.

Figures 5.12b and 5.12c respectively show 395 and 467 more bursts compared with Figure 5.12a. The results are more consistent than Figure 5.11. There is no significant change in the distribution of key bursts between Figures 5.12b and 5.12c.

Based on the observations made above, this chapter argues that the proposed burst detection method is more suitable for identifying key bursts for a broad range of events with different evolution patterns than the SOTA model.

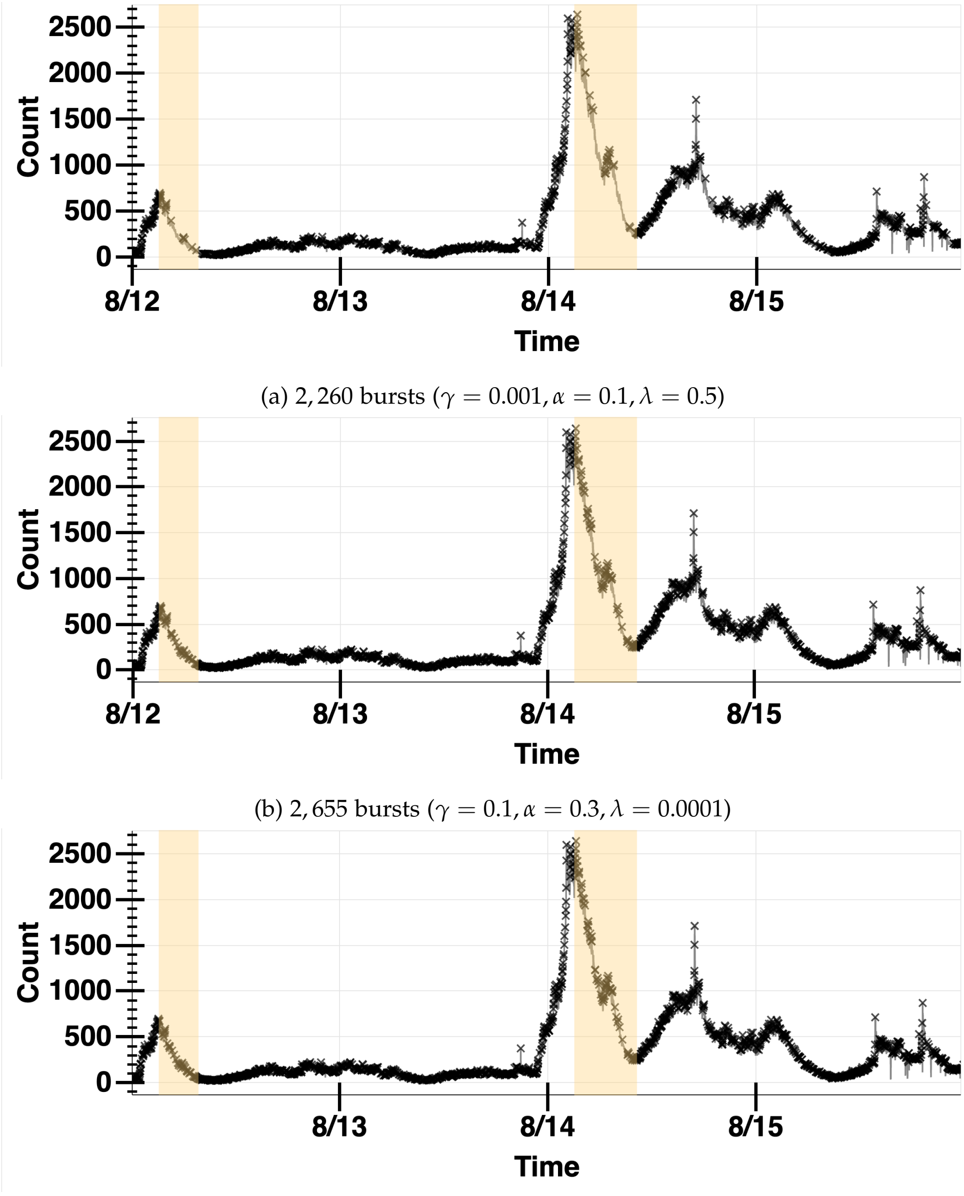


(a) 2, 398 bursts ($\alpha = 0.126, \theta = 0.676$)

(b) 2, 533 bursts ($\alpha = 0.929, \theta = 2.484$)

(c) 2, 655 bursts ($\alpha = 0.113, \theta = 0.571$)

Figure 5.11: Burst detection results obtained via (Peng et al., 2018) with different parameter values for the "fergusonunrest" event.

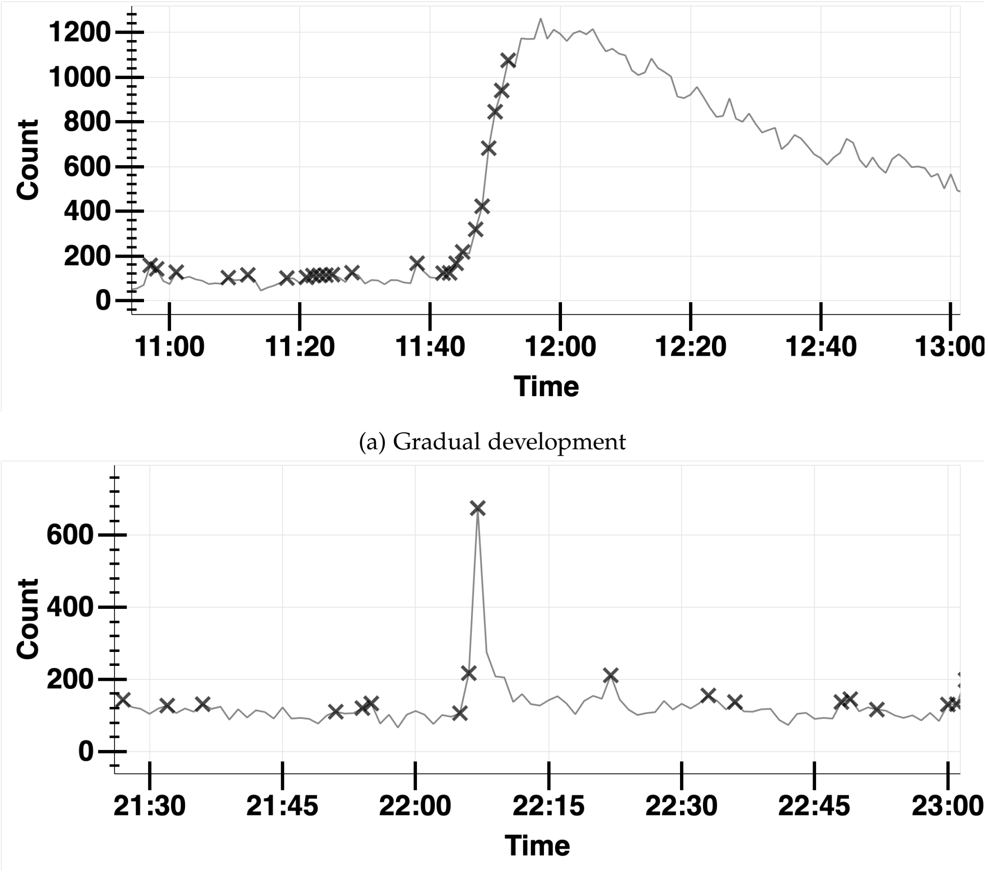


(a) 2,260 bursts ($\gamma = 0.001, \alpha = 0.1, \lambda = 0.5$)

(b) 2,655 bursts ($\gamma = 0.1, \alpha = 0.3, \lambda = 0.0001$)

(c) 2,727 bursts ($\gamma = 0.001, \alpha = 0.3, \lambda = 0.0001$)

Figure 5.12: Burst detection results obtained via the proposed method with different parameter values for the "fergusonunrest" event.



(a) Gradual development

(b) Rapid development

Figure 5.13: Two different patterns of development for the "germanwings" event. Detected bursts are marked with ×.

One noteworthy observation in Figure 5.8a is that the proposed method does not detect a time window which seems to be a burst or peak (i.e. 11:57 on 26 March 2015) for the "germanwings" event. Figure 5.13a shows this issue more clearly. Although several time windows before the highest point (i.e. 11:57) are detected, the peak and a couple of preceding windows are not labelled as key bursts. Intuitively, this can be because the popularity of the event has gradually developed during that period in comparison to a sudden surge between 22:05 and 22:07 on 24 March (see Figure 5.13b). To further investigate whether this issue poses a problem for the task of identifying potential rumours by missing new and important potential rumours, the top 10 summary tweets between 11:43 and 11:57 are examined. The summarisation method based on *dens* (see details in Section 5.3.3.3) is used for this analysis. Duplicated summary tweets are removed. Table 5.6 shows that all but one summary tweet (marked in blue) posted during the peak at 11:57 appeared in preceding time windows. It is worth noting that the one that appears at the peak for the first time is not completely new, and can be deduced by summary tweets appeared in preceding key bursts. More specifically, the fact that a co-pilot was alone and deliberately crashed the plane is already known. In addition, the first name and the initial of the surname of the co-pilot have also been identified in summaries of preceding key bursts (e.g. at 10:26 on 26 March): "German media naming Captain as 'Patrick S'. a dad of two with over ten years flying experience and co-pilot as '**Andreas L.**' #GermanWings."

The analysis results can lead to a conclusion that the proposed method might miss peaks followed by gradual development, yet does not miss newsworthy stories and rumours even without using linguistic signals. This is because the same or similar contents have already been detected from preceding key bursts. The results described in Table 5.6 further support the hypothesis of this chapter: emerging patterns are key signals for the identification of potential rumours rather than peaks.

Table 5.6: Unique summary tweets at 11:57 on 26 March 2015 for the "germanwings" and timestamps at which each tweet appears for the first time. "Label" indicates whether a corresponding timestamp is a key burst (1) or not (0).

Tweet	Time	Label
#BREAKING: Prosecutor says co-pilot, alone at helm of #Germanwings plane, began descent manually & intentionally. (Via @AP) #KTBS3	11:51	1
#crashA320 #germanwings "The pilot refused to open the door of the cockpit and deliberately crashed the plane"	11:43	1
BREAKING: Co-pilot of #Germanwings plane that crashed into French Alps took sole control; deliberately crashed the jet—French prosecutor	11:50	1
Co-Pilot, Andreas Lubitz, was alone and deliberately put the plane into decent. He was alive and alone in cockpit until impact. #GermanWings	11:57	0
RT @JWenbanSmith: Marseille Prosecutor Brice Robin says co-pilot was breathing normally, had no reason not to let captain in #4U9525	11:46	1

5.5.2 *Evaluation of Summarisation Methods*

As described in Section 5.4.5.2, representative tweets are extracted from each key burst for the five selected events in the *Twitter events 2012-2016 data*. Bursts detected using the parameters $\gamma = 0.1, \alpha = 0.3$, and $\lambda = 0.0001$, which detect 24.4% of the total number of time windows on average, are used for summarisation. Table 5.7 compares the different summarisation methods with different settings in terms of the total number of unique reference rumours that can be identified using the top 10 summaries of each key burst. The column names CH, FU, GW, OS, and SS refer to "charliehebdo", "fergusonunrest", "germanwings", "ottawashooting", and "sydneysiege", respectively. As the number of detected reference rumours at each key burst is added together for all key bursts, the reported values in the table can exceed 10.

For the term weighting schemes proposed in Section 5.3.3.3, three different settings are tested: 1) a percentage P used to selecting n-grams with high weights (see Section 5.3.3.4), 2) the number of terms (i.e. uni-grams and n-grams (n>1)), and 3) the use of IDF. For example, the method "K-cores (30, n-grams, idf)" uses scores of the top **30**% of **n-grams** multiplied by **IDF** values to compute tweet scores.

Summarisation performance varies according to events. Overall, Hybrid TF-IDF (Sharifi et al., 2013) identifies the greatest number of rumours, followed by *TextRank (10, uni-grams)*. For the "charliehebdo" and "fergusonunrest" events, graph-based methods tend to perform better than frequency-based methods. This probably has much to do with characteristics of tweets for the two events. Their tweets may have more structure and words establishing correlations among pairs of tweets compared with tweets for the other events. Therefore, such correlations can help graph-based models, which are more complex and relational than frequency-based ones, effectively learn underlying topical information (Inouye and Kalita, 2011) and extract more useful summaries. The opposite tendency is observed for the "ottawashooting" and "sydneysiege". Due to the limited number of references available for detection (see Table 5.4), most methods show similar performance for the "germanwings".

The impact of the parameter P is marginal. The performance of the proposed graph-based methods tend to increase when P is decreased to 10. The impact of IDF is marginal or can be adverse, which indicates that diminishing the weight of common words and increasing that of rare words are not very useful for graph-based summarisation methods for short texts like tweets in the context of potential rumour identification. It is notable that the TextRank-based method with $P = 10$ and uni-grams shows a notable improvement without IDF. As for n-grams, it is hard to conclude that the use of n-grams is more advantageous than that of uni-grams in graph-based summarisation methods. The benchmark model (Meladianos et al., 2018a) shows poor performance compared to the proposed graph-based methods. *SumBasic* (Nenkova and Vanderwende, 2005) shows strong performance particularly for the "ottawashooting" and "sydneysiege".

As described in Section 5.4.5, the first timestamp at which a method detects each reference is recorded. Table 5.7 compares the total number of recorded references. To evaluate model performance in terms of the early detection of potential rumours, this chapter compares the timestamps for each reference recorded by different methods. Specifically, if one or more methods detected a certain reference earlier than the others, it is considered that they succeeded in the early detection of that reference rumour. If all the methods detect a reference at the same time, they still earn one point. Table 5.8 shows the results. As for the early detection of potential rumours, the two frequency-based methods show strong performance. For the "fergusonunrest", *TextRank(10, n-grams)* significantly outperforms the others.

Table 5.9 shows how many rumours among the most popular 10 rumours (see Appendix a.2) for each event can be detected by different methods. Hybrid TF-IDF (Sharifi et al., 2013) detects about 8 popular rumours on average. The graph-based methods detect about 7 rumours on average with optimal settings. These results show that combining key burst detection and summarisation can successfully identify popular rumours in real-world events.

Table 5.7: Overall performance of different summarisation methods on potential rumour identification. The total number of detected rumour references in the top 10 summary tweets is shown. (CH: charliehebdo, FU: fergusonunrest, GW: germanwings, OS: ottawashooting, SS: sydneysiege)

Method	CH	FU	GW	OS	SS	Total
K-cores (100, n-grams, idf)	12	5	5	7	11	40
K-cores (90, n-grams, idf)	12	5	5	7	11	40
K-cores (50, n-grams, idf)	12	6	5	8	13	44
K-cores (30, n-grams, idf)	14	6	6	10	15	51
K-cores (10, n-grams, idf)	12	9	6	10	14	51
dens (100, n-grams, idf)	16	9	5	7	11	48
dens (90, n-grams, idf)	16	9	5	7	11	48
dens (50, n-grams, idf)	16	8	5	8	11	48
dens (30, n-grams, idf)	15	8	5	10	10	48
dens (10, n-grams, idf)	12	9	5	12	**19**	57
TextRank (100, n-grams, idf)	11	8	5	10	13	47
TextRank (90, n-grams, idf)	11	8	5	10	13	47
TextRank (50, n-grams, idf)	14	8	5	11	17	55
TextRank (30, n-grams, idf)	14	9	4	11	16	54
TextRank (10, n-grams, idf)	10	10	**7**	13	13	53
dens (30, n-grams)	13	10	6	10	12	51
dens (10, n-grams)	13	**11**	4	10	14	52
TextRank (30, n-grams)	12	8	4	10	17	51
TextRank (10, n-grams)	11	10	4	13	15	53
dens (30, uni-grams, idf)	14	8	5	11	10	48
dens (10, uni-grams, idf)	14	**11**	6	11	13	55
dens (10, uni-grams)	14	**11**	5	13	12	55
TextRank (30, uni-grams, idf)	13	7	5	11	13	49
TextRank (10, uni-grams, idf)	15	9	5	11	13	53
TextRank (10, uni-grams)	**19**	**11**	5	13	14	62
Meladianos et al. (2018a)	14	5	5	7	13	44
Hybrid TF-IDF	16	10	6	15	**19**	**66**
SumBasic	12	7	6	**16**	**19**	60

Table 5.8: Overall performance of different summarisation methods on the early detection of potential rumours. The total number of detected rumour references in the top 10 summary tweets is shown. (CH: charliehebdo, FU: fergusonunrest, GW: germanwings, OS: ottawashooting, SS: sydneysiege)

Method	CH	FU	GW	OS	SS	Total
K-cores (100, n-grams, idf)	2	1	1	1	3	8
K-cores (90, n-grams, idf)	2	1	1	1	3	8
K-cores (50, n-grams, idf)	3	1	2	2	3	11
K-cores (30, n-grams, idf)	5	3	3	2	3	16
K-cores (10, n-grams, idf)	4	5	4	2	2	17
dens (100, n-grams, idf)	5	2	2	1	6	16
dens (90, n-grams, idf)	5	2	2	1	6	16
dens (50, n-grams, idf)	5	2	2	1	5	15
dens (30, n-grams, idf)	4	2	2	1	2	11
dens (10, n-grams, idf)	1	4	2	1	4	12
dens (30, n-grams)	2	4	2	1	4	13
dens (10, n-grams)	1	3	1	2	3	10
dens (30, uni-grams, idf)	2	3	1	0	1	7
dens (10, uni-grams, idf)	2	4	3	0	3	12
dens (10, uni-grams)	3	2	2	3	2	12
TextRank (100, n-grams, idf)	1	2	2	2	8	15
TextRank (90, n-grams, idf)	1	2	2	2	8	15
TextRank (50, n-grams, idf)	3	4	2	2	12	23
TextRank (30, n-grams, idf)	3	5	4	2	11	25
TextRank (10, n-grams, idf)	1	3	2	4	4	14
TextRank (30, n-grams)	4	8	2	2	11	29
TextRank (10, n-grams)	3	10	2	3	5	23
TextRank (30, uni-grams, idf)	4	3	2	3	5	17
TextRank (10, uni-grams, idf)	6	4	2	2	5	19
TextRank (10, uni-grams)	8	5	2	4	7	26
Meladianos et al. (2018a)	6	3	1	4	4	18
Hybrid TF-IDF	7	6	6	8	8	35
SumBasic	7	2	3	8	15	35

Table 5.9: Number of detected top ten rumours for five events using different summarisation methods on potential rumour identification. The total number of detected rumour references in the top 10 summary tweets is shown. (CH: charliehebdo, FU: fergusonunrest, GW: germanwings, OS: ottawashooting, SS: sydneysiege)

Method	CH	FU	GW	OS	SS	Average
K-cores (100, n-grams, idf)	7	4	5	5	8	5.8
K-cores (90, n-grams, idf)	7	4	5	5	8	5.8
K-cores (50, n-grams, idf)	7	6	5	5	9	6.4
K-cores (30, n-grams, idf)	6	6	6	8	10	7.2
K-cores (10, n-grams, idf)	5	7	6	8	9	7
dens (100, n-grams, idf)	8	8	5	6	8	7
dens (90, n-grams, idf)	8	8	5	6	8	7
dens (50, n-grams, idf)	8	7	5	6	8	6.8
dens (30, n-grams, idf)	7	7	5	8	7	6.8
dens (10, n-grams, idf)	6	7	4	9	10	7.2
TextRank (100, n-grams, idf)	5	6	5	7	8	6.2
TextRank (90, n-grams, idf)	5	6	5	7	8	6.2
TextRank (50, n-grams, idf)	6	6	5	8	9	6.8
TextRank (30, n-grams, idf)	5	6	4	8	10	6.6
TextRank (10, n-grams, idf)	5	7	6	9	8	7
dens (30, n-grams)	7	8	6	8	9	7.6
dens (10, n-grams)	5	7	6	9	8	7
TextRank (30, n-grams)	5	7	4	8	10	6.8
TextRank (10, n-grams)	6	7	4	8	10	7
dens (30, uni-grams, idf)	7	5	5	8	6	6.4
dens (10, uni-grams, idf)	7	8	6	8	8	7.4
dens (10, uni-grams)	7	8	5	8	9	7.4
TextRank (30, uni-grams, idf)	5	5	5	8	9	6
TextRank (10, uni-grams, idf)	5	6	5	8	10	6.8
TextRank (10, uni-grams)	6	8	5	8	10	7.4
Meladianos et al. (2018a)	6	7	6	5	8	4.8
Hybrid TF-IDF	7	9	6	9	10	8.2
SumBasic	6	6	5	8	8	6.6

5.5.3 *Evaluation of Burst Detection for* ERD

Burst detection methods are evaluated using the *SumBasic* (Nenkova and Vanderwende, 2005). Evaluation is conducted using the same practice used to evaluate summarisation methods. Table 5.10 compares the total number of detected rumour references in the top 10 summaries of each key burst detected via different burst detection methods. As the number of detected reference rumours at each key burst is added together for all key bursts, the reported values in the table can exceed 10. Overall, the proposed burst detection method outperforms the SOTA methods. (Gillani et al., 2017) which aims to detect peak performs poorly, (Peng et al., 2018) which focuses on emerging performs as well as the proposed method. This observation supports the hypothesis of this chapter described in Section 3.2.2 and 5.4.2: bursts (i.e. time windows lying in increasing lines in a time series plot) are the key to discovering potential rumours in the early stages of rumour evolution. Table 5.11 compares the total number of *early* detected rumour references in the top 10 summaries of each key burst. The best-performing model is different among different events, but the proposed method and Peng's method shows very similar performance. Remember that both methods exploit emerging bursts in event time series plots. This suggests that extracting summaries from such bursts is an effective way to identify newly emerging potential rumours.

Table 5.10: Overall performance of different burst detection methods on potential rumour identification. The total number of detected rumour references in the top 10 summary tweets is shown.

Event	Proposed method	Peng et al. (2018)	Gillani et al. (2017)
charliehebdo	**12**	**12**	3
fergusonunrest	**7**	6	1
germanwings	**6**	**6**	2
ottawashooting	**16**	**16**	9
sydneysiege	**19**	18	13

Table 5.11: Overall performance of different burst detection methods on the early detection of potential rumours. The total number of detected rumour references in the top 10 summary tweets is shown.

Event	Proposed method	Peng et al. (2018)	Gillani et al. (2017)
charliehebdo	**12**	9	1
fergusonunrest	**6**	**6**	0
germanwings	4	**5**	1
ottawashooting	**15**	14	2
sydneysiege	17	**18**	6

5.6 CONCLUSION

This chapter proposed a framework for key burst detection and summarisation as a preliminary step for ERD. The main advantage of the proposed method is that it does not require the training of ML models and yet it can produces desired outcome for different events with different propagation patterns. Several studies on burst detection fail to clarify or include a parameter analysis, which makes it difficult to transfer and generalise their methods to new data. This chapter showed how each parameter affects the performance of burst detection. For evaluation, bursts detected using different methods were visualised. The proposed method and a SOTA baseline showed similar results. The evaluation of different methods in the context of potential rumour identification showed that bursts detected by the proposed methods contain more newsworthy stories and potential rumours than those detected by the SOTA baselines. Overall, the proposed methods are more suitable for real-world scenarios than SOTA methods for temporal signal-based burst detection.

As for summarisation, this chapter conducted a comprehensive analysis of graph- and frequency-based methods with various settings. The experiments showed that there is no summarisation method which always outperforms the others on the identification of potential rumours. The top-performing method varies between events. On the other hand, frequency-based models significantly outperformed the graph-based methods in the early identification of potential rumours. Overall, summaries of key bursts can effectively identify tweets related to newsworthy stories and rumours.

The identification of potential rumours via key burst detection and summarisation has several applications. One possible application is that generated summaries can help practitioners in several domains (e.g. journalists and emergency responders) efficiently and effectively understand trending topics. Another example is that generated summaries can be used as input to a tweet-level rumour detection model. In this case, generated summaries can be referred to as potential rumours. This component is particularly necessary to employ a rumour detection model in real-world scenarios such as newly emerging breaking news events because it is inefficient and impractical to analyse millions of social media messages.

6 CONTEXT-AWARE EARLY RUMOUR DETECTION

6.1 INTRODUCTION

As stated in Section 1.3, the proposed three research topics in this thesis fit together to form an end-to-end message-level rumour detection framework. First of all, in Chapter 4, labelled training data augmentation with weak supervision was researched to address the labelled data scarcity and class imbalance problems in current rumour data sets. Its output includes weakly labelled rumour data sets and a fine-tuned NLM. Secondly, in Chapter 5, the automatic identification of potential rumours with minimum human supervision was studied as a data reduction module for message-level ERD. In other words, its output is input to a rumour detection model. The augmented data in Chapter 4 was exploited in Chapter 5 to evaluate burst detection and text summarisation results in the context of potential rumour identification. This novel evaluation approach demonstrated that the proposed methods for potential rumour identification could actually solve the intended research problem. This chapter finally delves into context-aware, message-level ERD by addressing RQs 3.1 and 3.2. The augmented data and fine-tuned NLM obtained in Chapter 4 will be employed in this chapter. Previous research related to this chapter was introduced in Section 2.3.3.

The early detection of rumours on social media is important to prevent and resolve problems posed by them to limit rumour spreading and to prompt healthy information ecosystem. Some example problems include false beliefs and myths; unnecessary public expenditure on research and public campaigns aimed at debunk them; and biased public opinion on political and societal decisions (Lewandowsky et al., 2012). This task is very challenging because limited and noisy information is available during the early stages of event diffusion. Most of the existing methods have worked on event-level detection which are not appropriate for detecting different rumour stories in the very early stages (Section 2.1.2).

When humans find it difficult to make sense of a piece of information, it is natural to seek coherence in its surroundings (Mitra and Gilbert, 2015). In message-level rumour detection, social-temporal context typically refers to conversational threads of source tweets such as replies in the case of Twitter. They provide information concerning how people's reactions to tweets related rumours evolve and how misinformation is self-corrected on social networks over time. Therefore, investigating them in their early development stages can offer valuable insights as to how rumours propagate before they become widespread and have a far-reaching impact. This chapter exploits Twitter metadata-based features to obtain social contextual information. In specific, features associated with users who have engaged in a conversation and tweet-level attributes of replies are leveraged. Exploiting social-temporal context provides auxiliary information to ERD models and helps to boost their effectiveness because the textual content and metadata of tweets are highly correlated (Kıcıman, 2010).

This chapter proposes a novel context-aware hybrid neural network architecture, called *Rumour Propagation-Based Deep Neural Networks* (***RP-DNN***), to address the problem of message-level ERD. It combines a task-specific character-based BiLM and stacked LSTM networks to represent textual contents and social-temporal contexts of input source tweets. Therefore, it is able to model not only linguistic characteristics of rumours, but also propagation patterns of rumour sources (i.e. users' reactions to rumours) in the early stages of their development. Moreover, multi-layered attention mechanisms are applied to effectively learn important contexts, which results in performance gains.

In the *RP-DNN*, *source tweet contents (SC)* are encoded using a SOTA context-aware NLM fine-tuned specifically for the task of rumour detection. Social context is jointly represented by *context contents (CC)* and *context metadata (CM)*. CC is exploited to offer insights about how linguistic patterns towards corresponding source tweets evolve. CM is utilised to provide auxiliary information on underlying and implicit patterns of information diffusion which are strongly correlated with contents (Kıcıman, 2010).

This chapter is based on my publication in the proceedings of the 16th International Conference on Information Systems for Crisis Response And Management (Gao et al., 2020). I share co-first authorship with Jie Gao on this paper. The main contributions of this chapter are summarised as follows:

- This chapter proposes an extensible hybrid deep learning framework for rumour classification *at individual tweet level*, while the majority of recent studies focus on event-level classification. The *RP-DNN* advances SOTA performance for tweet-level ERD.

- This chapter researches a *context-aware* model which jointly learns SC, CC, and CM. Most recent solutions for context-aware ERD are event-level (Ruchansky et al., 2017; Ma et al., 2016; Zhou et al., 2019b; Chen et al., 2018; Guo et al., 2018). Only a few studies (Ma et al., 2018b; Liu and Wu, 2018; Kochkina et al., 2018a; Geng et al., 2019) have exploited conversational threads for message-level rumour detection. Moreover, little work on rumour detection has focussed on a NN architecture taking different types of contextual features in addition to contents as input. In the RP-DNN, for each input source tweet, tweet embeddings for SC and CC are obtained using a task-specific NLM. CM is represented as a numerical vector. Stacked LSTM networks and attention mechanisms are adopted to model social-temporal propagation dynamics (i.e. CC and CM). The experiments show that the *RP-DNN* can effectively learn joint representations of source contents and social contextual information.

- Labelled data scarcity is a known limitation in the field of ERD. Unlike most SOTA research on message-level rumour detection which evaluates proposed methods on small data, this chapter conducts experiments on a publicly available large-scale rumour data generated via weakly supervised data augmentation proposed in Chapter 4.

- This chapter undertakes extensive experiments and performs comprehensive evaluation through k-fold CV, LOOCV, and an ablation study. In particular, LOOCV is adopted to address the RQ 3.2. It shows the effectiveness of the *RP-DNN* in real-world scenarios which entail the

RQ 3.2: How can rumour detection architectures be evaluated in realistic scenarios in which detection models are required to identify unseen rumours?

classification of instances which have different characteristics (e.g. propagation patterns, linguistic features, etc.) from training data. An ablation study is conducted to examine the effectiveness and generalisability of the proposed solution. To the best of my knowledge, this chapter carries out the most comprehensive evaluation of message-level ERD which no work on message-level rumour detection has ever done.

- Experimental results show that the *RP-DNN* outperforms SOTA models for *message-level* rumour detection and achieves comparable performance with SOTA *event-level* rumour detection models.

6.2 METHODOLOGY

6.2.1 *Problem Statement*

Rumours are commonly considered as unverified statements that lack substantiation (see Section 2.1.1). In the task of message-level ERD proposed in this chapter, candidates for rumours (i.e. potential rumours) refer to source tweets which report updates on newsworthy events, but are deemed unsubstantiated at the time of their origination. The literature review presented in Section 2.3.3 shows that textual contents of tweets are an important cue for identifying rumours. As they are very short, however, they contain limited contexts of various lengths. To address this issue, this chapter represents each source tweet using its textual content (SC), that of its conversational thread (CC), and hand-crafted features extracted from the conversation (CM).

Formally, an input set of candidate source tweets is denoted by $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_i\}$, where each candidate $\mathbf{X}_k = \{[x_k, \mathbf{CC}_k, \mathbf{CM}_k], t_k\}$ consists of the k^{th} source tweet content x_k and two correlated context sets $\mathbf{CC}_k$ and $\mathbf{CM}_k$ at time t_k. $\mathbf{CC}_k$ consists of textual contents of all conversational contexts cc_{kj} at timestamp t_{kj}, and is denoted as $\mathbf{CC}_k = \{cc_{kj}, t_{kj}\}$. $\mathbf{CM}_k$ consists of their metadata cm_{kj} at timestamp t_{kj}, and is denoted as $\mathbf{CM}_k = \{cm_{kj}, t_{kj}\}$. $k = 1, 2, \cdots, i$ denotes the index of source tweets, and $j = 1, 2, \cdots,$ (`the number of replies to the source tweet`) denotes the chronological index of conversational contexts. Note that j allows limiting context size. Let $y_k = \{0, 1\}$ be binary labels, where $y_k = 0$ and $y_k = 1$ denote a non-rumour and rumour, respectively. The task is to predict the most probable tag for each candidate source tweet x_k based on source tweet content x_k and all types of contexts $\mathbf{CC}_k$ and $\mathbf{CM}_k$.

6.2.2 *Overview of the Architecture of the Proposed Solution*

The overall architecture of the proposed RP-DNN is shown in Figure 6.1. The figure only illustrates how it classifies a single source tweet. In the actual implementation of the model, it takes a sequence of source tweets and corresponding social contexts (i.e. replies) as input and outputs a sequence of labels (i.e. rumour or non-rumour) for individual input source tweets. The RP-DNN consists of five major parts including 1) data preparation and ingestion; 2) tweet text embedding layers; 3) contextual information encoding; 4) stacked LSTM layers with attention mechanisms; and 5) a classification layer.

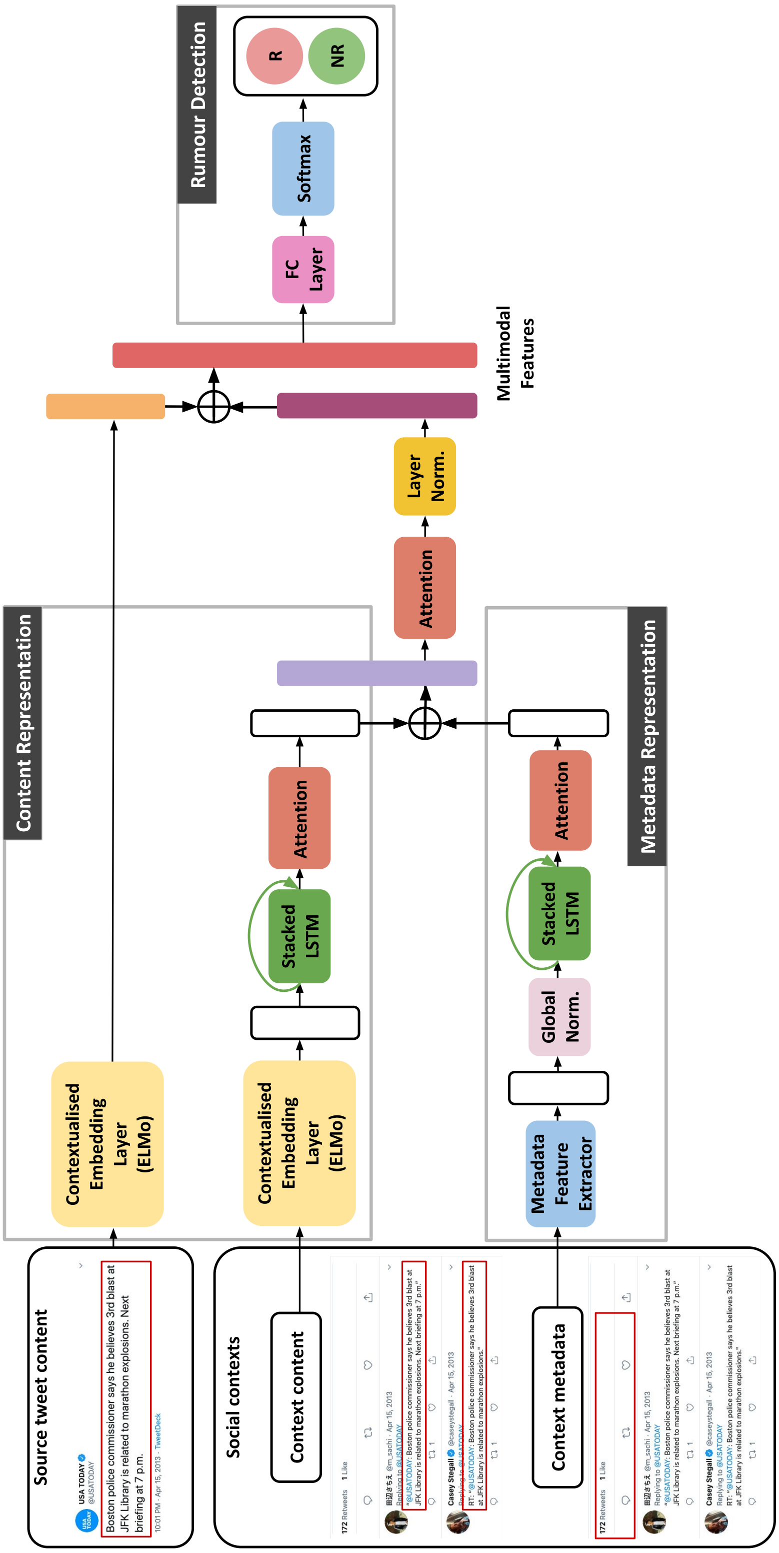

Source tweet content
USA TODAY
@USATODAY
Boston police commissioner says he believes 3rd blast at JFK Library is related to marathon explosions. Next briefing at 7 p.m.
10:01 PM · Apr 15, 2013 · TweetDeck
Social contexts
Context content
Context metadata
172 Retweets 1 Like
Casey Stegall @caseystegall · Apr 15, 2013
Content Representation
Contextualised Embedding Layer (ELMo)
Contextualised Embedding Layer (ELMo)
Stacked LSTM
Attention
Metadata Representation
Metadata Feature Extractor
Global Norm.
Stacked LSTM
Attention
Attention
Layer Norm.
Multimodal Features
Rumour Detection
FC Layer
Softmax
R
NR

Figure 6.1: Overview of the RP-DNN. ⊕ denotes the concatenation of two matrices.

Details for the key stages to perform tweet-level ERD using the RP-DNN are as follows:

1. **Data preparation and ingestion**: As the first step, Twitter data is collected and preprocessed. The details of data collection, preprocessing, and filtering techniques are illustrated in Section 6.3. Once a set of candidate source tweets has been created, twofold data ingestion is performed: source tweets are firstly loaded into the model. Secondly, corresponding contexts of a maximum size of j are loaded.

2. **Tweet text embedding**: This step handles source tweet contents (SC) and context contents (CC). Most standard word embedding techniques need to build a fixed set of unique words (i.e. vocabulary), which may cause an out-of-vocabulary problem in NLP (i.e. words appearing in input are not in a vocabulary; Peters et al. (2018)). The proposed solution employs a purely character-based NLM (ELMo; Peters et al. (2018)), which is pre-trained on a large Twitter corpus and then fine-tuned for rumour detection on social media, in order to represent tweet text contents. It has several advantages over conventional word embedding models. Firstly, it only requires basic text preprocessing to handle tweet content (see Section 6.3.3). Secondly, it is capable of representing individual words by considering the entire context (i.e. modelling polysemy) in which they are used (Peters et al., 2018). Conventional models are context-independent; they output a single vector for each word combining its multiple meanings. Finally, it does not need to build an extra vocabulary as it is purely based on characters rather than a dictionary of words. It maps individual tokens in each text (e.g. a sentence and paragraph) to sequences of character ids, which allows the model to output embeddings for tokens that are not found in the vocabularies of the pre-trained models (i.e. out-of-vocabulary problem).

3. **Contextual information encoding**: This step adresses the RQ 3.1. Conversational contexts are converted into inputs for two separate stacked RNNs layers for contextual modelling. It consists of CC embedding and CM encoding layers. The former converts all replies for the k^{th} source tweet into a sentence embedding matrix (i.e. an array of all replies' text embeddings) denoted by $\mathbf{V}^k_{cc}$. The latter uses the Metadata Feature Extractor (MFE) to extract a feature vector matrix $\mathbf{V}^k_{cm}$ (i.e. an array of metadata feature vectors for all replies), which can characterise user interactions and rumour diffusion patterns. Each feature in $\mathbf{V}^k_{cm}$ is normalised by applying a global mean and variance computed from the training data (see Table b.1). The main reason for using global normalisation is that outputs of batch normalisation are subject to batch sizes (Ba et al., 2016). Specifically, the choice of a wrong batch size can lead to poor training performance.

RQ 3.1: What contextual information can be leveraged into deep learning-based ERD? How can they be obtained and learnt?

4. **Stacked RNNs and attention layers**: This step adresses the RQ 3.1. They generate social-temporal context representations. Multiple LSTM networks are stacked together to form a *stacked LSTM* that takes input representations ($\mathbf{V}^k_{cc}$ and $\mathbf{V}^k_{cm}$) sorted in chronological order from the encoding layers. Let the number of layers be L. Stacked L-layer LSTM

networks ($L = 2$ in the case of this chapter) are utilised to process each of the two types of contextual data separately. The recurrent structure is capable of learning features of sequential inputs. Subsequently, soft hierarchical attention mechanisms (i.e. *the* 1^{st} *attention layer*) are applied on the top of the two stacked LSTM networks to produce optimal representations. The attention-weighted outputs (i.e. hidden states) of the two stacked LSTM layers are denoted by $\mathbf{H}^k_{cm}$ and $\mathbf{H}^k_{cc}$. They are combined to form a joint representation ($\mathbf{H}^k_{\text{cxt}}$) for the conversational context of the k^{th} source tweet. The third attention model (i.e. *the* 2^{nd} *attention layer*) is applied to a sequence of joint hidden states $\mathbf{H}^k_{\text{cxt}}$. Eventually, a compact representation of the input sequence of replies $\mathbf{C}^k_{\text{cxt}}$ is obtained. It is fed into a classification layer in the next stage to make classification of the k^{th} source tweet. Layer normalisation (Ba et al., 2016) is applied after *the* 2^{nd} *attention layer*. The details of the context representation learning layers are illustrated in Section 6.2.3 and 6.2.4, respectively.

5. **Classification layer**: This is the final output layer which outputs a label (i.e. rumour or non-rumour) for the k^{th} source tweet. An SC embedding and the joint representation of contexts are concatenated to form the final representation of each input source tweet. Finally, a 3-layer fully-connected NN with Leaky ReLu activations and softmax function takes the final representation to yield output. The architecture is trained using the cross-entropy. For the model settings, see the details in Section 6.2.7.

6.2.3 *Stacked LSTM Layer*

The motivation behind using conversational threads for rumour detection.

Compared to other traditional documents such as news articles, tweets are short and contain multiple languages; ungrammatical phrases and sentences; and unofficial abbreviations. Therefore, a single tweet contains limited information. Conversational threads of targeted source tweets have recently become popular in research on rumour detection as they give contextual information about corresponding source tweets (Kochkina et al., 2018a; Ma et al., 2018b). Based on these observations and findings, the proposed architecture leverages them to get additional information regarding rumour source tweets. As shown in Figure 6.2, social reactions are often represented as a temporally ordered sequence.

A brief introduction to RNNs

RNNs are a popular choice for sequence modelling. This chapter also employs them to model an input sequence of contexts. They process a sequential input in a way that resembles how humans do. An operation, defined by $h_t = f_W(x_t, h_{t-1})$, is performed on each element (i.e. reply) in an input sequence, where h_t is the hidden state at time step t and W is the weights of a network. The hidden state at each time step depends on the previous hidden state. Therefore, the temporal order of replies in an input sequence is important. Intuitively, this process enables RNNs to model the evolution of public opinion on each rumour source tweet and diffusion patterns of user engagement (e.g. retweets, likes) through metadata. RNNs can also handle sequences of variable lengths.

In order to represent contexts by utilising different types of features (e.g. textual contents and social-temporal features), conventional approaches (Ruchansky et al., 2017; Jin et al., 2017b; Li et al., 2019) simply concatenate

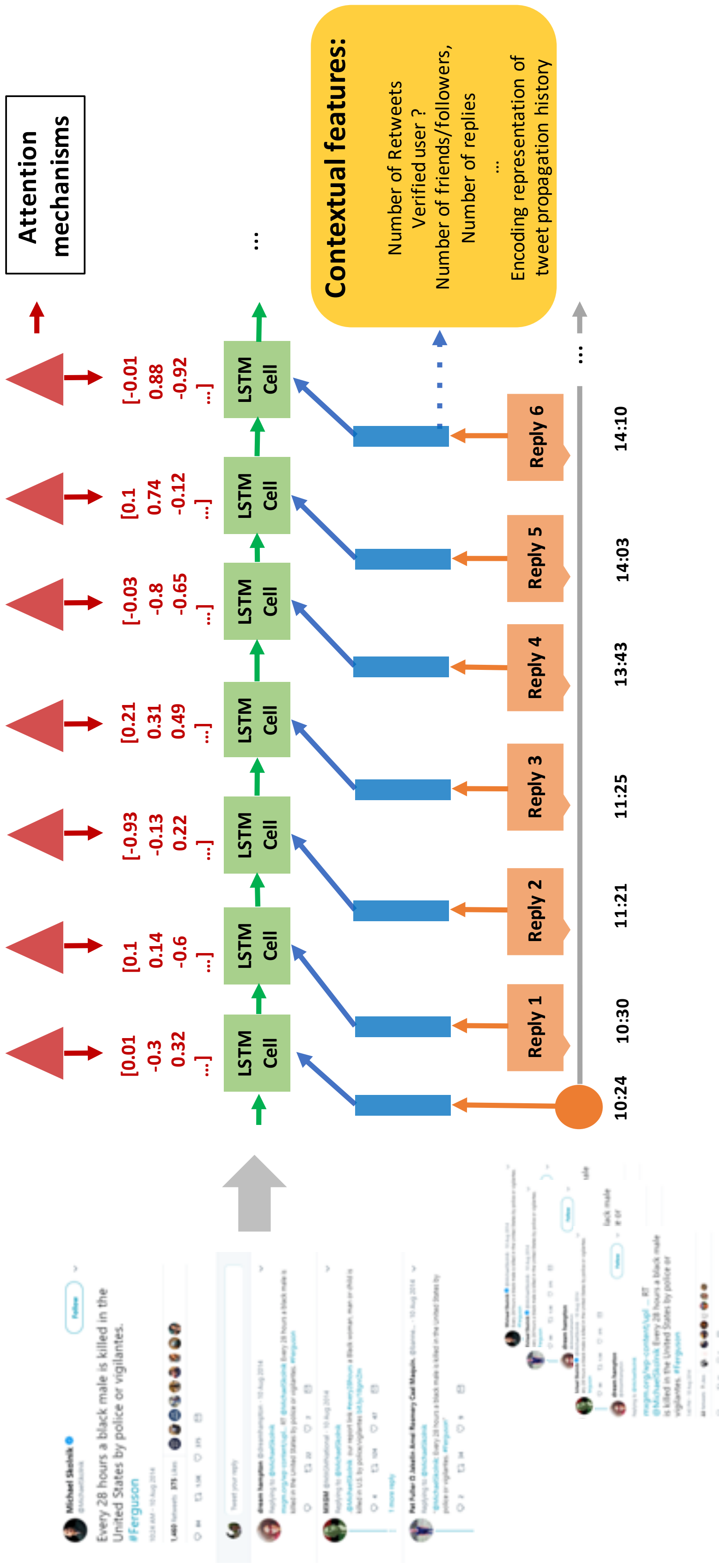


Figure 6.2: Overview of social-temporal context encoder with the 1st layer of attention mechanisms for CM representations. The attention mechanisms are applied on the top of the outputs of LSTM networks to generate attention-weighted context vectors for individual reply.

embeddings of different data inputs or process them via a linear combination to form a joint representation. This practice completely ignores correlations and differences between different types of contextual information. This chapter argues that a rumour detection model should be able to learn weights separately from different context inputs. This will make it possible to learn salient characteristics of each type of context. In addition, a model should have the ability to pay more attention to certain observations which give vital clues to the identification of rumours (see details in Section 6.2.4).

To this end, the proposed architecture employs two layers of forward LSTM (i.e. stacked LSTM) with the aim of learning abstract features of two correlated contexts (i.e. CC and CM). LSTM is described in Section 2.6.2. Concretely, a context content embedding matrix ($\mathbf{V}_{cc}^{k}$), which consists of a content embedding $v_{cc,t}^{k}$ for the reply at each time step t, is given as input to 2-layer forward LSTM to model the temporal evolution of public opinions. The output state $h_{cc,t}^{k} \in \mathbf{H}_{cc}^{k}$ at each time step t is given as follows:

$$\overrightarrow{h_{cc,t}^{k}} = \overrightarrow{LSTM_l}(\overrightarrow{h_{cc,t-1}^{k}}, v_{cc,t}^{k}), \ \forall t \in [1, j]. \tag{6.1}$$

As for learning diffusion patterns of user interactions and engagement, shallow features extracted from the explicit information (i.e. metadata) of social reactions are employed to build hierarchical RNNs, see Figure 6.2. In other words, a set of context metadata embeddings ($\mathbf{V}_{cm}^{k}$), which consists of a metadata embedding $v_{cm,t}^{k}$ for the reply at each time step t, is given as input to 2-layer forward LSTM. Unlike existing studies which exploit handcrafted features for rumour detection, the proposed LSTM-based model avoids painstakingly complicated feature engineering by allowing LSTM networks to learn the underlying social-temporal dynamics of complex hierarchical social structure. The output state $h_{cm,t}^{k} \in \mathbf{H}_{cm}^{k}$ at each time step t is given as follows:

$$\overrightarrow{h_{cm,t}^{k}} = \overrightarrow{LSTM_l}(\overrightarrow{h_{cm,t-1}^{k}}, v_{cm,t}^{k}), \forall t \in [1, j]. \tag{6.2}$$

6.2.4 *Stacked Soft Attention Mechanisms*

Different reactions in a conversational thread of a source tweet do not contribute equally to identifying whether the source is a rumour or not. Based on this insight, this chapter exploits attention mechanisms, which enable the proposed architecture to identify replies that are highly significant and useful for ERD. Stacked LSTM illustrated in the previous section emits its hidden state h_t at each time step t when processing a sequence. Conventionally, the hidden state of the last time step is exploited for classification. Although a latent representation of RNNs at each time step is a function of all previous steps, using the last hidden state might not be able to capture long-term memory due to its limited memory, even with LSTM (see Section 2.6.2).

Based on the findings illustrated above, attention mechanisms (see Section 2.6.3) are employed in this chapter. They are one of the recent advances in NNs (Vaswani et al., 2017). They aim to focus on certain parts of sequential data so that important elements can have a greater impact on the prediction of a target. They have successfully been applied to both areas of computer vision (Li et al., 2018; Grewal et al., 2018) and NLP (Choi et al., 2018; Tachibana et al., 2018; Hu, 2019). In the field of NLP, in particular, attention models highlight specific words or sentences to distil information from input text. These approaches have improved the performance of a wide range of applications in conjunction with deep learning architectures including CNNs and RNNs (Wang and Tax, 2016; Hu, 2019; Chaudhari et al., 2019).

Inspired by the impressive performance of attention mechanisms, this chapter explores how to employ them in context modelling to eliminate insignificant information and get more accurate and meaningful contextual information, thereby advancing SOTA performance in ERD. Several studies on visual recognition show that learning input data with multiple attention layers can progressively refine feature maps and focus on more salient features (Yang et al., 2016b; Wang et al., 2017).

In order to amplify the effects of important replies and filter noise and unnecessary information in a final representation of contexts, this chapter adopts the idea of hierarchical attention networks (Yang et al., 2016a) and introduces *context-level stacked attention mechanisms* into the RP-DNN. Formally, let $\mathbf{H}_*^k = \{h_*^1, h_*^2, \cdots, h_*^j\}$ be all recurrent hidden states for conversational contexts (* denotes input types and is one of CC, CM, and their joint representation (CXT)) of the k^{th} source tweet (see Section 6.2.3). Attention mechanism for reweighing the hidden state of each reply at each time step is denoted by *attention*($\mathbf{H}_*^k$)=$\mathbf{H}_{*\_new}^k = \{h_{*\_new}^1, h_{*\_new}^2, \cdots, h_{*\_new}^j\}$, and is shown in Eq. 6.3-6.5.

$$e_*^t = tanh(W_h h_*^t + b_h), \forall t \in [1, j]. \tag{6.3}$$

$$\alpha_*^t = softmax(e_*^t) \tag{6.4}$$

$$h_{*\_\text{new}}^t = \alpha_*^t h_*^t \tag{6.5}$$

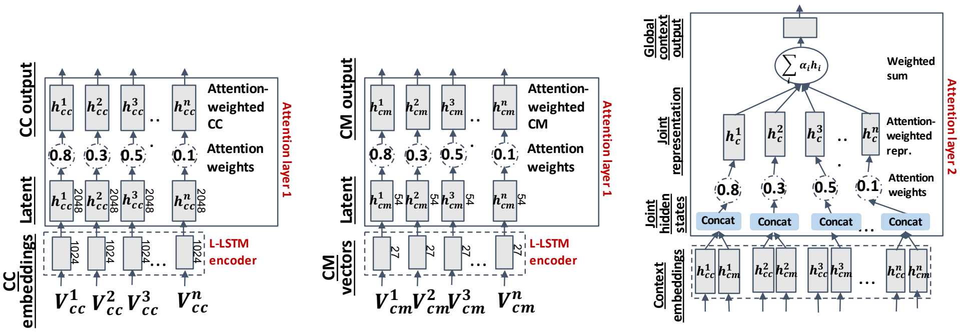
where $h_*^t \in \mathbf{H}_*^k$. W_h and b_h are an attention layer's weights and bias, which are initialised using He initialisation (He et al., 2015) and optimised during training. The standard softmax function (Martins and Astudillo, 2016) is used to approximate a normalised probability distribution over conversational context input. Zero padding is used to handle variable input lengths (i.e. the number of replies). Padded sequence vectors are masked with negative infinity following the practice introduced in (Vaswani et al., 2017). h_{*_new} is an attention-weighted context representation.

This chapter proposes to incorporate multiple layers of attention mechanisms as illustrated in Figure 6.3. The **first layer** is applied on the top of two separate the stacked LSTM networks described in Section 6.2.3. The **second layer** is applied to the joint representation of CC and CM.

Specifically, the first layer contains two sub-layers of attention mechanisms, each of which is respectively applied on the top of CC encoder output $\mathbf{H}_{cc}^k$ and CM encoder output $\mathbf{H}_{cm}^k$. Formally, this procedure can be formulated as Eq. 6.6 and 6.7. The two independent attention models are trained and output an attention weight for each hidden state. Subsequently, the hidden states of the two separate recurrent layers are weighted by attention weights. Outputs of the two attention models are denoted by $\mathbf{H}_{cc\_\text{new}}^k$ and $\mathbf{H}_{cm\_\text{new}}^k$. The attention-weighted hidden state vectors for all time steps from two context encoders are then concatenated. The aggregated representation is used as input to the second attention layer.

$$\mathbf{H}_{cc\_\text{new}}^k = attention_1(\mathbf{H}_{cc}^k) \tag{6.6}$$

$$\mathbf{H}_{cm\_\text{new}}^k = attention_1(\mathbf{H}_{cm}^k) \tag{6.7}$$



(a) Attention mechanisms for CC embeddings

(b) Attention mechanisms for CM embeddings

(c) Attention mechanisms for joint context representation

Figure 6.3: Stacked Soft Attentions

The second attention layer is introduced to jointly learn correlations between two types of contextual information. Different from the first layer that outputs attention-weighted hidden states for all elements in an input context sequence, the second layer computes a weighted sum of all attention-weighted hidden states. Eq. 6.8 and 6.9 formulate attention mechanisms in the second layer. The output vector will be fed into a fully-connected feed-forward network for classification. Finally, masked layer normalisation is applied before a dense layer in order to stabilise the training. This masking ensures that no update of weights is applied for padded elements.

$$\mathbf{H}^{k}_{\text{cxt\_new}} = attention_2(\mathbf{H}^{k}_{\text{cxt}}) = attention_2(\mathbf{H}^{t}_{cc\_new} \oplus \mathbf{H}^{t}_{cm\_new}) \tag{6.8}$$

$$\mathbf{V}^{k}_{\text{cxt}} = \sum_{t=1}^{j} h^{t}_{\text{cxt\_new}} \tag{6.9}$$

where $\oplus$ denotes concatenation and $\mathbf{V}^{k}_{\text{cxt}}$ is a final context vector. The intuition behind the use of the second layer is that it allows the architecture to learn fine-grained interactions between CC and CM. It facilitates the efficient and simultaneous representation of public opinions and diffusion patterns of public engagement.

The proposed stacked soft attention mechanisms are devised in the hope that they can incorporate context contents and auxiliary information into a unified framework, and consequently, achieve SOTA performance in ERD based on propagation contexts.

6.2.5 *Tweet Content Encoder*

Several studies on rumour detection (Zubiaga et al., 2018a; Kwon et al., 2017; Qazvinian et al., 2011) have demonstrated the effectiveness and advantages of using tweet text content for identifying emerging rumours on social media. For instance, some linguistic signals, e.g. "reportedly" and "I hear that", can effectively indicate the uncertainty of a candidate tweet (Zubiaga et al., 2017). For tweets that do not have sufficient signals, how people react to them can provide useful information. As introduced in Section 2.1.3, user reactions to rumours have extensively been studied to identify rumour-bearing tweets. Such reactions can be categorised into seven categories including *misinformation*, *speculation*, *correction*, *question*, *hedge*, *unrelated*, and *neutral/others* (Maddock et al., 2015; Zubiaga et al., 2018a).

In the RP-DNN, tweet content embeddings are obtained via ELMo (Peters et al., 2018), a SOTA context-aware NLM. The RP-DNN allows ELMo to learn linguistic and semantic signals for rumours from tweet contents without any hand-crafted features. It represents each word in an input corpus while considering the context of the entire corpus. The weight of each hidden state in ELMo is task-specific and can be learnt from domain-specific corpora. The ELMo employed in the RP-DNN is first fine-tuned for the task of rumour detection on social media (see Chapter 4). Specifically, ELMo is firstly pre-trained on 1 billion word benchmark corpus with vocabulary of 793, 471 tokens, and then fine-tuned on a large credibility-focused Twitter corpus with 6, 157, 180 tweets with 146, 340, 647 tokens and 2, 235, 075 vocabularies. The experimental results in Chapter 4 showed that the fine-tuned ELMo achieves low perplexity on domain-specific data sets and helps to achieve SOTA performance in tweet-level rumour detection based on textual contents. Representations from all three layers of the ELMo model for token embeddings are averaged out to provide the final representations of input tweets.

6.2.6 *Contextual Information*

The proposed architecture leverages 27 hand-crafted features grouped into two classes (i.e. tweet-level and user-based features) to provide auxiliary information to the RP-DNN. Table 6.1 describes each feature. Most of them are metadata provided by Twitter's API. Early work on rumour detection employs supervised learning techniques, and thus has extensively studied manually curated features related to contents, users, and networks to seek distinguishing features of online rumours (Qazvinian et al., 2011; Kwon et al., 2017; Yang et al., 2012; Sun et al., 2013; Zhao et al., 2015; Zhang et al., 2015c; Wu et al., 2015; Ma et al., 2015; Liu et al., 2016; Zubiaga et al., 2017; Hamidian and Diab, 2016). These studies have shown that hand-crafted features have the potential for distinguishing rumours from non-rumours. Recently, modern representation learning techniques such as deep learning architectures have been increasingly popular within the community of rumour detection. Although they need little or no feature engineering, some work (Ruchansky et al., 2017; Liu and Wu, 2018; Guo et al., 2018) has investigated whether the inclusion of hand-crafted features into neural architectures can provide significant improvements to SoA performance.

In particular, a recent study (Kwon et al., 2017) has demonstrated that linguistic and user features are good signals for ERD, while network and temporal features play a significant role in rumour detection over longer time periods. Based on a thorough search of related research, this chapter explores the effects of 27 features which are expected to provide weak signals for rumours in the early stages of their diffusion.

Table 6.1: Description of metadata used to represent the social context of source tweets.

Tweet-level features
Number of retweets
Number of favourites
Whether tweet has a question mark
Whether tweet is a duplicate of its source
Whether tweet contains URLs
Number of URLs embedded in tweet
Whether tweet has native media*
Number of words in tweet except source author's screen name
User-level features
Number of posts user has posted
Number of public lists user belongs to
Number of followers
Number of followings
Whether user has a background profile image
User reputation (i.e. followers/(followings+1))
User reputation (i.e. followers/(followings+followers+1))
Number of tweets user has liked so far (aka "user favourites")
Account age in days
Whether user is verified
User engagement (i.e. # posts / (account age+1))
Following rate (i.e. followings / (account age+1))
Favourite rate (i.e. user favourites / (account age+1))
Whether geolocation is enabled
Whether user has a description
Number of words in user description
Number of characters in user's name including white space
Whether user is source tweet's author
Response time (time difference between a reply and its source tweet in mins)

* multimedia shared with the Tweet user-interface not via an external link

6.2.6.1 *Tweet-Level Features*

The proposed method lets a NLM automatically learn syntactic and semantic representations of source tweets and replies. Therefore, tweet-level hand-crafted features are mainly related to URLs and multimedia embedded in tweets. Twitter users often use URLs as additional references due to a character limit (Qazvinian et al., 2011). The presence of URLs in tweets affects the credibility of messages as well as the behaviour of users. For instance, the inclusion of URLs tends to encourage more people to share rumours (Tanaka et al., 2014). URLs help to increase the trustworthiness of tweets as they provide users with supporting materials such as photos and videos (Gupta and Kumaraguru, 2012; Castillo et al., 2011). Some previous work studied behavioural patterns of social media posts containing external links. Friggeri et al. (2014) report that unverified information with links to websites for validating and debunking rumours often go viral on social media. Maddock et al. (2015) find that the first burst of the popularity of a rumour is primarily due to an increase in its textual variations, and shortly afterwards, variations of the original rumour linked to photos form the second burst.

6.2.6.2 *User-Level Features*

Rumour spreaders are individuals who seek attention and reputation (Sunstein, 2010). This has motivated researchers to delve into user features in the hope that they help to characterise online rumours. User features are key signals for rumours in their early stages along with linguistic features (Kwon et al., 2017). Features related to user profiles and reactions significantly contribute to the characterisation of rumours (Liu et al., 2015). Some studies have found that rumours tend to spread from low-impact users to influencers, whereas non-rumours have the opposite tendency (Ma et al., 2017; Kwon et al., 2017). Another study on the impact of users on rumour propagation reports that trustworthy sources such as mainstream media and verified users participate in rumour spreading by simply sharing rumour and maintaining neutrality (Li et al., 2016). The proposed methods exploit user features which are extracted from Twitter user account metadata.

6.2.7 *Model Training*

All the parameters of stacked LSTM and attention weights are trained by employing the derivatives of the cross-entropy loss function through back-propagation. The AdaGrad algorithm (Duchi et al., 2011) is used for parameter optimisation. The length of each ELMo content embedding is 1024 and that of each metadata feature vector is 27. The number of forward LSTM layers in each stacked LSTM is set to 2 and that of hidden units is set to twice input size. The learning rate and weight decay are set to $1e-4$ and $1e-5$, respectively. All training instances (i.e. SC) with corresponding context inputs (i.e. CC and CM) are iterated over in each epoch where batch size is 128. The number of epochs is set to 10 to reduce overfitting. Leaky ReLU (LReLU) is employed in 3 dense layers. Drop out rates 0.2, 0.3 and 0.3 are respectively applied after each of the three layers. Preliminary results show that the RPDNN suffers from the “dying ReLU” problem (Maas et al., 2013), which means weights in

NNs always drive all inputs to ReLU neurons to negative. This is problematic because ReLU neurons will always output 0 and will no longer useful in discriminating the input. Replacing ReLU with LReLU fixes the problem as it gives non-zero gradients for negative values. Models are implemented using Python 3.6, Allennlp (0.8.2) framework, and Pytorch 1.2.0. They are trained on one Tesla P100 SXM2 GPU node with maximum 16GiB RAM.

6.3 EXPERIMENTS

6.3.1 *Data*

The experiments use three data sets to generate training, hold-out, and test sets. A general description of each data set is as follows:

1. **PHEME (6392078; Kochkina et al. (2018a))**: This consists of manually labelled rumour and non-rumour source tweets and their replies for 9 breaking news events. It is used to generate test sets during evaluation.

2. **Aug-PHEME-filtered (i.e. the outcome of the Chapter 4)**: This is an augmented version of the *PHEME (6392078;* Kochkina et al. (2018a)) in Chapter 4 and its details are described in Section 4.6.2.2. To recap, it contains rumour and non-rumour source tweets and their contexts (i.e. replies and retweets) associated with six real-world breaking news events. Source tweets are labelled with weak supervision. In the experiments, the data sets filtered based on posting times are used. Specifically, source tweets posted before the occurrence date of each event are filtered out, and then contexts posted within the first seven days of the creation of their source tweets remain.

3. **Twitter15/16 (Ma et al., 2017)**: This data was introducted in Section 2.4.2. To recap, these two data sets consist of rumour and non-rumour source tweets and their replies. Replies of each source tweet are provided in the form of propagation trees. Source tweets are manually annotated with one of the following four categories: non-rumour, false rumour, true rumour and unverified rumour. As the experiment setup of this chapter is restricted to binary data sets, all but "non-rumour" class are aggregated into "rumour"class.

6.3.2 *Data Collection*

The *Aug-PHEME-filtered* is available on https://zenodo.org/record/3269768. As for the *PHEME(6392078)* and *Twitter 15/16*, source tweets and replies are downloaded by following the practice introduced in Section 4.2.5. In brief, source tweets are downloaded using an open source tweet collector called *Hydrator* [1]. Replies are collected via a HTML parsing technique implemented using Python libraries *Selenium* [2] and *BeautifulSoup* [3] and Twitter's API. Detailed procedure is described in Section 4.2.5.

[1] available via http://github.com/DocNow/hydrator

[2] available via http://selenium-python.readthedocs.io/

[3] available via http://www.crummy.com/software/BeautifulSoup/bs4/doc/

6.3.3 *Data Preprocessing*

All tweets are lowercased. Retweet symbols (i.e. "rt @"), URLs, user mentions, and special characters (e.g. !, ?, # etc.) are removed to reduce noise in tweet texts. Tweets with a minimum of 4 tokens are considered as tweets which lack enough textual features are generally unremarkable and add noise to data (Ifrim et al., 2014). This chapter aims to examine the effectiveness of leveraging conversational threads in providing contextual information to a rumour detection architecture, and therefore it is required that input source tweets have sufficient contexts. To ensure this, source tweets which have more than 5 replies remain in training, hold-out, and test sets. The maximum number of replies for each source tweet is set to 200.

6.3.4 *Leave-One (Event)-Out Cross-Validation (LOOCV)*

This section details the evaluation setting employed in this chapter to address the RQ 3.2. Several SOTA studies on rumour detection (Ma et al., 2016; Liu and Wu, 2018; Chen et al., 2018; Ma et al., 2018b; Zhou et al., 2019b; Tarnpradab and Hua, 2019) adopt conventional k-fold CV with different subset sizes k to estimate their models' performance. This practice allows that training and test sets have similar properties, and therefore, usually leads to good performance. However, it is not sufficient when ML models are required to generalise beyond the distribution of a training set. Generalisability is of paramount importance for ERD, of which the goal is to detect unseen rumours as early as possible. To achieve this aim, this chapter adopts LOOCV as an approximate evaluation of the proposed solution in realistic scenarios (Kochkina et al., 2018a). LOOCV has been exploited in a few studies on message-level rumour detection as described in Section 2.3. This thesis evaluated the impact of augmented rumour data on deep learning-based rumour detection in the LOOCV setting in Chapter 4.

RQ 3.2: How can rumour detection architectures be evaluated in realistic scenarios in which detection models are required to identify unseen rumours?

Table 6.2 presents the statistics of all event data sets filtered based on *posting time* (i.e. keeping source tweets posted after the occurrence of each event and replies posted within the first seven days of the creation of their source tweets) and *context size* (i.e. keeping source tweets with more than 5 replies). The "Mean tidff (hrs)" stands for the average time interval (in hours) between the posting time of each source tweet and that of each of its replies in each event data set. Overall procedure is similar to k-fold CV. Training and hold-out sets include samples automatically labelled with weak supervision (i.e. the output of Chapter 4), while test sets contain only manually labelled samples (PHEME (6392078; Kochkina et al. (2018a))). All data samples are randomly shuffled regardless of their types (i.e. a rumour and non-rumour).

12 data sets covering real-world events are used to generate training and hold-out sets. Specifically, 6 events in the *Aug-PHEME-filtered*, 4 events related to preselected rumours in the *PHEME (6392078; see Section 2.4.1)*, and 2 data sets in the *Twitter15/16* are used. Model parameters are fine-tuned using a hold-out set. As for test sets, 5 events except preselected rumours in the *PHEME (6392078)* and 2 sets in the *Twitter15/16* are used. In other words, 7 events in total are used to generate test sets, and thus 7−fold LOOCV is

performed. The 4 preselected rumour events in the *PHEME (6392078)* are not used as test sets because they are too small or contain only one class.

For each test event, 11 training events except itself are shuffled and split into training and hold-out sets in a ratio of 9 to 1. Class distributions in all data sets are balanced. The statistics of training, hold-out, and test sets are presented in Table 6.3, 6.4, and 6.5, respectively. Four evaluation metrics–precision, recall, F1, and accuracy–are adopted in the experiments for evaluation. All but accuracy are computed with respect to the positive class, i.e. rumour.

Table 6.2: Statistics of 12 data sets used for generating training and hold-out sets (R: rumours, NR: non-rumours, Med.: Median).

Event			**Replies**							
	# of sources (R)	**# of sources (NR)**	**Total**	**Mean**	**Min**	**Max**	**Med.**	**Mean tdiff (hrs) R**	**Mean tdiff (hrs) NR**	**Mean tdiff (hrs) R+NR**
charliehebdo	382	1,356	42,081	24	6	341	19	2.0	6.0	5.3
ferguson	266	746	26,565	26	6	288	18	7.8	6.7	7.0
germanwings	132	122	4,163	16	6	109	14	4.8	3.3	4.2
sydneysiege	480	784	26,435	21	6	341	17	1.6	3.0	2.5
ottawashooting	361	539	16,034	18	6	208	13	1.7	5.4	3.8
bostonbombings	75	584	23,210	35	6	207	20	1.9	5.8	5.5
ebola	13	0	208	16	6	26	15	1.2	0.0	1.2
gurlitt	1	1	23	12	7	16	12	0.3	2.8	1.0
prince	43	0	452	11	6	21	10	2.2	0.0	2.2
putinmissing	22	9	379	12	6	25	10	3.7	4.2	3.9
twitter15	782	323	47,324	43	6	458	28	89.4	46.0	74.8
twitter16	410	191	27,732	46	6	458	29	117.8	333.3	201.5
Total	2,967	4,655	214,606							

Table 6.3: Statistics of the training sets with balanced rumour and non-rumour source tweets (R: rumours, NR: non-rumours, Med.: Median).

Event			**Replies**							
	# of sources (R)	**# of sources (NR)**	**Total**	**Mean**	**Min**	**Max**	**Med.**	**Mean tdiff (hrs) R**	**Mean tdiff (hrs) NR**	**Mean tdiff (hrs) R+NR**
charliehebdo	2,337	2,337	138,777	30	6	250	18	64.7	58.2	61.4
ferguson	2,409	2,409	138,376	29	6	458	19	65.9	8.6	36.8
germanwings	2,572	2,572	147,378	17	6	458	19	49.8	51.1	50.5
sydneysiege	2,237	2,237	134,830	30	6	458	19	68.5	57.9	63.2
ottawashooting	2,338	2,338	140,325	30	6	458	19	50.4	9.2	29.1
twitter15	1,962	1,962	99,938	25	6	458	18	39.6	5.9	21.6
twitter16	2,300	2,300	124,052	27	6	458	18	41.7	7.6	24.0

Table 6.4: Statistics of the hold-out sets with balanced rumour and non-rumour source tweets (R: rumours, NR: non-rumours, Med.: Median).

Event			Replies							
	# of sources (R)	# of sources (NR)	Total	Mean	Min	Max	Med.	Mean tdiff (hrs) R	Mean tdiff (hrs) NR	Mean tdiff (hrs) R+NR
charliehebdo	248	248	14,375	29	6	250	19	28.9	7.1	17.6
ferguson	292	292	16,654	29	6	209	19	10.9	22.7	17.1
germanwings	263	263	15,860	30	6	250	19	123.4	5.9	67.5
sydneysiege	250	250	14,292	29	6	324	19	8.6	8.9	8.7
ottawashooting	268	268	16,362	31	6	458	19	148.3	8.3	77.4
twitter15	223	223	11,833	27	6	341	18	5.4	6.6	6.0
twtiter16	257	257	14,576	28	6	341	18	42.9	25.0	33.6

Table 6.5: Statistics of the test sets with balanced rumour and non-rumour source tweets (R: rumours, NR: non-rumours, Med.: Median).

Event			Replies							
	# of sources (R)	# of sources (NR)	Total	Mean	Min	Max	Med.	Mean tdiff (hrs) R	Mean tdiff (hrs) NR	Mean tdiff (hrs) R+NR
charliehebdo	340	340	15,515	23	6	341	19	2.0	5.0	3.6
ferguson	233	233	12,159	26	6	229	19	8.1	7.0	7.6
germanwings	106	106	3,627	17	6	109	15	5.1	3.2	4.3
sydneysiege	418	418	19,666	24	6	209	19	1.6	3.3	2.5
ottawashooting	289	289	11,944	21	6	208	17	1.8	3.0	2.4
twitter15	323	323	27,521	43	6	268	28	119.8	46.0	79.2
twtiter16	191	191	16,860	44	6	324	29	127.8	333.3	249.9

6.3.5 *K-Fold Cross-Validation (CV)*

In addition to LOOCV, the proposed models are also evaluated via 3-fold and 5-fold CV following common practice in the field of rumour detection. This chapter performs k-fold CV in order to provide a comparative evaluation with mainstream models. More importantly, a comparison between the performance of LOOCV and CV will show key challenges of ERD compared with other classification problems. The same four metrics as LOOCV (i.e. precision, recall, F1-score, and accuracy) are used to assess the performance of the proposed models.

For the 3-fold CV, the Twitter15/16 is randomly divided into three folds following the practice introduced in (Liu and Wu, 2018). 10% of the data is used as a hold-out set and the remainder is split into training and test sets in a ratio of three to one regardless of events. For evaluation, the average of the testing metrics of the 3 CV models is computed. The results are shown in Table 6.6.

For the 5-fold CV, all rumour and non-rumour source tweets in the Aug-PHEME-filtered (see Table 6.2) are aggregated irrespective of events, balanced, and shuffled. Training, hold-out, and test sets are generated via stratified sampling which ensures that the percentage of samples for each class (i.e. 50% as the data was balanced in advance) are preserved in returned folds. In each fold, the ratio of the number of source tweets in a *training set* to that in a *hold-out set* to that in a *test set* is $18:1:1$. Each fold's training set includes $4,382$ source tweets (i.e. $2,191$ rumours and $2,191$ non-rumours), and hold-out and test sets respectively contain 246 source tweets (i.e. 123 rumours and 123 non-rumours). For evaluation, the average of the testing metrics of the 5 CV models is computed. The results are shown in Table 6.7.

6.3.6 *Baselines*

This section introduces SOTA baselines leveraging context for message-level rumour detection. A detailed review of each model is introduced in Section 2.3.3. Since source code for most models is not publicly available, classification results reported in the original work are used for evaluation. On top of the these baselines, several variations of the RP-DNN are employed as baselines (see Section 6.3.7).

- **Zubiaga et al. (2017):** Conditional Random Fields (CRFs) utilise a sequence of rumour and non-rumour source tweets that have been posted up to the posting time of a tweet to be classified. CRFs perform classification based on content-based features extracted from tweet texts and social features obtained using the metadata of users. The reported evaluation results are precision, recall, and F1-score computed only for the positive class (i.e. rumour).

- **Model in Chapter 4:** A multi-task learning architecture proposed by Kochkina et al. (2018a) is modified for rumour detection only. The original model is based on LSTM and jointly performs rumour detection, stance classification, and rumour verification. Conversational threads (i.e. replies) of source tweets are leveraged as contexts. The four metrics are computed only for the positive class (i.e. rumour).

- **Ma et al. (2017):** A kernel-based SVM utilises propagation tree kernels built using rumour and non-rumour source tweets and their retweets and replies. F1-score for each class is reported.

- **Ma et al. (2018b):** Recursive NNs exploit tree structures of rumour diffusion (i.e. relations between source tweets and their contexts (i.e. replies)). F1-score for each class is reported.

- **Liu and Wu (2018):** A hybrid of CNNs and RNNs identifies rumour source tweets based on features of users who have participated in rumour spreading. F1-score for each class is reported.

- **Veyseh et al. (2019a):** A context-aware framework based on self-attention mechanisms exploits semantic similarity between all tweets in each source post's conversational threads for rumour identification. F1-score for each class is reported.

As for data sets for experiments, Zubiaga et al. (2017) use the *PHEME5* data. Han et al. (2019a) use the *Aug-PHEME-filtered*. Ma et al. (2018b), Ma et al. (2017), Liu and Wu (2018), and Veyseh et al. (2019a) use the *Twitter15/16*. As for training and evaluation settings, LOOCV is performed in (Zubiaga et al., 2017; Han et al., 2019a). The others perform k-fold CV. Liu and Wu (2018) use 10% of the entire data as a hold-out set and split the remainder into training and test sets in a ratio of three to one, regardless of events. Ma et al. (2017) use 10% of the entire data as a hold-out set and perform 3-fold CV for the remainder. Ma et al. (2018b) and Veyseh et al. (2019a) do not use hold-out sets and perform 5-fold CV.

6.3.7 *Ablation Study*

An ablation study is conducted to investigate the effects of different types of representations on message-level ERD. To this end, 8 configurations are investigated. The results are presented in Table 6.8 and 6.9.

- **RPDNN:** This is the fully configured architecture.
- **RPDNN -CXT:** This is the RP-DNN which is solely based on source tweet contents.
- **RPDNN -SC:** This is the RP-DNN which is solely based on two types of contextual information (i.e. CC and CM).
- **RPDNN -CC:** This is the full model excluding CC.
- **RPDNN -CM:** This is the full model excluding CM.
- **RPDNN -ATT:** This is the full model without stacked attention mechanisms. The last hidden state of stacked LSTM is fed into a fully connected layer for classification.
- **RPDNN -SC -CC**: This is the RP-DNN which is solely based on CM.
- **RPDNN -SC -CM**: This is the RP-DNN which is solely based on CC.

6.3.8 *Study of the Impact of Varying Context Lengths*

Kwon et al. (2017) reported that the impact of different types of hand-crafted features of rumours changes over time. Their experiments showed that linguistic and user features are good signals for ERD. On the other hand, network and temporal features play a significant role in rumour detection over longer time periods. Their findings open the following research questions: “Does context size (i.e. the number of replies) for an input source tweet affect the convergence and performance of rumour detection models?” and “What is the minimum/optimum size for ERD?” To address these questions, the context only model (RPDNN -SC) is evaluated over varying context sizes via LOOCV. Note that replies posted within the first **seven** days of the creation of their source tweets are kept in training data (Section 6.3.4) and the maximum context sequence length (i.e. the maximum number of replies for a source tweet) is set to 200 for training the RPDNN- SC. In this experiment, context sizes (i.e. time intervals for filtering out replies) are varied to investigate their impact on results. Context sizes in “minutes” used in the experiment are as follows: 5, 10, 15, 30, 45, 60, 90, 120, 180, 240, 360, 480, 600, 720, 840, 960, 1080, 1200, 1440 (1 day), 1800, 2160, 2520, 2880 (2 days), 3240, 3600, 3960, 4320 (3 days), 4680, 5040, 5400, 5760 (4 days), 6120, 6480, 6840, 7200 (5 days), 7560, 7920, 8280, 8640 (6 days), 9000, 9360, 9720, and 10080 (7 days). For instance, when the context size is set to 5, the pre-trained RPDNN -SC is evaluated using replies which were posted within the 5 minutes since the creation of their source tweets.

6.4 RESULTS

6.4.1 *Classification Performance*

6.4.1.1 *Overall Results*

This section presents overall evaluation results. The full model (i.e. RP-DNN) achieves average F1-scores of 0.915, 0.826 and 0.727 in 3-fold CV, 5-fold CV and 7-fold LOOCV, respectively. As expected, LOOCV is a stricter evaluation approach for assessing the performance of rumour detection models. The overall performance of the RP-DNN and its variations is lower in the LOOCV setting because they classify source tweets which have different characteristics (e.g. propagation patterns, linguistic features, etc.) from data learnt during training in LOOCV.

The 3-fold CV results shown in Table 6.6 demonstrate the performance of the RP-DNN compared with the SOTA models. Veyseh et al. (2019a), Liu and Wu (2018), Ma et al. (2018b), and Ma et al. (2017) report F1-score for each of the four classes for the *Twitter15/16* data. To compare results for binary classification, the average F1-score of three categories of rumours (i.e. false, true, and unverified rumours) is reported in the table. The RP-DNN outperforms the SOTA models. In particular, it improves the F1-score of the most competitive model (Liu and Wu, 2018) by 7.2%.

The 5-fold CV results shown in Table 6.7 demonstrate the performance of the RP-DNN and its variations. The performance of the RP-DNN is lower than that in the 3-fold CV. The most probable reason for this is the RP-DNN was trained on weakly labelled, augmented data which is larger, noisier and lower-quality, but covers a broader range of events and topics rather than the Twitter15/16. Therefore, it is expected that the RP-DNN learnt more types of rumours' propagation patterns and linguistic characteristics compared with the RP-DNN and SOTA models trained on small, manually labelled data sets.

Table 6.8 shows overall LOOCV performance. The values of the four evaluation metrics are the mean scores of all LOOCV iterations presented in Table 6.9. In comparison with the two SOTA baselines conducting LOOCV, the proposed models achieve the best performance in terms of all the four metrics. In particular, the full model RP-DNN is the best-performing model in terms of F1-score. It improves the F1-scores of (Han et al., 2019b) and (Zubiaga et al., 2017) by 7.1% and 12.6%, respectively. It is worth noting that the model solely based on conversational contexts (i.e. "RPDNN -SC") outperforms the most competitive baseline Han et al. (2019b), increasing F1-score by 3.8%. The RP-DNN and its variations achieve *relatively* low precision and high recall in relation to the SOTA baselines. This indicates that the RP-DNN is better at returning most of the relevant results (i.e. rumour sources) but is less effective

Table 6.6: Comparison of 3-fold CV results for the Twitter15/16 data.

Methods	Precision	Recall	F1	Accuracy
RP-DNN	0.852	0.989	0.915	0.872
Ma et al. (2017)	–	–	0.738	0.741
Liu and Wu (2018)	–	–	0.843	0.853
Ma et al. (2018a)	–	–	0.753	0.730
Veyseh et al. (2019a)	–	–	0.765	0.769

Table 6.7: Comparison of overall 5-fold CV results.

Methods	Precision	Recall	F1	Accuracy
RP-DNN	0.790	0.868	0.826	0.818
RPDNN - CXT	0.785	0.844	0.811	0.804
RPDNN - SC	0.730	0.839	0.780	0.762
RPDNN - CC	0.762	0.846	0.801	0.788
RPDNN - CM	0.754	0.868	0.805	0.789
RPDNN - ATT	0.766	0.847	0.803	0.792
RPDNN -SC-CM	0.779	0.733	0.754	0.762
RPDNN -SC-CC	0.624	0.597	0.609	0.617

in returning more relevant results than the irrelevant ones (i.e. non-rumour sources) compared with the baseline models.

Table 6.8: Comparison of overall LOOCV results.

Methods	Precision	Recall	F1	Accuracy
RP-DNN	0.648	0.834	0.727	0.684
RPDNN - CXT	0.626	0.863	0.725	0.669
RPDNN - SC	0.621	0.796	0.694	0.648
RPDNN - CC	0.631	0.800	0.705	0.654
RPDNN - CM	0.625	0.862	0.723	0.669
RPDNN - ATT	0.643	0.814	0.717	0.679
RPDNN -SC-CM	0.590	0.862	0.697	0.625
RPDNN -SC-CC	0.568	0.519	0.514	0.544
Han et al. (2019b)	0.716	0.614	0.656	0.685
Zubiaga et al. (2017)	0.692	0.559	0.601	–

Table 6.9 compares 7-fold LOOCV results of the proposed models and SoA baselines *per event*. The "Event" column in the table shows 7 different events used as a test set at each iteration of LOOCV. No work performed LOOCV with *Twitter15/16* data. Evaluation results vary among events. It is noteworthy that removing some features from the RP-DNN achieves improvements in most of the events. For example, the model without source tweet contents (i.e. "RPDNN -SC") achieves the best performance in terms of F1-score for the "fergusonunrest" and "Twitter 16". The effects of stacked attention mechanisms also vary among events. Overall, the proposed models achieve high recall. Except for the "germanwings" for which (Zubiaga et al.,

2017) is best-performing, the proposed models outperform the two SOTA baselines. It is also observed that they show relatively low performance on the “germanwings”, “Twitter 15”, and “Twitter16” events compared with their performance on the other events.

Event	Models	Precision	Recall	F1	Accuracy
charliehebdo	RPDNN	0.7426	0.8824	0.8065	0.7882
	RPDNN - CXT	0.7002	0.9000	0.7876	0.7574
	RPDNN - SC	0.7544	0.7588	0.7566	0.7559
	RPDNN - CC	0.7120	0.9235	0.8041	0.6979
	RPDNN - CM	0.7346	0.9441	0.8263	0.8015
	RPDNN - Att	0.7506	0.8676	0.8049	0.7897
	RPDNN -SC-CM	0.6974	0.8676	0.7733	0.7456
	RPDNN -SC-CC	0.5407	0.6441	0.5879	0.5485
	Han et al. (2019b)	0.723	0.817	0.767	0.752
	Zubiaga et al., 2017	0.545	0.762	0.636	–
ferguson	RPDNN	0.5903	0.8841	0.7079	0.6352
	RPDNN -CXT	0.6192	0.9142	0.7383	0.6760
	RPDNN -SC	0.6409	0.8884	0.7446	0.6953
	RPDNN -CC	0.5671	0.7983	0.6631	0.5944
	RPDNN -CM	0.5646	0.9571	0.7102	0.6094
	RPDNN -ATT	0.6265	0.6695	0.6473	0.6352
	RPDNN -SC-CM	0.5273	0.9957	0.6895	0.5515
	RPDNN -SC-CC	0.5957	0.2403	0.3425	0.5386
	Han et al. (2019b)	0.707	0.535	0.609	0.657
	Zubiaga et al., 2017	0.566	0.394	0.465	–
germanwings	RP-DNN	0.5940	0.7453	0.6611	0.6179
	RPDNN -CXT	0.5301	0.8302	0.6471	0.5472
	RPDNN -SC	0.4817	0.7453	0.5852	0.4717
	RPDNN -CC	0.5546	0.6226	0.5867	0.5613
	RPDNN -CM	0.5556	0.7076	0.6224	0.5708
	RPDNN -ATT	0.6015	0.7547	0.6695	0.6274
	RPDNN -SC-CM	0.5114	0.8491	0.6383	0.5189
	RPDNN -SC-CC	0.4924	0.6132	0.5462	0.4906
	Han et al. (2019b)	0.601	0.652	0.558	0.630
	Zubiaga et al. (2017)	0.743	0.668	0.704	–
ottawashooting	RP-DNN	0.6469	0.9446	0.7679	0.7145
	RPDNN -CXT	0.6717	0.9273	0.7791	0.7370
	RPDNN -SC	0.6050	0.9170	0.7290	0.6592
	RPDNN -CC	0.7427	0.8789	0.8051	0.7872
	RPDNN -CM	0.6500	0.9446	0.7701	0.7180
	RPDNN -ATT	0.6519	0.9135	0.7608	0.7128
	RPDNN -SC-CM	0.6154	0.8858	0.7262	0.6661
	RPDNN -SC-CC	0.6047	0.3599	0.4512	0.5623
	Han et al. (2019b)	0.850	0.710	0.770	0.800
	Zubiaga et al., 2017	0.841	0.585	0.690	–
sydneysiege	RP-DNN	0.7842	0.8086	0.7962	0.7931
	RPDNN -CXT	0.7190	0.8876	0.7944	0.7703
	RPDNN -SC	0.6745	0.8230	0.7414	0.7129

sydneysiege	RPDNN -CC	0.6728	0.8708	0.7591	0.7237
	RPDNN -CM	0.6834	0.8469	0.7564	0.7273
	RPDNN -ATT	0.6842	**0.9019**	0.7781	0.7428
	RPDNN-SC-CM	0.6341	0.8995	0.7438	0.6902
	RPDNN-SC-CC	0.6528	0.4139	0.5066	0.5969
	Han et al. (2019b)	0.755	0.644	0.695	0.717
	Zubiaga et al., 2017	0.764	0.385	0.512	–
Twitter15	RP-DNN	0.5903	0.7895	0.6755	0.6208
	RPDNN -CXT	0.5816	0.7833	0.6675	0.6099
	RPDNN -SC	0.5706	0.6130	0.5910	0.5759
	RPDNN -CC	0.5813	0.7307	0.6475	0.6022
	RPDNN -CM	0.5803	**0.8390**	0.6861	0.6161
	RPDNN -ATT	**0.5948**	0.7863	**0.6773**	**0.6254**
	RPDNN -SC-CM	0.5646	0.6904	0.6212	0.5789
	RPDNN -SC-CC	0.4792	0.7864	0.5955	0.4659
Twitter16	RPDNN	0.5882	0.7853	0.6726	0.6178
	RPDNN - CXT	0.5609	0.7958	0.6580	0.5864
	RPDNN - SC	**0.6220**	0.8272	**0.7101**	**0.6623**
	RPDNN - CC	0.5850	0.7749	0.6667	0.6126
	RPDNN - CM	0.6080	0.7958	0.6893	0.6414
	RPDNN - ATT	0.5885	0.8011	0.6785	0.6204
	RPDNN -SC-CM	0.5833	**0.8429**	0.6895	0.6204
	RPDNN -SC-CC	0.6023	0.8168	0.6933	0.6387

Table 6.9: LOOCV results

6.4.1.2 *Ablation Study Results*

This section further discusses the results shown in Table 6.7 and 6.8. Comparisons of the internal baseline models introduced for an ablation study demonstrate several interesting findings: 1) the textual content of candidate source tweets is the most important and influential signal for ERD; 2) stacked attention mechanisms improve model performance; and 3) conversational contexts can provide auxiliary information useful for ERD.

The first observation is consistent with several existing studies (Kwon et al., 2017; Castillo et al., 2011; Qazvinian et al., 2011; Kochkina et al., 2018a; Ma et al., 2018a; Veyseh et al., 2019a). A comparison between results obtained by the model without source tweet content (i.e. RPDNN -SC) and full model (i.e. RPDNN) shows that utilising SC increases F1-score by 4.6% and 3.3% in the CV and LOOCV, respectively. Also, the source content only model (i.e. RPDNN -CXT) achieves comparable performance with the full model; only differences of 1.5% and 0.2% in F1-score are observed in the CV and LOOCV.

Secondly, the benefits of incorporating stacked attention mechanisms into the proposed architecture are further justified based on a comparison between the "RP-DNN" and "RPDNN -ATT". The results show that they improve all evaluation metrics, in particular, F1-score by 2.3% and 1.0% in the CV and LOOCV, respectively. More detailed discussions regarding the stacked attention mechanisms and their outputs are described in Section 6.4.2.

As for the third finding, the context content only model (i.e. RPDNN -SC-CM) outperforms the SOTA baselines in the LOOCV setting. A comparison between the "RPDNN -CC" and "RP-DNN" shows that using context contents improve all the evaluation metrics. In particular, F1-score is increased by 2.5% and 2.2% for the CV and LOOCV, respectively. These results support previous findings that public opinion and self-correction appearing in conversational threads of rumour source tweets are useful for identifying rumours in the early stages of event diffusion (Kochkina et al., 2018a; Ozturk et al., 2015; Zhao et al., 2015). The context metadata model (i.e. RPDNN -SC-CC) achieves the lowest performance among all the models compared in the experiments because metadata contains limited information compared with textual contents. Although impact is marginal, combining CM with SC and CC shows improvements. This indicates that metadata and textual contents are correlated and play complementary roles (Kıcıman, 2010). Specifically, comparing "RPDNN -CM" with "RPDNN" shows that using metadata features improves precision and F1-score in both evaluation settings. A comparison between the "RP-DNN" and "RPDNN -CXT" shows that utilising contextual information can bring performance gains in ERD. Overall, the observations suggest that context content is more useful than context metadata, but both of them provide weak signals for rumours.

6.4.2 *Case Study: Analysis of Attention Degrees*

This section investigates attention degrees paid to each context (i.e. reply) of individual source tweets. In Figures 6.4-6.7, Figures 6.4a-6.7a illustrate replies for four rumour source tweets ranked according to attention weights learnt during LOOCV training and Figures 6.4b-6.7b show heatmaps of Pearson's correlation coefficients (PCCs) between 27 context metadata features and attention weights. For CM, the visualisation of PCCs is provided rather than raw CM values for each reply because the former provides a better and intuitive understanding of the relationship between context rankings and attention weights for CM. In other words, unlike context content (CC), it is difficult to find correlations between 27 context CM values and rankings based on raw values.

As described in Section 6.2.4, the *first layer* consists of two stacked attention mechanisms applied on the top of two stacked LSTM, each of which models CC and CM, respectively. In Figures 6.4a-6.7a, the "CC" and "CM" columns show attention weights obtained via the first layer and the rank of each reply according to the weights. The *second layer* is applied to the joint representations of CC and CM. The "CC+CM" column shows its attention weights and the rank of each reply. Numbers in the coloured cells denote the rank of each reply. The smaller a number is, the more important a reply is.

The results obtained by the second attention layer (i.e. CC+CM) show some characteristics of rumours found by previous research on rumour detection. Replies expressing doubts and/or questions (see Figures 6.4a, 6.5a, and 6.7a) and supports (see Figure 6.6a) tend to have high attention weights. It can also be observed that replies, which duplicate source tweet content and add exclamations (e.g. OMG) and comments, tend to be higher in rank. This supports previous findings that people generally tend to disseminate rumours without expressing their beliefs when they lack evidence to verify the rumour (Buckner, 1965; Li et al., 2016), and thus, new variants of rumours in the early stages of event diffusion are mostly textual variants of source tweets (Maddock et al., 2015; Zhao et al., 2015; Chen et al., 2018). Interestingly, for some replies, the first and second attention layers produce contradictory results, but the latter tends to output more logical results. For instance, the reply "@MailOnline @CathyYoung63" in Figure 6.7a is in the first rank according to the first layer's results. However, it does not contain any useful information and its metadata does not indicate that this tweet is particularly high-impact (e.g. a great number of retweets). It is ranked last by the second layer. This observation supports the motivation behind adopting multiple attention layers (Yang et al., 2016b; Wang et al., 2017), that is, they can progressively refine feature maps and focus on more salient features.

To investigate how the attention mechanisms for CC in the first layer perform, this chapter analyses relations between CC attention weights and textual contents of replies (see Figures 6.4a-6.7a). Previous research has discovered that linguistic signals are useful for the characterisation of rumours. For instance, users are more likely to use negative (e.g. not, never), cognitive (e.g. cause, know), doubtful (e.g. unsure, doubt), and tentative (e.g. perhaps, guess) expressions for rumours (Kwon et al., 2013; Kwon et al., 2017; Bahuleyan and Vechtomova, 2017). Certain phrases expressing enquiries for verification and corrections such as "is it true?", "unconfirmed", "reportedly", and "really?" appear in users' reactions in the early stage of rumour spreading (Zhao et al., 2015). Rankings based on CC attention weights show that the RP-DNN successfully learns such signals, and consequently, high attention weights are assigned to replies containing signal expressions such as "reporting", "not", and "doubt", and "true". However, reasons for some results are not entirely clear. For instance, the replies "@MailOnline @CathyYoung63" in Figure 6.7a and "@newscomauHQ still unverified footage" in Figure 6.5a have the highest CC attention weights in conversational threads to which they belong. However, their CC embeddings only contain zeros because they contain less than 4 tokens after removing user mentions (i.e. "@[username]"), see Section 6.3.3. It is worth noting that each of the two replies is the first reaction to the corresponding source tweet. Putting it all together, it could be inferred that it is a common pattern that very short reactions posted at a very early stage are useful signals for rumour-bearing tweets.

To gain a better insight regarding CM attention weights, PCCs between CM weights and metadata features (see Section 6.2.6) are computed and visualised in Figures 6.4b-6.7b. PCC is a measure of a linear correlation between two variables. It can range from -1 to 1, where 1, 0, and -1 indicate a perfectly positive linear, no linear, and perfectly negative linear relationships. The grey cells in the figures indicate that PCCs are not available for features which are constant across all replies in a thread. To identify features which are mostly correlated with CM and CC+CM attention weights respectively, features with the absolute values of PCCs above 0.4[4] are selected. The selected ones are sorted by their absolute values in descending order and are shown in Table 6.10. Features correlated with CM weights vary among source tweets. Overall, both tweet-level and user-based features are significantly correlated with CM attention weights for source tweets shown in Figures 6.4, 6.5, and 6.7, while only the user-based features have an impact on them for the source tweet in Figure 6.6. In contrast, PCCs between the output of the second attention layer (i.e. CC+CM attention weights) and features display a common pattern across different source tweets. Specifically, "response time (i.e. time difference between each reply and its source tweet in minutes)" and "profile name length (i.e. the number of characters in an author's name including white space)" are correlated with attention weights for the joint representations of contexts. Based on all the observations made using the four examples, stacked attention mechanisms proposed in this chapter are effective in paying more attention to key replies in conversational threads. In particular, the second attention layer is useful in jointly learning correlations between two types of contextual information and producing outputs close to human judgements. This is consistent with the finding of existing studies (Yang et al., 2016b; Wang et al., 2017) on advantages of stacking several attention mechanisms.

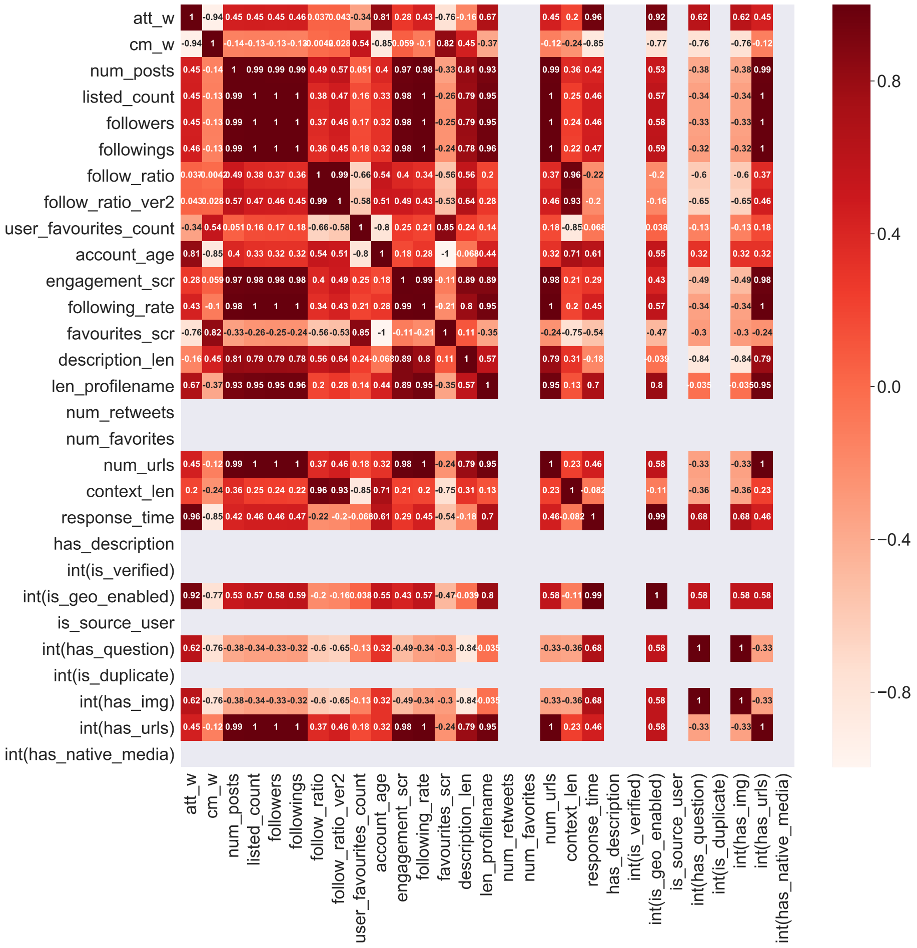
[4] There is no standard way to define thresholds for the strength of correlations (e.g. strongly correlated). For instance, some say a correlation of 0.4 is moderate (http://www.shortell.org/book/chap18.html), while others say it is weak (https://www.andrews.edu/~calkins/math/edrm611/edrm05.htm). The aim of this analysis is to study whether there is a common pattern of correlations between different source tweets. Considering the number of selected features, 0.4 is chosen.

Source tweet content

CXN: 1 gunman shot dead in Parliament Hill attack, soldier shot in Ottawa http://t.co/pp6hcfWcRw No report soldier dead. #OttawaShooting

	Attention weights					
Context content		**CC**		**CM**		**CC+CM**
@CBCNews ctv reporting he's alive	1	0.5611	1	0.2727	4	0.2027
@CBCNews stop showing clip of the CPR for the solider at the War Memorial on the online feed. Not necessary, old, move on.	2	0.1997	2	0.2508	3	0.2367
OMG! @CBCNews @tbeaudrymellor CXN: 1 gunman shot dead in Parliament Hill attack, soldier shot Ottawa http://t.co/rkzy5NRHq5 #OttawaShooting	3	0.1398	3	0.2467	2	0.2756
@CBCNews European news reporting soldiers gun was not loaded?	4	0.0993	4	0.2298	1	0.2849
Weight sum		0.9999		1		0.9999

(a) Analysis of attention weights for an example rumour source tweet.



(b) Heatmap of the PCC matrix of metadata features for an example rumour source tweet.

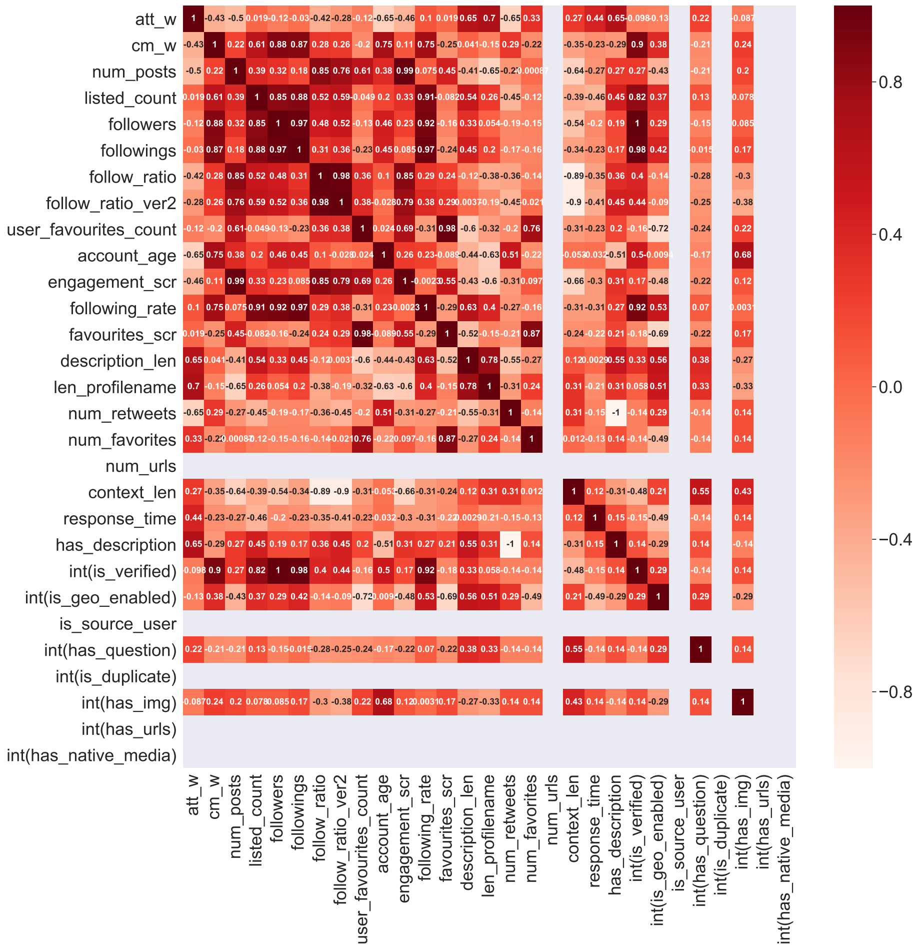
Figure 6.4: Visualisation of attention weights for an example tweet.

Source tweet content

Authorities collecting passports at #MH17 crash site. Australian coat of arms clearly visible. http://t.co/ai16vY46FV http://t.co/JA0gjQt3P5

	Attention weights					
Context content	**CC**		**CM**		**CC+CM**	
@newscomauHQ still unverified footage	2	0.2703	1	0.2015	6	0.1226
@newscomauHQ collecting... They were taking them and showing the cameras the faces of passengers and then throwing them back down. :(	1	0.3427	2	0.1493	8	0.1092
@newscomauHQ @Harriett_Bur it's not authorities...	3	0.1355	3	0.1154	7	0.1115
@newscomauHQ such heart breaking news!	4	0.0614	8	0.1043	5	0.1271
@newscomauHQ Is it just mean who finds these images disturbing. To what length would you have to go to have these passports in your hands?	5	0.0476	5	0.1074	4	0.1303
@newscomauHQ How do you identify the lost souls. They are people with families, probably going on holiday or business not war!	6	0.0475	4	0.1097	3	0.1310
@newscomauHQ Strange that passports look in very good condition when rest of plane demolished.	6	0.0475	7	0.1060	2	0.1329
@newscomauHQ why are they in such good condition reminiscent of the ones found on 9/11	6	0.0475	6	0.1063	1	0.1353
Weight sum		1		0.9999		0.9999

(a) Visualisation of attention weights for an example rumour source tweet.

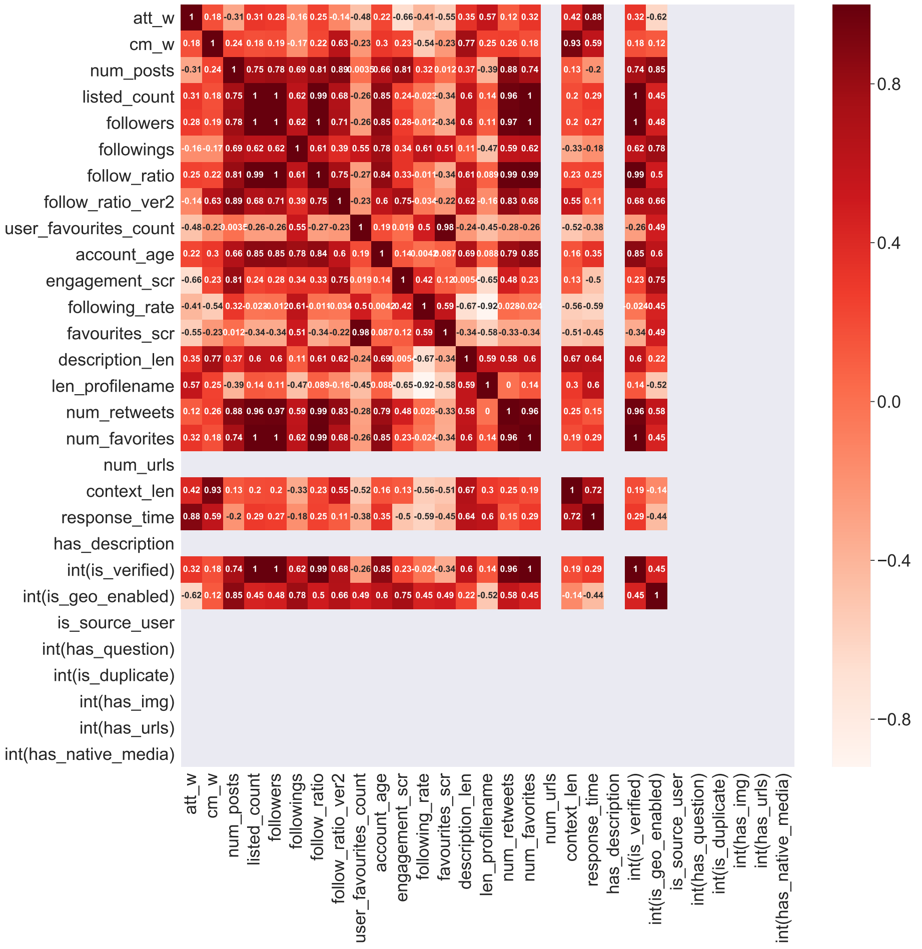


(b) Heatmap of the PCC matrix of metadata features for an example rumour source tweet.

Figure 6.5: Visualisation of attention weights for an example tweet.

Source tweet content						
Trocadero, it was a fasle alert (Ministry of Interior) Via @WilliamMolinie						
	Attention weights					
Context content	CC		CM		CC+CM	
Fausse alerte au Trocadero “@JulienPain: Trocadero, it was a fasle alert (Ministry of Interior) Via @WilliamMolinie”	6	0.1666	2	0.1811	6	0.1290
“@JulienPain: Trocadero, it was a fasle alert (Ministry of Interior) Via @WilliamMolinie”	1	0.1667	4	0.1518	5	0.1465
CHOUETTE “@JulienPain: Trocadero, it was a fasle alert (Ministry of Interior) Via @WilliamMolinie”	1	0.1667	5	0.1442	4	0.1657
“@JulienPain: Trocadero, it was a fasle alert (Ministry of Interior) Via @WilliamMolinie” @BuzzFeed @BuzzFeedNews	1	0.1667	6	0.1423	2	0.1829
Let’s hope so. MT @JulienPain Trocadero, it was a False alert (Ministry of Interior) Via @WilliamMolinie	1	0.1667	3	0.1756	3	0.1827
@acarvin Lines 6 and 9 stopped transit for a short while through trocadéro, now resumed, see official @Ligne6_RATP @Ligne9_RATP	1	0.1667	1	0.2051	1	0.1931
Weight sum		1.0001		1.0001		0.9999

(a) Analysis of attention weights for an example rumour source tweet.

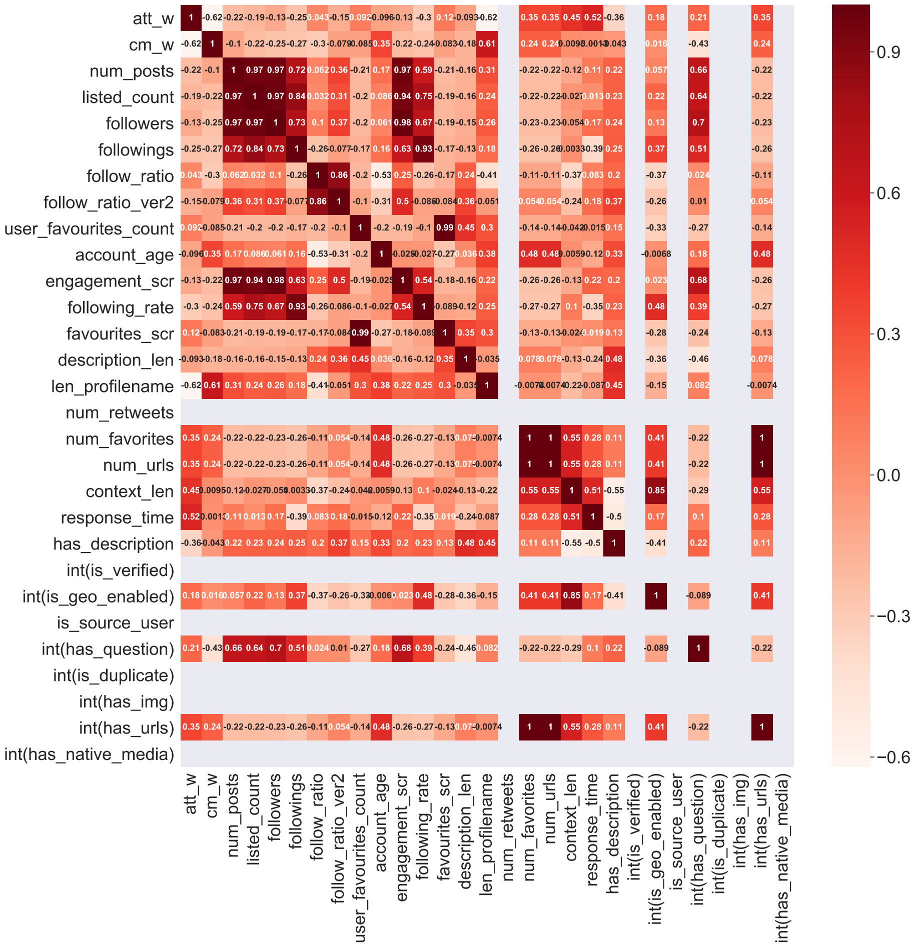


(b) Heatmap of the PCC matrix of metadata features for an example rumour source tweet.

Figure 6.6: Visualisation of attention weights for an example tweet.

Source tweet content						
Reports claim Putin disappeared due to impending political coup http://t.co/8IpndT2bsI						
	Attention weights					
Context content	**CC**		**CM**		**CC+CM**	
@MailOnline @CathyYoung63	1	0.2755	1	0.1203	10	0.0932
@MailOnline Ah yes to be closer to his billions of rubles	2	0.1386	3	0.1015	8	0.0966
@MailOnline Sure?	3	0.0775	8	0.0946	3	0.1023
@MailOnline Nothing to do with his wife giving birth then?	4	0.0731	10	0.0926	7	0.0998
@MailOnline That's stupid	5	0.0726	9	0.0928	6	0.1004
@MailOnline He should disappear 6 feet under.	5	0.0726	7	0.0963	8	0.0996
@MailOnline he has prolly been having a facelift	5	0.0726	5	0.0981	4	0.1012
Something big is happening right now in Moscow “@MailOnline: Putin disappeared due to impending political coup http://t.co/MKClBsKfvK”	5	0.0726	2	0.1055	2	0.1030
@MailOnline would be nice if it's true but I doubt it. Just one more of Putin's games.	5	0.0726	4	0.1010	1	0.1031
@MailOnline are we ready for war?	5	0.0726	6	0.0973	5	0.1007
Weight sum		1.0003		1		0.9999

(a) Analysis of attention weights for an example rumour source tweet.



(b) Heatmap of the PCC matrix of metadata features for an example rumour source tweet.

Figure 6.7: Visualisation of attention weights for an example tweet.

Table 6.10: Features significantly correlated with CM and CC+CM attention weights.

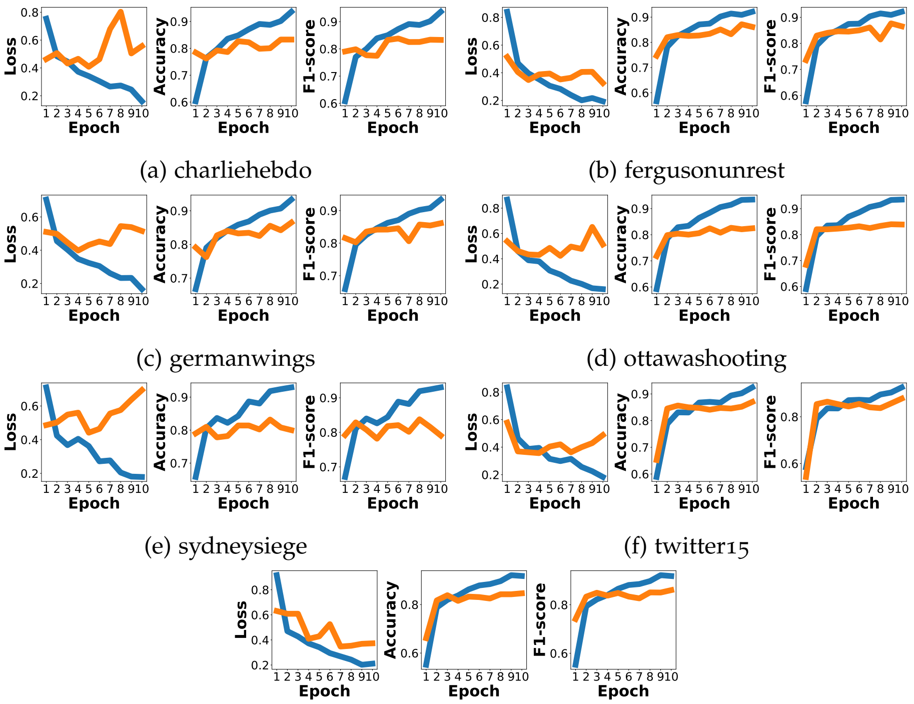
Dependent variable	Tweet 1 (Figure 6.4)	Tweet 2 (Figure 6.5)	Tweet 3 (Figure 6.6)	Tweet 4 (Figure 6.7)
CM weights	response_time*	context_len	is_verified	len_profilename
	account_age*	description_len	followers	has_question*
	favourites_scr	follow_ratio_ver2	followings	
	is_geo_enabled*	response_time	following_rate	
	has_question*	following_rate*	account_age	
	has_img*		listed_count	
	user_favourites_count			
	description_len			
CC+CM	**response_time**	**response_time**	**len_profilename**	**len_profilename***
	account_age	engagement_scr*	has_description	**response_time**
	favourites_scr*	is_geo_enabled*	num_retweets*	context_len
	len_profilename	**len_profilename**	description_len	
	has_img	favourites_scr*	account_age*	
	has_question	user_favourites_count*	num_posts*	
	followings	context_len	engagement_scr*	
	listed_count	following_rate*	**response_time**	
	followers		follow_rate*	
	num_urls			
	has_urls			
	num_posts			
	following_rate			

* negative correlation

6.4.3 *Training Loss and Performance Analysis*

To diagnose whether there are problems with learning such as overfitting for every test event in LOOCV, learning curves which show variations in the learning performance of models over time are plotted. Figure 6.8 presents loss, accuracy, and F1-score curves of the RP-DNN over 10 epochs for the 7-fold LOOCV. Loss is a measure of the difference between real values and predictions made by a model. In the experiments, cross-entropy is used. Blue and orange lines are graphs for training and validation sets, respectively. The average training time of the RP-DNN for all the folds is around 28 hours with GPU.

The results show that the "sydneysiege" event is the most difficult one to fit. The loss curve for this event widely diverges since the 5th epoch and the gap between training and validation sets in F1 and accuracy curves are larger than the other events. Increasing the number of epochs may improve learning for the "twitter16" and "fergusonunrest" because their validation losses tend to decrease during the 10 epochs. For all the events, the training loss curves tend to steadily decrease, while validation loss tends to fluctuate. Training accuracy and F1-score are constantly increasing and validation accuracy and F1-score tend to show significant increases during the first two epochs and stay constant afterwards, which is a sign of overfitting.



(a) charliehebdo

(b) fergusonunrest

(c) germanwings

(d) ottawashooting

(e) sydneysiege

(f) twitter15

(g) twitter16

Figure 6.8: Loss, accuracy, and F1-score curves for training (blue line) and validation (orange line) sets in each fold of the 7-fold LOOCV. For each event, loss (left), accuracy (centre), and F1-score (right) curves are visualised.

Overall, the gap between the training and validation loss and accuracy after $4-5$ epochs indicates that the RP-DNN model is overfitting training data. The RP-DNN already uses regularisation techniques including dropout and L2 regularisation (i.e. weight decay) to reduce overfitting. One of the major reasons for overfitting is limited labelled training data. Apart from regularisation, data augmentation is another effective way to reduce overfitting. This thesis already proposed this research topic in Chapter 4 and its outcome (i.e. *Aug-rnr* data) is used as training data in the LOOCV. However, the results presented in this section indicate that the augmented data is still not large enough. Generating larger data sets using the methodology proposed in Chapter 4 can be a promising approach to address this issue. As pointed out in Chapter 4, future research may extend the data augmentation framework to other events in a broad range of domains rather than breaking news events.

6.4.4 *Impact of Varying Context Lengths*

RP-DNN -SC: the RP-DNN which is solely based on two types of contextual information (i.e. CC and CM).

Figure 6.9 shows F1-scores achieved by the "RP-DNN -SC" model over varying context sizes. For most of the events, F1-scores increase up to 60 or 90 minutes and tend to stay constant for longer context sizes. This is intuitive because a model can have more information about how rumour sources propagate with longer context sizes (i.e. more replies). In contrast, F1-scores for the "twitter15" and "germanwings" data sets decrease up to 840 and 1800 minutes, respectively. These results indicate that different events propagate in different ways; in specific, conversational threads in the early stages of the diffusion of these two events may contain noisy and irrelevant

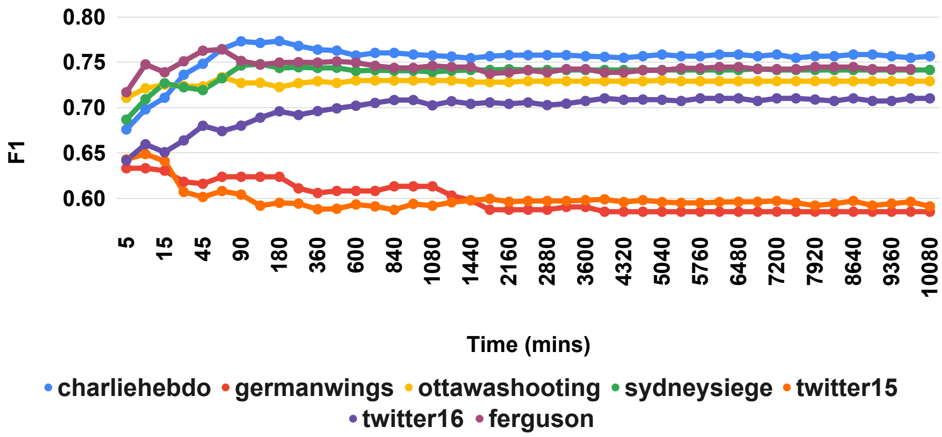


Figure 6.9: F1-scores achieved by the "RPDNN -SC" model over varying context sizes.

contents which reduce the effectiveness of rumour detection models. Filtering out such contents and only considering replies which can provide useful, discriminative signals for rumours can be done to tackle this issue in future work.

Overall, it is not always true that more context is beneficial to rumour detection in real worlds scenarios, which highlights a potential need for a more advanced reply filtering technique and/or the application of other new language models for more effective representations of tweets. The results also indicate that contextual information is one of several useful signals for rumours which can improve the effectiveness of ERD, and therefore, researching and identifying other signals will be helpful for ERD.

6.5 CONCLUSION AND FUTURE WORK

This chapter researched the task of message-level ERD. While event-level rumour detection requires large amounts of data reporting an event of interest to make judgements at an aggregate level, message-level rumour detection aims to identify rumour-bearing messages in the early development stages of events where highly limited information is available. The proposed solution is a novel hybrid NN architecture which combines SOTA deep contextualised BiLM for tweet text embeddings and stacked LSTM networks with attention mechanisms for representing the temporal, social, and textual dynamics of contexts. In more detail, the BiLM was fine-tuned specifically for the task of rumour detection. For social contexts (mainly Twitter metadata), which have extensively been studied and have shown the potential for distinguishing rumours from non-rumours, were extracted from conversational threads of source tweets. The motivation behind the inclusion of such domain-independent, hand-crated features is the desire to help a neural model effectively learn salient and implicit features of rumours and their propagation such as uncertainty, credibility, and virality at an individual post level. Unlike most SOTA work employing attention mechanisms at word level, this chapter employed context-level attention mechanisms to amplify the effects of certain replies which are important for characterising rumour sources and to filter out less significant information in the final representation of contexts.

The experiments were conducted on extensive data which contain 2,967 rumours covering 12 real-world events. To the best of my knowledge, this data is the largest data on which message-level rumour detection is performed and evaluated. This chapter adopted two approaches for evaluation. The first way was CV in which all data samples are shuffled and randomly split into training, hold-out, and test sets regardless of events. The second way was LOOCV which allows investigating the generalisability of the proposed model to unseen rumours. This evaluation approach assesses rumour detection models in a setting closer to real-world scenarios. The results showed that the proposed model advanced SOTA performance by achieving a F1-score of 0.727 in LOOCV, which is an increase of 7.1% compared to the best SOTA performance. This chapter conducted an ablation study to examine the relative contribution of each component of the proposed architecture. The results showed that the textual content of source tweets is the most important signal for ERD and context-level attention mechanisms and conversational contexts are useful in improving the effectiveness of the RP-DNN. A case study on attention weights demonstrated the effectiveness of stacked attention mechanisms in paying more attention to key replies in conversational threads by progressively refining feature maps. The experiment on the impact of varying context sizes illustrated that the F1-score of the context only model (RPDNN -SC) tends to converge after 180 minutes for most events.

The generalisability of a rumour detection model to new domains (e.g. politics) and the feasibility of end-to-end ERD are still great challenges to be investigated in future research. Possible future research directions include the exploration of the impact of word-level attention mechanisms and deeper NNs on message-level ERD. This chapter only investigated context-level attention mechanisms, i.e. determining the relative importance of each reply in a conversational thread. Some SOTA research on rumour detection (Guo et al., 2018; Chen et al., 2018) explored word-level attention mechanisms and have shown their effectiveness in the task. It could be researched whether adopting different types of attention mechanisms would be helpful. The proposed solution could not have deeper NNs due to limited training data although this was the first research which performed message-level ERD over data combining three publicly available data sets for rumour detection. Training loss and performance analysis results also indicated that larger data will be helpful for reducing overfitting. Stacked LSTM networks can be further explored in future work (e.g. more LSTM layers). In the mean time, generating larger training data with weak supervision can be investigated as stated in Section 4.8.

7 CONCLUSIONS

7.1 CONCLUSIONS

Online rumours are an active research area. This thesis has studied current challenges in online rumour research and how to develop an end-to-end architecture for it by focussing on the task of ERD. A recent study (Zubiaga et al., 2018a) proposed a rumour resolution pipeline consisting of four sub-tasks: rumour detection, tracking, stance classification and verification. Early work on online rumours usually aimed to address one of the four sub-tasks. Recently, multi-task learning aiming to perform several rumour-related tasks at the same time has become popular in the research community. However, connecting the dots between different sub-tasks to understand and build an end-to-end framework for rumour resolution remains to be researched extensively. Motivated by such a gap, this thesis extended the existing rumour resolution pipeline as illustrated in Figure 7.1.

This thesis started with a vital component of most ML tasks: data. The field of rumour detection suffers from a lack of larger training data. This thesis augmented one of the most popular data sets for message-level rumour detection by employing semantic relatedness as weak supervision and using a fine-tuned contextualised NLM. The augmented data was employed to evaluate generated potential rumours to demonstrate the proposed method can actually solve the intended research problem (Chapter 5). The data and fine-tuned NLM were also used in ERD in this thesis (Chapter 6), which distinguishes my research from SOTA work on ERD. Likewise, they can contribute to the other sub-tasks in the rumour resolution process such as rumour verification.

In the second part of this thesis, potential rumour detection was studied as a preliminary task (i.e. data reduction) for message-level ERD. In practice, identified potential rumours are input to an automatic rumour detection model. Data reduction has received minimal attention in rumour resolution process because most rumour studies pay more attention to achieving performance gains in specific tasks using publicly available data sets. In other words, they do not need to worry about curating training and test data sets out of immense amounts of raw data. However, solely relying on already available data sets is not enough, in particular, if they reach saturation points as demonstrated by almost perfect scores achieved by SOTA models. Therefore, this thesis explored domain-agnostic potential rumour identification which can provide meaningful candidates for rumours to both researchers and practitioners during training and validation stages of ERD. Based on an extensive search of related research and feature analysis in Chapter 2, this thesis hypothesised that temporal signals (key bursts) available during event diffusion are effective, early signals for potential rumours. Identifying potential rumours from key bursts was performed via extractive text summarisation. Extensive experiments with summarisation methods based on graph- and frequency-based keyword ranking algorithms proved the hypothesis.

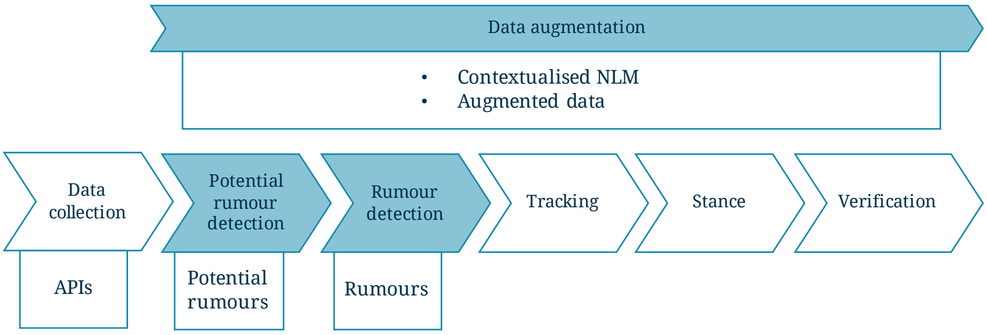


Figure 7.1: An extension of the rumour resolution pipeline proposed by Zubiaga et al. (2018a). The three sub-tasks in blue boxes were researched in this thesis.

Finally, context-aware, message-level ERD was researched. Research on ERD in this thesis opened up new opportunities for future rumour detection models in terms of the design of architecture, model training and evaluation. The proposed NN architecture is novel in that it incorporates effective tweet content embeddings obtained using a task-specific, fine-tuned NLM and it learns different types of social-temporal contexts highly correlated with source contents. While most deep learning architectures for rumour detection left the impact of different input features on model performance inexplicable, this thesis attempted to interpret what is going on in a black-box by incorporating attention mechanisms and thoroughly analysing results. Last but not least, the proposed model advanced SOTA performance in message-level ERD by 7-18% in different evaluation settings.

To sum up, this thesis has paved the way for generalisable, end-to-end frameworks not only for message-level rumour detection but for the entire rumour resolution process.

Section 1.2 set several research questions which have been investigated in respect of ERD. The rest of this chapter details how effectively the research of this thesis answered these research questions and summarises the contributions of this thesis. To recap, the contributions of this thesis have concerned:

Chapter 4. Labelled rumour data augmentation based on semantic relatedness

- Developing a data augmentation framework which leverages large-scale, real-world social media data, unlike current artificial data augmentation methods based on modifications to existing data or reliance on limited knowledge bases.

- Incorporating a SOTA context-sensitive NLM and fine-tuning it on a large-scale social media corpus with associated manual credibility annotations.

- Evaluating different models for word representation for paraphrase identification. The results show that the SOTA context-sensitive NLM fined-tuned with the large-scale Twitter corpus outperforms other SOTA word embedding models in terms of identifying paraphrase tweets.

- Evaluating the effect of augmented data on ERD via deep learning. The results of experiments on real-world event data sets show that data augmentation improves the performance of a SOTA DNN model for message-level rumour detection.

Chapter 5. Potential rumour identification via temporal signals

- Studying reasons why burst detection approaches based on temporal signals are more suitable for the task of early identification of potential rumours than bursty topic detection methods.

- *Key burst detection* is efficient, easy to reproduce, and portable to different domains in that it exclusively relies on *temporal signals*. Proposed temporal signals can better characterise bursty patterns of event evolution on social media than those proposed by SOTA methods do.

- Studying whether *extractive summarisation* based on graph- and term frequency-based keyword extraction algorithms can effectively select potential rumours from noisy Twitter corpora.

- Evaluating burst detection and summarisation methods in the context of ERD. The results show that extracted summaries of bursts are qualified as a collection of potential rumours although they are still noisy without further processing such as manual examinations and the application of automated rumour detection. Newsworthy stories and rumours during breaking news events can be identified using summaries of key bursts.

Chapter 6. Context-aware ERD

- Proposing a generalisable NN framework for ERD that learns the unified representation of rumours by combining sentence embeddings obtained using SOTA NLMs and social-temporal propagation features. Social and temporal context features can be hand-picked from raw metadata available from public Twitter's API. In other words, they do not require any domain-specific feature engineering.

- Exploring whether domain-specific SOTA NLMs trained on a large-scale social media corpus can provide the better syntactic and semantic representation of rumours and short social media texts than pre-trained NLMs and SOTA word emebdding models.

- Demonstrating that fine-tuning a SOTA word embedding model using domain-specific social media corpora has the potential for improvements in ERD.

- Learning representations of rumours and their spreading patterns by employing large training sets generated via weak supervision.

- Evaluating the effectiveness of the proposed model in a setting close to real-world scenarios.

7.1.1 *Labelled Rumour Data Augmentation*

Deep learning techniques are being actively researched and leveraged in a wide range of ML tasks. The main reason for this is that they achieve new SOTA performance with little or no feature engineering which is labour-intensive and time-consuming. When it comes to rumour detection, in particular, a recent study (Kwon et al., 2017) on rumour detection raised a concern regarding the predictive power of widely used hand-crafted features. Remember that the ultimate goal of this thesis is to develop an end-to-end, message-level rumour detection architecture which identifies rumours in the early stages of event evolution based on their *contents* and *context* and is *generalisable* to new events. To this end, this thesis decided to leverage the power of deep learning and raised the following research question: "What are the main reasons for the poor generalisation and transferability of deep learning-based rumour detection architectures to new data?"

As discussed in Section 1.1 and 2.7, *the scarcity of labelled training examples* limits the generalisability of existing rumour detection models to new data and settings. A fundamental reason why labelled rumour data is scarce is that the identification of a rumour requires profound domain knowledge and rigorous inspection unlike common annotation tasks such as named-entity annotation. For example, this thesis's benchmark data set, *PHEME (6392078;* (Kochkina et al., 2018a; Zubiaga et al., 2016a)), was manually annotated by journalists (i.e. domain experts) under a sophisticated scheme (Zubiaga et al., 2014). As tweets are short and contain very limited contents and contexts on their own, domain experts were given a conversational thread of each tweet to be annotated. The difficulty of manual annotation has a side effect called *class imbalance*. Xu and Chen (2015) state that most popular approaches for addressing class imbalance such as oversampling manually labelled data and generating artificial data samples may lead to overfitting and poor generalisability.

Limited labelled data can have a great impact on the training of DNNs as they learn training data better with more data samples. To address labelled data scarcity and class imbalance which hinder achieving the full potential of DNNs, Chapter 4 researched methods for automatically augmenting publicly available rumour data sets.

Section 3.1.2 discussed how rumours spread on social media and characteristics of variants of rumour sources in the early stages of event evolution based on the findings of existing work. In addition, Section 2.3.1 introduced related work on artificial data augmentation for textual data. Most work generated variations of manually labelled texts by transforming words and phrases such as replacing adjectives and/or entities with their synonyms. This approach can increase the size and diversity of training data, but cannot augment *contexts* (i.e. replies). Remember that this thesis aims at message-level rumour detection, the input of which is source tweets for rumours and non-rumours and their conversational threads (i.e. replies). Specifically, each training instance in input consists of a source tweet and its contexts (i.e. replies). A data augmentation method for such a setting was required to enrich not only source tweets but also corresponding context. *Source tweets* refer to tweets that initiate a new Twitter conversation (i.e. not replying to existing tweets) (Hoi, 2015). Replies cannot be collected for synthetic source tweets which are textual variations of real tweets.

Given the insights gained from relevant research, this thesis decided to enrich publicly available, limited labelled rumour source tweets (i.e. *references*) with large-scale social media corpora associated with real-world breaking news events (i.e. *candidates*). Duplicates and textual variants of *references* were selected from *candidates* based on semantic similarity between every pair of a reference and candidate in input. Subsequently, conversational threads for augmented source tweets were collected to build the final data sets. For the representation of short texts, this thesis fined-tuned a SOTA context-sensitive NLM called *ELMo* (Peters et al., 2018) on a large-scale social media corpus with associated manual credibility annotations. The performance of the fined-tuned model was evaluated on the task of paraphrase identification. The results illustrate that it can provide more meaningful and effective representations of short social media texts than other SOTA models for word representations. To the best of my knowledge, no work has fine-tuned the ELMo on a large-scale domain-specific Twitter corpus. The fine-tuned model can be utilised in several applications related to the credibility and veracity of information.

Section 4.6.2 showed that the proposed method increased not only the size but also the diversity of manually labelled data without human supervision. It also addressed the class imbalance problem. Returning to the first research question, this thesis studied whether data augmentation can improve the generalisability of ERD models based on DNN. It evaluated the impact of data augmentation on message-level rumour detection by using a SOTA DNNs rumour detection model (Kochkina et al., 2018a). This thesis modified the original implementation for stricter evaluation suitable for real-world scenarios in which a rumour detection model performs classification on unseen and new data.

"What are the main reasons for the poor generalisation and transferability of deep learning-based rumour detection architectures to new data?"

The research described in Chapter 4 fulfilled the aims stated in Section 3.1. The experiments clearly illustrated that data augmentation improves the performance and generalisability of a SOTA ERD model exploiting contextual information by alleviating labelled data scarcity and addressing class imbalance.

7.1.2 *Identification of Potential Rumours*

Going back to the problem statement introduced in Section 1.1, there exists another challenge that should preliminarily be addressed to make message-level ERD applicable to real-world situations: *data reduction*. In the context of rumour detection, data reduction can be referred to as the annotation of source tweets as *potential rumours* (Hoi, 2015). Specifically, this task is to uncover emerging stories or events which are likely to be associated with rumours. When dealing with large-scale and noisy social media data generated at a fast speed, one of the first important things that need to be addressed is: "How should individual posts be processed for further data analysis?" It is well known that it is impracticable to analyse every single post generated during breaking news events, especially in real time. In particular, it is not feasible and pointless for a message-level rumour detection model to classify every single post related to an event without any filtering. Despite the importance of data reduction as a preliminary step for rumour detection, most work on rumour detection skips it by assuming that source posts and/or their context are input by humans or are selected based on their popularity represented by the number of reposts (Zubiaga et al., 2016b).

A thorough search of the relevant literature opened the following research questions: "how potential rumours be identified with minimal manual supervision and time delay?", "What are signals which can identify potential rumours in the early stages of event evolution?", and "How can the signals be leveraged to identify potential rumours?"

These research aims were introduced in Section 3.2 and investigated in Chapter 5. The main goal of the chapters was to partially fill the research gap by studying methods for automatically and effectively reducing the number of input posts. This can be done by eliminating less significant examples in the initial stages of data analysis. To this end, the task of the identification of potential rumours was first decomposed into two tasks, *key burst detection* and *extractive summarisation*, which are recursively performed over a temporally segmented Twitter corpus (i.e. a time series). The objective was to select high-quality potential rumours which can be used as input into an ERD model.

This thesis provided background information on why it researched key burst detection and extractive summarisation for identifying potential rumours. Section 2.5 investigated several hand-crafted features which are simply based on metadata obtained from Twitter's API. Most of the investigated features can be extracted from other social media platforms as well. The goal was to identify signals which can distinguish rumours from non-rumours with event- and task-agnostic algorithms. The results show that *temporal patterns* are the most promising signal. Moreover, Section 2.5.4 thoroughly discussed the advantages of using temporal signals, particularly bursts, based on the findings of related work. For example, these sections explained why a combination of *key burst detection* and *summarisation* is more suitable for potential rumour identification than *bursty topic detection*. As the latter has been well studied for several applications, it could be argued that it can fulfil the aims stated in Section 3.2 more efficiently and effectively without decomposing the proposed task into two sub-tasks. To recap, given a set of social media posts, bursty topic detection models usually produce key topic words representing the entire corpus. However, potential rumours for message-level rumour detection should be real individual social media posts, from which contextual information can be extracted. Remember that a similar issue was raised in Chapter 4; data augmentation methods for rumour detection should enrich existing data with real social media posts. Bursty topics may be useful in event-level rumour detection because topic words can be used to collect posts related to an event of interest. Accordingly, this thesis employed *extractive summarisation* rather than *generative summarisation*.

As for key burst detection, input is a time series generated using posting times of input tweets related to an event. It is ideal to perform rumour detection in real time. However, due to the difficulty of data connection, most studies on real-time rumour detection did not conduct experiments on streaming data. They conducted experiments on historical data assuming that models receive data in real time. To do so, their methods detect rumours based on information observed up to each time step. This thesis adopted the same setting to potential rumour identification. At each time step, the proposed method extracts temporal features representing temporal dynamics observed up to the time window and determines whether the window is a key burst or not. Unlike existing solutions for burst detection that omitted to explain how model parameters affect the results, this thesis performed a thorough analysis of model parameters so that the proposed method can easily be reproducible and generalise to new data.

For the evaluation of key burst detection, this thesis compared the proposed method with SOTA methods based on patterns of detected bursts and the behaviour of model parameters. Although the results showed the generalisability and effectiveness of the proposed method, it was not entirely clear whether and how the proposed method was more advantageous for potential rumour identification than the baselines. To clarify its contribution to the proposed task, this thesis proposed a novel approach for evaluating burst detection methods specifically in the context of potential rumour identification. Section 5.4.5.1 described its details. The experimental results presented in Section 5.5.3 show that the proposed method is more effective than baselines in discovering potential rumours. In other words, bursts identified by it contain more potential rumours and those detected by the baselines.

As for summarisation, this thesis proposed three extractive summarisation methods based on three graph-based keyword extraction algorithms. Section 5.3.3 described the details of the methods. Ideally, as soon as a key burst has been identified, summarisation is performed. Therefore, once a key burst is identified, a summarisation method receives a collection of tweets which were posted during each burst as input. In the experiments, the proposed methods were compared with several term frequency- and graph-based baseline methods. As documented in Section 5.4.5.2, this thesis evaluated methods according to the applicability of output summaries to potential rumour identification. The results show that there is no one-for-all solution; the number and types of identified potential rumours vary according to events and summarisation methods. Overall, frequency-based baselines identify more potential rumours than the graph-based methods.

To the best of my knowledge, this is the first work researching a gap between data collection and analysis in the setting of ERD. Incorporating potential rumour identification into a deep learning-based rumour detection architecture enables the development of entirely automated rumour identification systems. The experiments on the identification of potential rumours fulfil all aims described in Section 3.2; however, the reasons for the varying results of burst summarisation are not completely clear. In specific, future research can investigate why certain methods work well with specific events but not with others.

7.1.3 *Context-Aware Early Rumour Detection*

A typical rumour resolution process consists of four sub-tasks (Zubiaga et al., 2018a): rumour detection, tracking, stance classification, and verification. This thesis highlighted the importance of the development of automated ERD for the success of the entire rumour resolution success. To this end, it first identified several challenges for the task and researched labelled rumour data augmentation in Chapter 4 and potential rumour identification in Chapter 5.

Section 2.1.2 introduced two main approaches for rumour detection and their differences: event-level and message-level classification. The aim of the former is to determine whether an event is a rumour or not given a set of messages related to it, while the latter aims to identify individual rumour-bearing messages. The latter is more appropriate for ERD in that it can be performed before events become viral, i.e. before many people engage in event diffusion by expressing their thoughts and sharing related information. However, the fact that individual social media posts contain limited information and a large amount of noise due to a character limit makes it difficult to identify whether a single message is related to a rumour or not. Based on this insight, Chapter 6 researched the task of message-level ERD and has addressed its four main challenges which will be recapped in the following parts.

Firstly, this thesis exploited contextual information to handle the problem of limited context in analysing individual source tweets. In the field of message-level rumour detection, "context" usually refers to conversational threads of source tweets such as replies in the case of Twitter. Several studies have explored linguistic characteristics of users' reactions to source tweets and profiles of users who have engaged in a conversation. They reported that these are helpful for distinguishing rumours from non-rumours. Above all, it is natural for humans to look for coherence between an unknown problem and its surroundings (Zubiaga et al., 2016a). Deep learning techniques mimic how human brains function, and therefore, providing conversational context to deep learning-based rumour detection models is a promising approach to address the first challenge introduced in this thesis. Based on the previous findings and inference, Chapter 6 proposed a context-aware hybrid DNNs, called **RP-DNN**. It takes a sequence of source tweets and corresponding contexts as input and outputs a binary label (i.e. rumour or non-rumour) for each source tweet. It jointly learns the text content of source tweets; that of replies; and tweet-level and user-based hand-crafted features. It could be argued that exploiting hand-crafted features limits the main advantage of DNNs, that is, they can automatically and effectively learn representations of input with little or no feature engineering. The employed features were shallow and could be extracted from metadata provided by Twitter without any painstakingly complicated and domain-specific feature engineering. In addition, the text content and metadata of tweets are highly correlated (Kıcıman, 2010). Therefore, their impact on the efficiency of the proposed architecture was negligible, but they still could provide the RP-DNN with auxiliary information.

To investigate the effects of different components of the RP-DNN, an ablation study with 8 different configurations was performed (see Section 6.3.7). The results presented in Section 6.4 show that source tweet contents, context content, and context metadata play complementary roles in rumour detection. The impact of context metadata is marginal, while context contents produce a relatively significant effect.

Different reactions in a conversational thread of a source tweet show differences in importance for identifying whether the source is a rumour or not. Paying more attention to more significant replies can filter out noisy and unnecessary information, thereby obtaining more accurate and meaningful representations of contexts. To incorporate this into the RP-DNN, attention mechanisms were introduced. They enable DNNs to selectively focus on the most important and useful segments of the sequence, and to effectively learn long-range dependencies (Vaswani et al., 2017). The results of the ablation study show that the proposed attention mechanisms improve rumour detection performance.

Secondly, this thesis researched how to effectively represent noisy social media posts particularly for rumour detection. The most fundamental input to rumour detection is the textual content of source tweets. Rather than manually curating linguistic signals for rumours, this thesis exploited a SOTA contextualised NLM called ELMo (Peters et al., 2018). Most research on deep learning-based rumour detection relies on standard word embedding techniques (e.g. word2vec). This thesis employed ELMo which had been pre-trained on a large corpus and fined-tuned with a domain-specific corpus for rumour detection on social media. It has several advantages over conventional embedding models. They build a fixed set of unique words (i.e. vocabulary) that may lead to an out-of-vocabulary problem (i.e. words appearing in input are not in a vocabulary). In contrast, the ELMo is purely character-based and does not suffer from the problem. As shown in Chapter 4, the fine-tuned ELMo is particularly advantageous as it provides the better syntactic and semantic representation of short, noisy, and rumour-mongering tweets.

Thirdly, as already discussed in Chapter 4, limited labelled training data is a known limitation in the field of rumour studies. It particular, it can lead to overfitting in deep learning models. To address this issue, this thesis aggregated three publicly available rumour data sets including large-scale data augmented with weak supervision and generated large training data for rumour detection. The experiment results show that the RP-DNN can be trained effectively with the data and achieve performance comparable to SOTA baselines on an unseen data samples. However, it is worth further exploring ways to have a larger training set with a minimum of human supervision. This will lead to a more generalisable model.

Finally, little work has conducted strict evaluation to assess the effectiveness of a rumour detection model in a setting close to real-world scenarios. A large body of work on rumour detection evaluated models via cross-validation in which a data set is split into training and test subsets for model training and evaluation. However, it is not an optimal way to evaluate whether a model can generalise to unseen data because training and test sets have similar distributions in CV. This thesis evaluated the proposed models via LOOCV in addition to standard k-fold CV. The former is stricter, but more suitable for assessing generalisability.

The experiments in Chapter 6 fulfil all aims stated in Section 3.3. However, the impact of contextual information was relatively weak compared to that of source tweets' textual contents. It would be interesting to address several limitations discussed in Section 6.5 in future work.

7.2 FUTURE WORK AND OUTLOOK

7.2.1 *Comprehensive Evaluation of Data Augmentation*

For the selection of an embedding model for data augmentation task, word2vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014) and ELMo (Peters et al., 2018) were evaluated on paraphrase identification as described in Sections 4.5.1 and 4.5.1.1. Future research will incorporate more advanced SOTA NLMs such as OpenAI GPT (Radford et al., 2019; Radford et al., 2018) and Google BERT (Devlin et al., 2018) into evaluation.

In Chapter 4, the evaluation of the proposed data augmentation framework was threefold. Firstly, the chapter showed that data augmentation increased the size of publicly available rumour data sets. Secondly, a few source tweets randomly sampled from the augmented data were manually examined to investigate whether data augmentation contributes to diversity. Finally, the chapter examined whether augmented data could improve the performance of a SOTA DNN architecture for ERD (Kochkina et al., 2018a).

This thesis suggests that future research examines more comprehensive evaluation of data augmentation. The manual inspection of randomly sampled examples is very limited. Future research will need to thoroughly examine the diversity of augmented data.

According to Appendix a.2 which shows rumour references and the number of rumour source tweets relevant to each individual reference in the augmented data, it is likely that there exists a bias toward certain rumours. This finding opens the following research questions: "Does imbalance in augmented training data for rumour detection make a rumour detection model produce biased classification results?" and "If so, how can bias be reduced?"

The evaluation of data augmentation in the setting of ERD was done using a benchmark model (Kochkina et al., 2018a). The effectiveness of the augmented data was evaluated based on F1-score, precision, and recall. However, the reason for performance gains with the augmented data is not entirely clear. Future research still needs to examine whether performance improvement is due to increased data size and/or balanced class distributions. Moreover, no experiment was conducted to examine whether augmented data can facilitate deeper NNs for ERD. Future work plans to design and conduct experiments for this issue.

7.2.2 *Context-Aware Potential Rumour Identification*

The annotation of rumours is a challenging task because it requires domain knowledge and human supervision. For example, this thesis's benchmark data set, *PHEME (6392078;* Kochkina et al. (2018a)), was manually annotated by domain experts under a sophisticated scheme (Zubiaga et al., 2014). As tweets are short and contain very limited contents and contexts on their own, domain experts were given a conversational thread of each tweet. In detail, domain experts considered the *polarity* (e.g. positive and negative), *modality* (e.g. certain and probable), *presentation* (e.g. claim and comment), *evidentiality* (e.g. witnessed and quoting source), and *plausibility* (e.g. plausible and dubious) of source tweets. They also considered the *acceptability* (e.g. agreement and disagreement on the corresponding source tweet) and *veracity* (e.g. true and false) of reactions (i.e. replies).

More evidence has suggested that exploiting contextual information is influential in performing rumour-related tasks. Section 2.1.3 introduced the finding of several related studies which investigated the behaviour of recipients of a source tweet to characterise rumours and their diffusion on social media. Chapter 4 also considered the importance of context by removing source tweets without replies from the augmented data.

Despite the usefulness of context, this thesis has not considered it for potential rumour identification. In particular, the summarisation component of the proposed framework extracted potential rumours solely based on constituent words of posts. It is highly likely that summaries of key bursts are appealing enough to make the public eager to spread, verify, and/or debunk them. However, it could be researched how contextual information could be combined with the findings of this thesis. One possible direction for future work is utilising the previous findings introduced in Section 2.1.3 to develop a context-aware potential rumour identification system. Most work characterised rumours based on *stances* of replies towards their sources. Specifically, future work could research how stances can be incorporated into weakly supervised potential rumour identification without adding great complexity to the entire rumour detection architecture.

7.2.3 *Generalisability of Rumour Detection Architectures*

Ensuring the generalisability of rumour detection models is crucial because most rumours emerging during breaking news events in the real world are new (i.e. unseen in previous events). Chapter 6 highlighted this by generating an extensive training data set by combining three publicly available rumour data sets. It also evaluated the proposed models via LOOCV following very few studies (Kochkina et al., 2018b; Zubiaga et al., 2017) which acknowledged the importance of a strict evaluation setting. However, it is still challenging to show the generalisability of DNNs for rumour detection due to limited training data. It is known that they learn training data better with more data samples. Future work will research more approaches for generating training data for rumour detection with a minimum of human supervision.

Another interesting research direction is to investigate whether jointly learning rumours from completely different domains can improve generalisability. Although this thesis studied rumours regarding several real-world events, most of them are limited to breaking news events, in particular, terrorist attacks and hostage-taking. Only a few source tweets in the data used in the experiments cover different domains such as business and entertainment. It would be interesting to incorporate a large number of rumours from a wide range of domains such as sports and politics and investigate the effects of joint learning on generalisability. Increasing training data size and diversity can allow exploring deeper NNs, which are expected to improve generalisability (Hernández-García and König, 2018).

7.2.4 *Exploration of Additional Representations*

The RP-DNN proposed in Chapter 6 exploits the textual content of source tweets, that of replies, and social contexts of replies represented by Twitter metadata. This thesis employed 27 hand-crafted features. The major advantage of using them is that labour-intensive and time-consuming feature engineering is not required. However, the experimental results in Section 6.4 showed that the impact of such features is negligible. Future research needs to further investigate this issue and explore what contextual information or signals for rumours can bring substantial improvements.

Previous studies have employed follower-following relationships (Kwon et al., 2017; Kwon et al., 2013; Zamani et al., 2017; Vosoughi, 2015) or propagation structure (Ma et al., 2017; Ma et al., 2018b; Kochkina et al., 2018b) for rumour detection. The major drawback of these representations is that they require feature engineering or the stance classification of replies. A thorough search for contextual information for ERD is required in future work. At the same time, it should delve into methods for addressing a trade-off between the efficiency of feature extraction and model performance. A more fundamental problem is, however, there is no way to obtain such features for the publicly available Twitter data sets for rumour studies because Twitter's API does not provide them for historical data. A possible future work direction could be to generate new data sets about events which prompt several rumours.

7.2.5 *An End-To-End Message-Level* ERD

An end-to-end message-level rumour detection framework in this thesis refers to a system which (1) automatically collects data, (2) selects potential rumours from massive amounts of noisy social media data, and (3) identifies rumours before they become viral. Firstly, data for message-level rumour detection can be collected using general keywords describing target events (Zubiaga et al., 2016a). Twitter's Search API[1] enables users to collect tweets containing certain words. For instance, for the Charlie Hebdo shooting which occurred in 2015, the following keywords can be used to collect data: "#charliehebdo", "#jesuischarlie", "charlie hebdo", and "paris". For the second component, in Chapter 5, the automatic identification of potential rumours with a minimum of human supervision was studied to facilitate the application of ERD models in real-world scenarios (e.g. breaking news events) in which massive amounts of noisy social media data is generated. Finally, in Chapter 6, a novel context-aware rumour detection model was researched. However, no experiment connecting the outcomes of these two chapters was conducted because identified potential rumours only have weak labels (i.e. annotations obtained via weak supervision). Test data instances should be high-quality and precise in order to properly evaluate the effectiveness of a rumour detection model. They is also why the PHEME (6392078; Kochkina et al. (2018b)) was used to generate test sets rather than the PHEME augmented with weak supervision in LOOCV experiments in Chapter 6. Therefore, engaging domain experts such as journalists to manually annotate identified potential rumours

[1] https://developer.twitter.com/en/docs/tweets/search/api-reference/get-search-tweets

can be considered in future work. This will also help to identify limitations of the current methods for the identification of potential rumours and further improvements to their effectiveness.

Part IV

APPENDIX

a

APPENDIX FOR CHAPTER 4

A.1 REFERENCES USED FOR DATA AUGMENTATION.

Table a.1 shows 20 example references for six events used for data augmentation (see Section 4.2.4 for details).

Boston marathon bombings

1	Two explosions near finish line	11	Bombs were pressure cookers and placed in black duffel bags
2	Boston bombings are a false flag operation	12	Cell phone service has been shut down
3	8 years old child died	13	Third bomb has gone off at the Boston Library
4	8 years old girl died while running for her best friend	14	Suspects Became Citizens on 9/11
5	78 years old runner is knocked down by blast	15	Suspect in Boston bombing described as dark skinned male
6	78 years old runner knocked down is Bill Iffrig	16	A fire broke out at the JFK library, not an explosion
7	8 years old girl is not one of the dead, a 8 years old boy is dead	17	FBI now acknowledges they interviewed Tsarnaev 2 years ago at the request of a foreign country
8	Mom and sister of the 8 years old boy who was killed are injured	18	One suspect in Boston Marathon bombing shot and killed by police. The other suspect on the run
9	The third explosion at the JFK library unknown connection	19	Friend of Boston Bombing Suspect Shot and Killed by FBI Agent
10	A Saudi Arabian descent is in custody	20	8 years old boy who was killed was waiting for his dad to finish his marathon

Charlie Hebdo shooting

1	An armed man has taken a hostage in a kosher store in Paris (Porte de Vincennes)	11	The two Charlie Hebdo suspects have been killed in a police assault in Dammartin
2	Four cartoonists were killed in the Charlie Hebdo attack	12	Multiple gunmen were involved in the Charlie Hebdo attack
3	The gunmen at Charlie Hebdo yelled, "The Prophet is avenged"	13	The gunmen were asking for the journalists by name
4	Three gunmen are on the run after the Charlie Hebdo attack	14	5 people have been injured in the Charlie Hebdo attack
5	The gunmen said, "You tell the media it was al-Qaeda in Yemen"	15	Kosher supermarket gunman is asking for release of Charlie Hebdo suspects
6	Ahmed Merabet was the first victim of the Charlie Hebdo attack	16	The gunman at the kosher store is linked to the killing of the police officer in Montrouge
7	ID card of one of the Charlie Hebdo suspects found in car abandoned by gunmen	17	Kosher restaurants/Jewish shops (and schools, synagogues, etc.) are closing in Paris in wake of Porte de Vincennes hostage-taking
8	A police operation (convoy, helicopters) is underway to catch the Charlie Hebdo suspects northeast of Paris	18	The Kosher market suspect has links to the Charlie Hebdo suspects.
9	Suspect Hamyd Mourad was in class at the time of the Charlie Hebdo shooting	19	The gunmen were armed with Kalashnikovs and rocket launchers
10	The shooting of two policemen in Montrouge is not connected to the Charlie Hebdo attack	20	ISIS has claimed responsibility for Charlie Hebdo attack and announced more attacks to come in the West

Ferguson unrest			
1	M. Brown was involved in a robbery before being shot	11	Two of the four police departments in Ferguson were trained by Israel
2	Initial contact between police officer and M. Brown was not related to the robbery	12	Pentagon supplied St. Louis county police with military-grade weapons
3	Ferguson police once beat up a man and charged him for bleeding on their uniforms	13	Officer who shot M. Brown did not write an incident report
4	Anonymous has audio files of police and EMS calls of Brown shooting	14	Mike Brown was shot 10 times
5	A woman was shot in a drive-by shooting	15	Media is barred from entering Ferguson
6	Only one TV reporter was present at August 10 night-time Ferguson protests	16	Nearly 7,000 blacks were killed in 2013, most of them by blacks
7	People at Ferguson protests are shouting “Kill the police”	17	KKK raising money for officer who shot M. Brown
8	Ferguson police to release name of police officer who shot M. Brown today (August 15)	18	Every 28 hours a black male is killed in the US
9	Ferguson police released store surveillance video because of media requests	19	Americans are eight times more likely to be killed by police than by a terrorist
10	Police officer who shot M. Brown has already left town	20	Fox News is not covering the Ferguson protests

Germanwings plane crash			
1	A Germanwings airplane has crashed in the French Alps near Digne (southern France)	11	The Germanwings flight disappeared from the radar at 9.39 UTC
2	67 German nationals were onboard the flight	12	The co-pilot interrupted his training because of burnout or depression / had a serious depressive episode in 2009
3	There were 148 people onboard the Germanwings plane	13	A distress/mayday/SOS call was sent from the airplane before the crash
4	The pilots were Patrick Sondheimer (captain) and Andreas Lubitz (co-pilot)	14	(Up to) 150 people perished in the crash (144 passengers, 6 crew)
5	The co-pilot was a convert to Islam	15	The pilots called "emergency, emergency" from the cockpit (last words before the crash)
6	The Germanwings plane that reportedly crashed was flying from Barcelona to Dusseldorf (flight 4U9525)	16	The Germanwings plane experienced a rapid descent before crashing
7	Only one pilot was in the cockpit at the time of the crash/the other pilot was (deliberately) locked out	17	There were 142 passengers on board the Germanwings flight
8	The airplane lost signal at 6,800 feet.	18	There are no survivors in Germanwings crash
9	The co-pilot received a commendation from the FAA for his flying skills	19	Two babies were among the passengers on the Germanwings flight
10	A German school group (16 students, 2 teachers) from Haltern among the passengers on Germanwings	20	

Ottawa shooting			
1	Shots fired on Parliament Hill	11	There were three separate shooting incidents
2	Soldier shot dead is Cpl. Nathan Cirillo	12	Three patients injured in Ottawa shooting released from hospital
3	The Leafs-Senators game in Ottawa has been postponed	13	There was a shooting incident near/at the Rideau Centre
4	A second shooting suspect has been shot	14	The suspect’s name is Michael Zehaf-Bibeau
5	A soldier has been shot at National War Memorial	15	30 shots were fired inside/on Parliament Hill
6	The suspect had his passport confiscated by Canadian government	16	The shooting was a coordinated attack

7	There are multiple shooting suspects still at large	17	The FBI is assisting Canadians with the Ottawa shooting
8	The suspect is on the roof of building at Metcalfe and Sparks Street	18	The soldier who was killed had a six-year-old son
9	Canadian officials are calling the incident a terrorist attack	19	Authorities were aware of potential ISIS-related attack
10	Suspect is (also) named Michael Joseph Hall	20	Suspect was a (Canadian) convert to Islam

	Sydney siege		
1	The gunman and/or hostages have made contact with Sydney media outlet(s) (radio station, etc.)	11	There is a hostage situation at a cafe in Sydney
2	A (black) Islamic flag is being held up in the window of cafe in Sydney's Martin Place	12	There is more than one gunman involved in the hostage-taking at the Sydney cafe
3	Hostages are being held by men waving an ISIS flag inside a cafe in Sydney's Martin Place	13	The Sydney airspace has been closed
4	The gunman has said he has "devices" (bombs) placed in other parts of the city	14	The gunman has made specific demands to authorities
5	Uber introduced surge pricing in downtown Sydney during hostage crisis	15	The hostage-taker has been killed/shot (by police)
6	A bomb detection robot has entered the cafe	16	Hostages are running out/escaping from the cafe
7	(Alleged) ISIS militants are behind the hostage-taking in the Sydney cafe	17	The gunman wants to speak to Prime Minister Tony Abbott
8	The Lindt cafe was targeted because it was not halal-certified	18	A police officer has a gunshot wound to the head/is injured
9	Media outlets know identity of gunman but are not publishing it	19	One hostage texted "I'm OK" to his mother
10	The hostage-takers are wearing suicide belts	20	Police have arrested a suspect at Martin Place

Table a.1: Examples of references used for data augmentation.

A.2 STATISTICS OF RELEVANT REFERENCES IN THE AUGMENTED DATA.

Boston marathon bombings

Reference	Count
8 years old girl died while running for the Sandy Hook kids	1,546
8 years old boy died while running for the Sandy Hook kids	1,473
8 years old boy who was killed was waiting for his dad to finish his marathon.	1,111
Officials found what they believe are five undetonated explosive devices in Boston area.	910
Boston police commissioner: 3rd explosion at JFK Library	659
Bombs were pressure cookers and placed in black duffel bags	620
Google has created a Person Finder for those affected by the explosions	436
Westboro Baptist church to picket funerals of those dead by Boston bombs.	434
2 explosive devices found	287
8 years old boy's mom got a brain injury and his sister lost a leg.	232
Westboro Baptist church picketing	152
Mom and sister of the 8 years old boy who was killed are injured.	118
A suspicious driver leads police, FBI to apartment in Revere, Mass.	110
2 more explosives found at Boston marathon	53
Two explosions near finish line	51
No one is in custody	44
8 years old girl is not one of the dead, a 8 years old boy is dead	34
A suspect caught on a CCTV surveillance camera	30
The third explosion at the JFK library unknown connection	28
Cell phone service has been shut down	23
A fire broke out at the JFK library. Not an explosion.	21
Boston bombings are a false flag operation	16
Third bomb has gone off at the Boston Library	15
8 years old boy died	14
Suspects Became Citizens on 9/11	13
8 years old child died	11
8 years old girl died while running for the newtown kids	11
Doctors: bombs contained pellets, shrapnel and nails that hit victims	11
Boston Marathon suspect in custody	10
8 years old boy killed has been identified as Martin Richard	10
There will be a controlled explosion opposite the library within one minute as part of bomb squad activities	9
2 more explosive devices found	6
Police tell people to stay away from the JFK library	5
The third device found	5
Danny Amendola will donate $100 catch &$200 drop next season	4
8 years old girl died while running for her best friend	4
The third explosion has been reported	3
8 years old boy died while running for the newtown kids	3
The suspect has the same rights as any Americans is on a student visa.	3
8 years old girl or boy?	2
April 15: Lincoln dies (1865), Titanic sinks (1912), Aer Lingus and McDonalds founded (1936/1955), Hillsborough (1989), Boston bombings (2013)	2

A man on the roof of a nearby building	1
Family Guy television series depicting the Boston Marathon bombings	1
Explosions were planned and coordinated by sinister force such as the government	1
Dzhokhar Tsarnaev has finally been arrested and is in custody.	1

Charlie hebdo shooting

Reference	**Count**
An armed man has taken a hostage in a kosher store in Paris (Porte de Vincennes)	5,165
At least one person has been killed (and several injured) in shootout with Charlie Hebdo suspects northeast of Paris	3,033
The Charlie Hebdo suspects say they want/are ready to die as martyrs	1,497
The gunmen at Charlie Hebdo yelled, "The Prophet is avenged"	1,388
Four cartoonists were killed in the Charlie Hebdo attack	1,339
The two Charlie Hebdo suspects have been killed in a police assault in Dammartin	1,027
There are at least five hostages in Porte de Vincennes kosher store	991
ID card of one of the Charlie Hebdo suspects found in car abandoned by gunmen	746
One person was injured in shootout in kosher store	652
Ahmed Merabet was the first victim of the Charlie Hebdo attack	552
Two Charlie Hebdo suspects have been spotted (in a town/gas station in Northern France)	507
At least two dead in hostage-taking in Porte de Vincennes	502
The gunmen said, "You tell the media it was al-Qaeda in Yemen"	480
The gunman at the kosher store is linked to the killing of the police officer in Montrouge	386
Kosher supermarket gunman is asking for release of Charlie Hebdo suspects	358
Several hostages have been freed in Porte de Vincennes	349
ISIS has claimed responsibility for Charlie Hebdo attack and announced more attacks to come in the West	340
Three suspects (inc. two brothers) have been identified in the Charlie Hebdo attacks	329
At least two reportedly injured in Porte de Vincennes hostage situation	328
The Charlie Hebdo suspects are holed up inside printing business in Dammartin	326
A baby is among the hostages in the Kosher market	314
A police operation (convoy, helicopters) is underway to catch the Charlie Hebdo suspects northeast of Paris	273
Stephane Charbonnier was critically injured in the Charlie Hebdo attack	252
Two people (seriously) wounded in shootout in hunt for Charlie Hebdo suspects northeast of Paris	234
The gunmen were armed with Kalashnikovs and rocket launchers	221
Kosher restaurants /Jewish shops (and schools, synagogues, etc.) are closing in Paris in wake of Porte de Vincennes hostage-taking	213
The hostage being held by the Charlie Hebdo suspects is a 26-year-old male	175
The suspect in the hostage-taking at the Kosher market is dead	155
Hostages have been taken (at a business) in Dammartin-en-Goele	130
(At least) 10 people are dead at Charlie Hebdo offices	119

The shooting of two policemen in Montrouge is not connected to the Charlie Hebdo attack	107
11 people died during the Charlie Hebdo attack	107
Killing of Montrouge police officer is linked to Charlie Hebdo attack (or killer is linked to CH suspects)	101
Three gunmen are on the run after the Charlie Hebdo attack	88
The hostage in Dammartin has been freed following police assault	53
The Charlie Hebdo suspects came out firing from the Dammartin building	48
Suspect Hamyd Mourad was in class at the time of the Charlie Hebdo shooting	47
One Charlie Hebdo suspect (H. Mourad) has handed himself in to police	27
The Kosher market suspect has links to the Charlie Hebdo suspects	22
5 people have been injured in the Charlie Hebdo attack	19
12 people died in connection with the Charlie Hebdo attack	14
Police officer shot at Montrouge has died of her injuries	8
Nine journalists died in the Charlie Hebdo attack	6
The gunmen were asking for the journalists by name	4
Attackers yelled out "Allahu Akbar" on their way out	4
Shots were fired at Montrouge in Paris	4
The Charlie Hebdo gunmen had military training	3
A police officer has been shot/wounded in Montrouge	3
“Multiple” gunmen were involved in the Charlie Hebdo attack	3
The Charlie Hebdo gunmen fled in a stolen car	2
2 police officers died during the Charlie Hebdo attack	1
Shots have been fired (and a car chase taken place) in Dammartin-en-Goele	1
There was gunfire at the Porte de Vincennes	1

Ferguson unrest

Reference	**Count**
Ferguson police once beat up a man and charged him for bleeding on their uniforms	4,219
Ferguson police to release name of police officer who shot M. Brown today (August 15)	1,406
Every 28 hours a black male is killed in the US	1,209
KKK raising money for officer who shot M. Brown	1,150
Initial contact between police officer and M. Brown was not related to the robbery	308
Americans are eight times more likely to be killed by police than by a terrorist	253
Two of the four police departments in Ferguson were trained by Israel	174
Nearly 7,000 blacks were killed in 2013, most of them by blacks	124
Ferguson police swat team have "murder insurance" to protect from lawsuits	107
M. Brown was stopped by police for walking in the middle of the street	85
Mike Brown was shot 10 times	81
Ferguson police released store surveillance video because of media requests	40
Pentagon supplied St. Louis county police with military-grade weapons	36
Witnesses have cellphone video of the Mike Brown shooting	35
M. Brown assaulted the police officer who shot him	30
M. Brown told his mother, “The world will know who Michael Brown is”	27
Officer who shot M. Brown did not write an incident report	27
People at Ferguson protests are shouting “Kill the police”	24

Anonymous has audio files of police and EMS calls of Brown shooting	11
M. Brown was involved in a robbery before being shot	4
Ferguson police are lying about the circumstances leading up to M. Brown's death	3
M. Brown was shot while walking down the street	2
Police officer who shot M. Brown has already left town	2
Woman shot in drive-by shooting shot video of protests	2

Germanwings plane crash	
Reference	**Count**
The Germanwings flight disappeared from the radar at 9.39 UTC	1,524
The Germanwings plane that reportedly crashed was flying from Barcelona to Dusseldorf (flight 4U9525)	883
The co-pilot was a convert to Islam	841
Only one pilot was in the cockpit at the time of the crash / the other pilot was (deliberately) locked out	807
The airplane lost signal at 6,800 feet	707
Two babies were among the passengers on the Germanwings flight	348
A Germanwings airplane has crashed in the French Alps near Digne (southern France)	83
(Up to) 150 people perished in the crash (144 passengers, 6 crew)	80
There are no survivors in Germanwings crash	79
A German school group (16 students, 2 teachers) from Haltern among the passengers on Germanwings	29
There were 142 passengers onboard the Germanwings flight	21
The Germanwings plane experienced a rapid descent before crashing	9
There were 148 people onboard the Germanwings plane	8

Ottawa shooting	
Reference	**Count**
The soldier shot at War Memorial has died	4,360
NORAD on high-alert posture	3,506
A soldier has been shot at National War Memorial	3,310
Shots fired on Parliament Hill	2,608
Soldier shot dead is Cpl. Nathan Cirillo	1,521
Honorary citizenship ceremony for Malala has been cancelled	1,130
There will be a police news conference	875
Rideau Centre has been evacuated	639
As many as 50 shots have been fired on Parliament Hill	381
Three patients injured in Ottawa shooting released from hospital	318
Canadian officials are calling the incident a terrorist attack	309
Parliament Hill is on lockdown	265
The Prime Minister will make a statement later today	249
The Leafs-Senators game in Ottawa has been postponed	226
The University of Ottawa is on lockdown	208
At least 20 shots were fired inside Parliament buildings	204
The soldier who was shot is a reservist from Hamilton	194
30 shots were fired inside/on Parliament Hill	139
Canadians have given name of suspect to U.S. officials	136

There was a shooting incident near/at the Rideau Centre	125
Authorities were aware of potential ISIS-related attack	119
Suspect was a (Canadian) convert to Islam	90
Rideau Centre is on lockdown	73
The suspect's name is Michael Zehaf-Bibeau	72
There were "several" shooting incidents	33
Suspected shooter was carrying a rifle	31
The FBI is assisting Canadians with the Ottawa shooting	22
All bridges to/from Ottawa are closed	17
The suspect was prevented from travelling abroad to join ISIS	16
The shooting was a coordinated attack	14
Shooter is still on the loose	9
Suspect has been apprehended on Parliament Hill	5
Two new patients admitted to hosptial	5
A second shooting suspect has been shot	4
There were three separate shooting incidents	4
U.S. Army has increased security at Tomb of the Unknown at Arlington National Cemetary	4
The suspect had his passport confiscated by Canadian government	2
The soldier who was killed had a six-year-old son	2
Suspected shooter has been killed/is dead	1
Obama to speak with Harper today	1

Sydney siege

Reference	Count
A gunman has taken hostages at a cafe in Sydney's Martin Place	2,815
Hostages are being held by men waving an ISIS flag inside a cafe in Sydney's Martin Place	2,749
A sixth hostage has escaped/run out of the Sydney cafe	2,688
Up to 20 people are being held hostage inside the Sydney cafe	2,256
There are 13 hostages inside the Sydney cafe	1,493
There is a hostage situation at a cafe in Sydney	1,461
40-50 hostages are being held at cafe in Sydney (according to Lindt CEO)	1,235
Uber introduced surge pricing in downtown Sydney during hostage crisis	1,139
Up to five more hostages have escaped from the cafe (after initial 5)	759
A police officer has a gunshot wound to the head/is injured	598
An ISIS (IS) flag is being displayed (by hostages) at the cafe in Sydney's Martin Place	435
A (black) Islamic flag is being held up in the window of cafe in Sydney's Martin Place	418
"Several" more hostages have escaped from the cafe (after initial 5)	381
The Sydney Opera House has been evacuated	377
The script on the flag reads: "There is no God but Allah and Muhammad is the messenger of God"	354
The gunman behind the hostage-taking is named Man Monis	325
At least 2 people have died	300
Hostages are running out/escaping from the cafe	299
The US Consulate in Sydney has been evacuated	288
One hostage texted "I'm OK" to his mother	235

The flag being held up is a Shahadah flag with Islamic creed (script) commonly used by militants	200
The gunman wants to speak to Prime Minister Tony Abbott	192
The gunman’s headband reads, “We are your soldiers O Muhammad”	134
Paramedics are carrying out a number of hostages from the cafe	132
The Lindt cafe hostage-taking may be a diversion for something else (another bigger terrorist event)	131
A bomb detection robot has entered the cafe	84
Police have stormed the cafe	62
Police (authorities) have been in contact with the hostage-taker	53
Six more hostages have escaped from the cafe (after initial 5)	52
At least four people have been injured	49
Martin Place/CBD is in lockdown (and surrounding buildings have been evacuated)	48
Police have established the identity of the gunman	42
The gunman is using a female hostage as a shield inside the cafe	37
The Sydney airspace has been closed	33
The siege/hostage-taking crisis is over	13
At least 7 hostages have been released from cafe (after initial 5 ran out)	12
(Alleged) ISIS militants are behind the hostage-taking in the Sydney cafe	10
At least 12 more hostages have emerged from cafe (after initial 5)	9
The hostage situation in the Sydney cafe is an armed robbery	7
There is more than one gunman involved in the hostage-taking at the Sydney cafe	5
The hostage-taker is down	5
At least one person has been shot / injured	5
The gunman has released (three) hostages / 3 hostages are free	4
Two more (female) hostages have run out of the cafe (for total of five hostages free so far)	3
The hostage-taker has been killed/shot (by police)	3
The gunman has said he has "devices" (bombs) placed in other parts of the city	2
Hostages are posting messages on social media from inside cafe	2
There has been an incident at Sydney Opera House	1
One more hostage has run out of the cafe (for total of 6)	1

Table a.2: Examples of references used for data augmentation.

b

APPENDIX FOR CHAPTER 6

b.1 GLOBAL MEAN AND VARIANACE OF THE TRAINING DATA

Table b.1 presents the statistics of context metadata features computed from all training sets used in LOOCV. Mean and standard deviation are used to normalise feature representations of context metadata.

Table b.1: Statistics of all training sets used in LOOCV.

	Feature	global mean	global std.	global max	global min
user features	# of posts	53687.5965	127457.9930	4420429	1
	listed count	102.4828	1228.3984	199211	0
	followers	7174.8032	233721.4625	42822106	0
	followings	1679.2319	8920.4870	1602789	0
	follow ratio	163.1336	31374.2225	9918751.75	0
	follow ratio (v2)	0.4722	0.2282	1	0
	# of user favourites	18493.6203	49741.6807	2419251	0
	account age	1197.2359	686.5547	3611	1
	is verified	0.0269	0.1618	1	0
	engagement score	82.5727	534.0954	164474.5	0.0005
	following rate	3.7328	32.6771	6207.4	0
	favourites score	38.3545	407.0863	85537	0
	is GEO enabled	0.5282	0.4992	1	0
	description length	11.7235	8.7322	64	0
	profile name length	11.6878	5.5836	50	0
	is source user	0.0098	0.0987	1	0
tweet features	# of retweets	263.1094	1040.1257	77459	0
	# of favourites	0.3600	36.3189	22836	0
	has question	0.0579	0.2336	1	0
	is duplicated	0.0011	0.0324	1	0
	has image	0.8762	0.3294	1	0
	has URL	0.3083	0.4618	1	0
	# of URL	0.31187853	0.4712	5	0
	has native media	0.1679	0.3738	1	0
	context content length	15.8925	5.5011	47	0
	response time (mins.)	1664.1876	42754.3823	3152616.7667	0
	has user profile description	0.8529	0.3542	1	0

BIBLIOGRAPHY

Abulaish, Muhammad and Kumar Sah Amit (2019). “A Text Data Augmentation Approach for Improving the Performance of CNN.” In: *Proceedings of the MINDS Workshop,the 11th International Conference on Communication Systems and Networks (COMSNETS)*. Banglore, India, pp. 1–6.

Ackerman, Margareta and Sanjoy Dasgupta (2014). “Incremental clustering: The case for extra clusters.” In: *Advances in Neural Information Processing Systems*, pp. 307–315.

Aggarwal, Charu C. (2013). *Outlier Analysis*. Springer. ISBN: 978-1-4614-6395-5. DOI: 10.1007/978-1-4614-6396-2. URL: https://doi.org/10.1007/978-1-4614-6396-2.

Aker, Ahmet, Arkaitz Zubiaga, Kalina Bontcheva, Anna Kolliakou, Rob Procter, and Maria Liakata (2017). “Stance classification in out-of-domain rumours: A case study around mental health disorders.” In: *International Conference on Social Informatics*. Springer, pp. 53–64.

Allport, Gordon W. and Leo J. Postman (1965). *The psychology of rumor*. Russell & Russell. URL: https://books.google.co.uk/books?id=N6O4AAAAIAAJ.

Alsaedi, Nasser, Pete Burnap, and Omer Rana (2016a). “Automatic summarization of real world events using twitter.” In: *Tenth International AAAI Conference on Web and Social Media*.

Alsaedi, Nasser, Pete Burnap, and Omer F. Rana (2016b). “Temporal TF-IDF: A High Performance Approach for Event Summarization in Twitter.” In: *2016 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, pp. 515–521.

Andrews, Cynthia, Elodie Fichet, Yuwei Ding, Emma S. Spiro, and Kate Starbird (2016). “Keeping Up with the Tweet-dashians: The Impact of 'Official' Accounts on Online Rumoring.” In: *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*. CSCW ’16. San Francisco, California, USA: ACM, pp. 452–465. ISBN: 978-1-4503-3592-8. DOI: 10.1145/2818048.2819986. URL: http://doi.acm.org/10.1145/2818048.2819986.

Arif, Ahmer, John J. Robinson, Stephanie A. Stanek, Elodie S. Fichet, Paul Townsend, Zena Worku, and Kate Starbird (2017). “A Closer Look at the Self-Correcting Crowd: Examining Corrections in Online Rumors.” In: *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*. CSCW ’17. Portland, Oregon, USA: ACM, pp. 155–168. ISBN: 978-1-4503-4335-0. DOI: 10.1145/2998181.2998294. URL: http://doi.acm.org/10.1145/2998181.2998294.

Ba, Jimmy Lei, Jamie Ryan Kiros, and Geoffrey E Hinton (2016). “Layer normalization.” In: *arXiv preprint arXiv:1607.06450*.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2015). “Neural Machine Translation by Jointly Learning to Align and Translate.” In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. URL: http://arxiv.org/abs/1409.0473.

Bahuleyan, Hareesh and Olga Vechtomova (2017). "UWaterloo at SemEval-2017 Task 8: Detecting stance towards rumours with topic independent features." In: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pp. 461–464.

Barabási, Albert-László (2005). "The Origin of Bursts and Heavy Tails in Human Dynamics." In: *Nature* 435, p. 207. URL: http://www.citebase.org/abstract?id=oai:arXiv.org:cond-mat/0505371.

Barrios, Federico, Federico López, Luis Argerich, and Rosa Wachenchauzer (2016). "Variations of the Similarity Function of TextRank for Automated Summarization." In: *CoRR* abs/1602.03606. arXiv: 1602.03606. URL: http://arxiv.org/abs/1602.03606.

Batagelj, Vladimir and Matjaz Zaversnik (2003). "An O(m) algorithm for cores decomposition of networks." In: *arXiv preprint cs/0310049*.

Becker, Hila, Mor Naaman, and Luis Gravano (2011). "Beyond trending topics: Real-world event identification on twitter." In: *Fifth international AAAI conference on weblogs and social media*.

Blei, David M, Andrew Y Ng, and Michael I Jordan (2003). "Latent dirichlet allocation." In: *Journal of machine Learning research* 3.Jan, pp. 993–1022.

Boididou, Christina, Symeon Papadopoulos, Yiannis Kompatsiaris, Steve Schifferes, and Nic Newman (2014). "Challenges of computational verification in social multimedia." In: *Proceedings of the 23rd International Conference on World Wide Web*. ACM, pp. 743–748.

Boididou, Christina, Stuart E Middleton, Zhiwei Jin, Symeon Papadopoulos, Duc-Tien Dang-Nguyen, Giulia Boato, and Yiannis Kompatsiaris (2018). "Verifying information with multimedia content on twitter." In: *Multimedia Tools and Applications* 77.12, pp. 15545–15571.

Bordia, Prashant and Ralph L Rosnow (1998). "Rumor Rest Stops on the Information Highway Transmission Patterns in a Computer-Mediated Rumor Chain." In: *Human Communication Research* 25.2, pp. 163–179.

Bordia, Prashant, Nicholas DiFonzo, and Artemis Chang (1999). "Rumor as group problem solving: Development patterns in informal computer-mediated groups." In: *Small Group Research* 30.1, pp. 8–28.

Buckner, H Taylor (1965). "A Theory of Rumour Transmission." In: *Public Opinion Quarterly* 29.1, pp. 54–70.

Castillo, Carlos (2016). *Big Crisis Data: Social Media in Disasters and Time-Critical Situations*. 1st. New York, NY, USA: Cambridge University Press. ISBN: 1107135761, 9781107135765.

Castillo, Carlos, Marcelo Mendoza, and Barbara Poblete (2011). "Information credibility on twitter." In: *Proceedings of the 20th international conference on World wide web*. ACM, pp. 675–684.

— (2013). "Predicting information credibility in time-sensitive social media." In: *Internet Research* 23.5, pp. 560–588.

Chakrabarti, Deepayan and Kunal Punera (2011). "Event summarization using tweets." In: *Fifth International AAAI Conference on Weblogs and Social Media*.

Chaudhari, Sneha, Gungor Polatkan, Rohan Ramanath, and Varun Mithal (2019). "An attentive survey of attention models." In: *arXiv preprint arXiv:1904.02874*.

Chelba, Ciprian, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson (2013). "One billion word benchmark for measuring progress in statistical language modeling." In: *arXiv preprint arXiv:1312.3005*.

Chen, Tong, Xue Li, Hongzhi Yin, and Jun Zhang (2018). "Call attention to rumors: Deep attention based recurrent neural networks for early rumor detection." In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, pp. 40–52.

Cho, Kyunghyun, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio (2014). "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation." In: *ArXiv* abs/1406.1078.

Choi, Heeyoul, Kyunghyun Cho, and Yoshua Bengio (2018). "Fine-grained attention mechanism for neural machine translation." In: *Neurocomputing* 284, pp. 171–176.

Collins, Michael and Nigel Duffy (2002). "Convolution kernels for natural language." In: *Advances in neural information processing systems*, pp. 625–632.

Damoulas, Theodoros, ed. (2015). *Artificial Intelligence for Cities, Papers from the 2015 AAAI Workshop, Austin, Texas, USA, January 25, 2015*. Vol. WS-15-04. AAAI Workshops. AAAI Press. ISBN: 978-1-57735-715-5. URL: http://www.aaai.org/Library/Workshops/ws15-04.php.

Dayani, Raveena, Nikita Chhabra, Taruna Kadian, and Rishabh Kaushal (2015). "Rumor detection in twitter: An analysis in retrospect." In: *2015 IEEE International Conference on Advanced Networks and Telecommuncations Systems (ANTS)*, pp. 1–3.

Derczynski, Leon and Kalina Bontcheva (2014). "Pheme: Veracity in Digital Social Networks." In: *Posters, Demos, Late-breaking Results and Workshop Proceedings of the 22nd Conference on User Modeling, Adaptation, and Personalization co-located with the 22nd Conference on User Modeling, Adaptation, and Personalization (UMAP2014), Aalborg, Denmark, July 7-11, 2014*. URL: http://ceur-ws.org/Vol-1181/pros2014_paper_05.pdf.

Derczynski, Leon, Kalina Bontcheva, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Arkaitz Zubiaga (2017). "SemEval-2017 Task 8: RumourEval: Determining rumour veracity and support for rumours." In: *arXiv preprint arXiv:1704.05972*.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding." In: *arXiv preprint arXiv:1810.04805*.

DiFonzo, Nicholas and Prashant Bordia (2002). "Corporate rumor activity, belief and accuracy." In: *Public Relations Review* 28.1, pp. 1–19.

— (2007). *Rumour Psychology: Social and Organizational Approaches*. American Psychological Association. ISBN: 978-1-59147-426-5. URL: https://books.google.co.uk/books?id=hZe5AAAAIAAJ.

Doerr, Benjamin, Mahmoud Fouz, and Tobias Friedrich (2012). "Why rumors spread so quickly in social networks." In: *Communications of the ACM* 55.6, pp. 70–75.

Doman, Keisuke, Taishi Tomita, Ichiro Ide, Daisuke Deguchi, and Hiroshi Murase (2014). "Event Detection Based on Twitter Enthusiasm Degree for Generating a Sports Highlight Video." In: *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, pp. 949–952.

Duan, Yajuan, Zhumin Chen, Furu Wei, Ming Zhou, and Heung-Yeung Shum (2012). "Twitter topic summarization by ranking tweets using social influence and content quality." In: *Proceedings of COLING 2012*, pp. 763–780.

Duchi, John, Elad Hazan, and Yoram Singer (2011). "Adaptive subgradient methods for online learning and stochastic optimization." In: *Journal of Machine Learning Research* 12.Jul, pp. 2121–2159.

El Boukkouri, Hicham, Olivier Ferret, Thomas Lavergne, and Pierre Zweigenbaum (2019). "Embedding Strategies for Specialized Domains: Application to Clinical Entity Recognition." In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pp. 295–301.

Erkan, Günes and Dragomir R Radev (2004). "Lexrank: Graph-based lexical centrality as salience in text summarization." In: *Journal of artificial intelligence research* 22, pp. 457–479.

Friggeri, Adrien, Lada A. Adamic, Dean Eckles, and Justin Cheng (2014). "Rumour Cascades." In: *Proceedings of the Eighth International Conference on Weblogs and Social Media, ICWSM 2014, Ann Arbor, Michigan, USA, June 1-4, 2014.* URL: http://www.aaai.org/ocs/index.php/ICWSM/ICWSM14/paper/view/8122.

Gao, Jie, Sooji Han, Xingyi Song, and Fabio Ciravegna (2019). *RP-DNN: A Tweet level propagation context based deep neural networks for early rumor detection in social media.*

— (May 2020). "RP-DNN: A Tweet Level Propagation Context Based Deep Neural Networks for Early Rumor Detection in Social Media." In: *Proceedings of The 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, pp. 6094–6105. URL: https://www.aclweb.org/anthology/2020.lrec-1.748.

Geng, Yue, Zheng Lin, Peng Fu, and Weiping Wang (2019). "Rumor Detection on Social Media: A Multi-view Model Using Self-attention Mechanism." In: *ICCS*.

Gillani, Mehreen, Muhammad U. Ilyas, Saad Saleh, Jalal S. Alowibdi, Naif Aljohani, and Fahad S. Alotaibi (2017). "Post Summarization of Microblogs of Sporting Events." In: *Proceedings of the 26th International Conference on World Wide Web Companion*. Perth, Australia, pp. 59–68. ISBN: 978-1-4503-4914-7. DOI: 10.1145/3041021.3054146. URL: https://doi.org/10.1145/3041021.3054146.

Grewal, Monika, Muktabh Mayank Srivastava, Pulkit Kumar, and Srikrishna Varadarajan (2018). "Radnet: Radiologist level accuracy using deep learning for hemorrhage detection in ct scans." In: *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*. IEEE, pp. 281–284.

Guo, Han, Juan Cao, Yazi Zhang, Junbo Guo, and Jintao Li (2018). "Rumor detection with hierarchical social attention network." In: *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, pp. 943–951.

Guo, Weiwei and Mona Diab (2012). "Modeling sentences in the latent space." In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-volume 1*. Association for Computational Linguistics, pp. 864–872.

Gupta, Aditi and Ponnurangam Kumaraguru (2012). “Credibility Ranking of Tweets During High Impact Events.” In: *Proceedings of the 1st Workshop on Privacy and Security in Online Social Media*. PSOSM ’12. Lyon, France: ACM, 2:2–2:8. ISBN: 978-1-4503-1236-3. DOI: 10.1145/2185354.2185356. URL: http://doi.acm.org/10.1145/2185354.2185356.

Hamidian, Sardar and Mona T. Diab (2015). “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.

Hamidian, Sardar and Mona Diab (2016). “Rumor identification and belief investigation on twitter.” In: *Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pp. 3–8.

Han, Sooji and Fabio Ciravegna (2019). “Rumour Detection on Social Media for Crisis Management.” In: *Proceedings of the 16th International Conference on Information Systems for Crisis Response And Management*. ISCRAM, pp. 660–673.

Han, Sooji, Jie Gao, and Fabio Ciravegna (May 2019a). “Data augmentation for rumor detection using context-sensitive neural language model with large-scale credibility corpus.” In: *Proceedings of the 7th International Conference on Learning Representations. Learning from Limited Labeled Data: ICLR 2019 Workshop*. OpenReview. URL: http://eprints.whiterose.ac.uk/145668/.

— (2019b). “Neural language model based training data augmentation for weakly supervised early rumor detection.” In: *Proceedings of 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. IEEE.

Hasan, Mahmud, Mehmet A Orgun, and Rolf Schwitter (2016). “TwitterNews+: a framework for real time event detection from the Twitter data stream.” In: *International Conference on Social Informatics*. Springer, pp. 224–239.

— (2018). “A survey on real-time event detection from the twitter data stream.” In: *Journal of Information Science* 44.4, pp. 443–463.

Hastie, Trevor, Robert Tibshirani, Jerome Friedman, and James Franklin (2005). “The elements of statistical learning: data mining, inference and prediction.” In: *The Mathematical Intelligencer* 27.2, pp. 83–85.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2015). “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification.” In: *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034.

Helmstetter, Stefan and Heiko Paulheim (Aug. 2018). “Weakly Supervised Learning for Fake News Detection on Twitter.” In: *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. Vol. 00, pp. 274–277. DOI: 10.1109/ASONAM.2018.8508520. URL: doi.ieeecomputersociety.org/10.1109/ASONAM.2018.8508520.

Hernández-García, Alex and Peter König (2018). “Further advantages of data augmentation on convolutional neural networks.” In: *International Conference on Artificial Neural Networks*. Springer, pp. 95–103.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). “Long short-term memory.” In: *Neural computation* 9.8, pp. 1735–1780.

Hoi, Geraldine Wong Sak (2015). *D8. 2 Annotated Corpus of Newsworthy Rumours*. Tech. rep. Technical report, PHEME project deliverable.

Hsieh, Liang-Chi, Ching-Wei Lee, Tzu-Hsuan Chiu, and Winston H. Hsu (2012). "Live Semantic Sport Highlight Detection Based on Analyzing Tweets of Twitter." In: *2012 IEEE International Conference on Multimedia and Expo*, pp. 949–954.

Hu, Dichao (2019). "An introductory survey on attention mechanisms in nlp problems." In: *Proceedings of SAI Intelligent Systems Conference*. Springer, pp. 432–448.

Hu, Ying, Changjun Hu, Shushen Fu, Mingzhe Fang, and Wenwen Xu (2017). "Predicting Key Events in the Popularity Evolution of Online Information." In: *PloS one* 12.1, e0168749.

Ifrim, Georgiana, Bichen Shi, and Igor Brigadir (2014). "Event detection in twitter using aggressive filtering and hierarchical tweet clustering." In: *Second Workshop on Social News on the Web (SNOW), Seoul, Korea, 8 April 2014*. ACM.

Imran, Muhammad, Carlos Castillo, Fernando Diaz, and Sarah Vieweg (June 2015). "Processing Social Media Messages in Mass Emergency: A Survey." In: *ACM Comput. Surv.* 47.4, 67:1–67:38. ISSN: 0360-0300. DOI: 10.1145/2771588. URL: http://doi.acm.org/10.1145/2771588.

Ingram, Mathew (Sept. 2016). *Here's Why Trust in the Media Is at an All-Time Low*. URL: http://fortune.com/2016/09/15/trust-in-media/.

Inouye, David and Jugal K Kalita (2011). "Comparing twitter summarization algorithms for multiple post summaries." In: *2011 IEEE Third international conference on privacy, security, risk and trust and 2011 IEEE third international conference on social computing*. IEEE, pp. 298–306.

Jain, Suchita, Vanya Sharma, and Rishabh Kaushal (2016). "Towards Automated Real-time Detection of Misinformation on Twitter." In: *2016 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2016, Jaipur, India, September 21-24, 2016*, pp. 2015–2020. DOI: 10.1109/ICACCI.2016.7732347. URL: http://dx.doi.org/10.1109/ICACCI.2016.7732347.

Jiang, Ye, Johann Petrak, Xingyi Song, Kalina Bontcheva, and Diana Maynard (2019). "Team Bertha von Suttner at SemEval-2019 Task 4: Hyperpartisan News Detection using ELMo Sentence Representation Convolutional Network." In: *Proceedings of the 13th International Workshop on Semantic Evaluation*, pp. 840–844.

Jin, Zhiwei, Juan Cao, Han Guo, Yongdong Zhang, Yu Wang, and Jiebo Luo (2017a). "Detection and analysis of 2016 us presidential election related rumors on twitter." In: *International conference on social computing, behavioral-cultural modeling and prediction and behavior representation in modeling and simulation*. Springer, pp. 14–24.

Jin, Zhiwei, Juan Cao, Han Guo, Yongdong Zhang, and Jiebo Luo (2017b). "Multimodal Fusion with Recurrent Neural Networks for Rumor Detection on Microblogs." In: *Proceedings of the 25th ACM International Conference on Multimedia*. MM '17. Mountain View, California, USA: ACM, pp. 795–816. ISBN: 978-1-4503-4906-2. DOI: 10.1145/3123266.3123454. URL: http://doi.acm.org/10.1145/3123266.3123454.

Jin, Zhiwei, Juan Cao, Han Guo, Yongdong Zhang, Yu Wang, and Jiebo Luo (2017c). "Rumor Detection on Twitter Pertaining to the 2016 US Presidential Election." In: *arXiv preprint arXiv:1701.06250*.

Johnson, Justin M and Taghi M Khoshgoftaar (2019). “Survey on deep learning with class imbalance.” In: *Journal of Big Data* 6.1, p. 27.

Kıcıman, Emre (2010). “Language differences and metadata features on Twitter.” In: *Web N-gram Workshop*, p. 47.

Kim, Yoon (2014). “Convolutional neural networks for sentence classification.” In: *arXiv preprint arXiv:1408.5882*.

Knapp, Robert H (1944). “A psychology of rumor.” In: *Public opinion quarterly* 8.1, pp. 22–37.

Kobayashi, Sosuke (June 2018). “Contextual Augmentation: Data Augmentation by Words with Paradigmatic Relations.” In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 452–457. DOI: 10.18653/v1/N18-2072. URL: https://www.aclweb.org/anthology/N18-2072.

Kochkina, Elena, Maria Liakata, and Arkaitz Zubiaga (Aug. 2018a). “All-in-one: Multi-task Learning for Rumour Verification.” In: *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, pp. 3402–3413. URL: https://www.aclweb.org/anthology/C18-1288.

— (June 2018b). “PHEME dataset for Rumour Detection and Veracity Classification.” In: DOI: 10.6084/m9.figshare.6392078.v1. URL: "https://figshare.com/articles/PHEME_dataset_for_Rumour_Detection_and_Veracity_Classification/6392078".

Kolomiyets, Oleksandr, Steven Bethard, and Marie-Francine Moens (2011). “Model-portability experiments for textual temporal analysis.” In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*. Association for Computational Linguistics, pp. 271–276.

Kong, Shoubin, Fei Ye, Ling Feng, and Zhe Zhao (Dec. 2015). “Towards the Prediction Problems of Bursting Hashtags on Twitter.” In: *J. Assoc. Inf. Sci. Technol.* 66.12, pp. 2566–2579. ISSN: 2330-1635. DOI: 10.1002/asi.23342. URL: https://doi.org/10.1002/asi.23342.

Kotteti, Chandra Mouli Madhav, Xishuang Dong, and Lijun Qian (2018). “Multiple Time-Series Data Analysis for Rumor Detection on Social Media.” In: *2018 IEEE International Conference on Big Data (Big Data)*, pp. 4413–4419.

Kwon, Sejeong, Meeyoung Cha, Kyomin Jung, Wei Chen, and Yajun Wang (Dec. 2013). “Prominent Features of Rumor Propagation in Online Social Media.” In: *2013 IEEE 13th International Conference on Data Mining*, pp. 1103–1108. DOI: 10.1109/ICDM.2013.61.

Kwon, Sejeong, Meeyoung Cha, and Kyomin Jung (2017). “Rumor detection over varying time windows.” In: *PLOS ONE* 12.1, pp. 1–19. DOI: 10.1371/journal.pone.0168344. URL: https://doi.org/10.1371/journal.pone.0168344.

Lai, Siwei, Liheng Xu, Kang Liu, and Jun Zhao (2015). “Recurrent Convolutional Neural Networks for Text Classification.” In: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. AAAI’15. Austin, Texas: AAAI Press, pp. 2267–2273. ISBN: 0-262-51129-0. URL: http://dl.acm.org/citation.cfm?id=2886521.2886636.

Le, Quoc and Tomas Mikolov (2014). "Distributed representations of sentences and documents." In: *International conference on machine learning*, pp. 1188–1196.

Lee, Chung-Hong, Chih-Hong Wu, and Tzan-Feng Chien (2011). "BursT: a dynamic term weighting scheme for mining microblogging messages." In: *International Symposium on Neural Networks*. Springer, pp. 548–557.

Lee, Hyegyu and Hyun Jung Oh (2017). "Normative mechanism of rumor dissemination on Twitter." In: *Cyberpsychology, Behavior, and Social Networking* 20.3, pp. 164–171.

Leevy, Joffrey L, Taghi M Khoshgoftaar, Richard A Bauder, and Naeem Seliya (2018). "A survey on addressing high-class imbalance in big data." In: *Journal of Big Data* 5.1, p. 42.

Lewandowsky, Stephan, Ullrich KH Ecker, Colleen M Seifert, Norbert Schwarz, and John Cook (2012). "Misinformation and its correction: Continued influence and successful debiasing." In: *Psychological Science in the Public Interest* 13.3, pp. 106–131.

Li, Chenliang, Aixin Sun, and Anwitaman Datta (2012). "Twevent: segment-based event detection from tweets." In: *Proceedings of the 21st ACM international conference on Information and knowledge management*. ACM, pp. 155–164.

Li, Quanzhi, Xiaomo Liu, Rui Fang, Armineh Nourbakhsh, and Sameena Shah (2016). "User Behaviors in Newsworthy Rumors: A Case Study of Twitter." In: *Proceedings of the Tenth International Conference on Web and Social Media, Cologne, Germany, May 17-20, 2016.* Pp. 627–630. URL: http://www.aaai.org/ocs/index.php/ICWSM/ICWSM16/paper/view/13068.

Li, Quanzhi, Qiong Zhang, and Luo Si (July 2019). "Rumor Detection by Exploiting User Credibility Information, Attention and Multi-task Learning." In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 1173–1179. DOI: 10.18653/v1/P19-1113.

Li, Wei, Xiatian Zhu, and Shaogang Gong (2018). "Harmonious attention network for person re-identification." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2285–2294.

Lin, Chin-Yew (July 2004). "ROUGE: A Package for Automatic Evaluation of Summaries." In: *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, pp. 74–81. URL: https://www.aclweb.org/anthology/W04-1013.

Liu, Xiaohua, Yitong Li, Furu Wei, and Ming Zhou (2012). "Graph-based multi-tweet summarization using social signals." In: *Proceedings of COLING 2012*, pp. 1699–1714.

Liu, Xiaomo, Armineh Nourbakhsh, Quanzhi Li, Rui Fang, and Sameena Shah (2015). "Real-time rumor debunking on twitter." In: *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, pp. 1867–1870.

Liu, Yahui, Xiaolong Jin, Huawei Shen, and Xueqi Cheng (2017). "Do rumors diffuse differently from non-rumors? a systematically empirical analysis in sina weibo for rumor identification." In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, pp. 407–420.

Liu, Yang and Yi-Fang Brook Wu (2018). “Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks.” In: *Thirty-Second AAAI Conference on Artificial Intelligence*.

Liu, Yang, Songhua Xu, and Georgia Tourassi (2016). “Detecting Rumors Through Modeling Information Propagation Networks in a Social Media Environment.” In: *IEEE Transactions on Computational Social Systems* 3.2, pp. 46–62. ISSN: 2329-924X. DOI: 10.1109/TCSS.2016.2612980.

Longadge, Rushi and Snehalata Dongre (2013). “Class imbalance problem in data mining review.” In: *arXiv preprint arXiv:1305.1707*.

Lu, Xinghua, Bin Zheng, Atulya Velivelli, and ChengXiang Zhai (2006). “Enhancing text categorization with semantic-enriched representation and training data augmentation.” In: *Journal of the American Medical Informatics Association* 13.5, pp. 526–535.

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

Luque, Franco M and Juan Manuel Pérez (2018). “Atalaya at TASS 2018: Sentiment analysis with tweet embeddings and data augmentation.” In: *Proceedings of TASS* 2172.

Ma, Jing, Wei Gao, Zhongyu Wei, Yueming Lu, and Kam-Fai Wong (2015). “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, pp. 1751–1754.

Ma, Jing, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J. Jansen, Kam-Fai Wong, and Meeyoung Cha (2016). “Detecting Rumors from Microblogs with Recurrent Neural Networks.” In: *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. IJCAI’16. New York, New York, USA: AAAI Press, pp. 3818–3824. ISBN: 978-1-57735-770-4. URL: http://dl.acm.org/citation.cfm?id=3061053.3061153.

Ma, Jing, Wei Gao, and Kam-Fai Wong (July 2017). “Detect Rumors in Microblog Posts Using Propagation Structure via Kernel Learning.” In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, pp. 708–717. DOI: 10.18653/v1/P17-1066. URL: https://www.aclweb.org/anthology/P17-1066.

— (2018a). “Detect rumor and stance jointly by neural multi-task learning.” In: *Companion of the The Web Conference 2018 on The Web Conference 2018*. International World Wide Web Conferences Steering Committee, pp. 585–593.

— (July 2018b). “Rumor Detection on Twitter with Tree-structured Recursive Neural Networks.” In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, pp. 1980–1989. URL: https://www.aclweb.org/anthology/P18-1184.

Maas, Andrew L, Awni Y Hannun, and Andrew Y Ng (2013). “Rectifier nonlinearities improve neural network acoustic models.” In: *Proc. icml*. Vol. 30. 1, p. 3.

Maddock, Jim, Kate Starbird, Haneen J Al-Hassani, Daniel E Sandoval, Mania Orand, and Robert M Mason (2015). "Characterizing online rumoring behavior using multi-dimensional signatures." In: *Proceedings of the 18th ACM conference on computer supported cooperative work & social computing*. ACM, pp. 228–241.

Martins, Andre and Ramon Astudillo (2016). "From softmax to sparsemax: A sparse model of attention and multi-label classification." In: *International Conference on Machine Learning*, pp. 1614–1623.

Matsubara, Yasuko, Yasushi Sakurai, B Aditya Prakash, Lei Li, and Christos Faloutsos (2012). "Rise and fall patterns of information diffusion: model and implications." In: *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 6–14.

Matthews, Christopher (2013). *How Does One Fake Tweet Cause a Stock Market Crash?* URL: http://business.time.com/2013/04/24/how-does-one-fake-tweet-cause-a-stock-market-crash/.

Meladianos, Polykarpos, Giannis Nikolentzos, François Rousseau, Yannis Stavrakas, and Michalis Vazirgiannis (2015). "Degeneracy-Based Real-Time Sub-Event Detection in Twitter Stream." In: *ICWSM*. Ed. by Meeyoung Cha, Cecilia Mascolo, and Christian Sandvig. AAAI Press, pp. 248–257. ISBN: 978-1-57735-733-9. URL: http://dblp.uni-trier.de/db/conf/icwsm/icwsm2015.html#MeladianosNRSV15.

Meladianos, Polykarpos, Christos Xypolopoulos, Giannis Nikolentzos, and Michalis Vazirgiannis (2018a). "An Optimization Approach for Sub-event Detection and Summarization in Twitter." In: *ECIR*.

— (2018b). "An Optimization Approach for Sub-event Detection and Summarization in Twitter." In: *ECIR*.

Mendoza, Marcelo, Barbara Poblete, and Carlos Castillo (2010). "Twitter under crisis: Can we trust what we RT?" In: *Proceedings of the first workshop on social media analytics*. ACM, pp. 71–79.

Meyer, Robinson (May 2018). *The Grim Conclusions of the Largest-Ever Study of Fake News*. URL: https://www.theatlantic.com/technology/archive/2018/03/largest-study-ever-fake-news-mit-twitter/555104/.

Mihalcea, Rada and Paul Tarau (2004). "Textrank: Bringing order into text." In: *Proceedings of the 2004 conference on empirical methods in natural language processing*.

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean (2013). "Efficient estimation of word representations in vector space." In: *arXiv preprint arXiv:1301.3781*.

Miller, George (1998). *WordNet: An electronic lexical database*. MIT press.

Mitra, Tanushree and Eric Gilbert (2015). "Credbank: A large-scale social media corpus with associated credibility annotations." In: *Ninth International AAAI Conference on Web and Social Media*.

Moschitti, Alessandro (2004). "A study on convolution kernels for shallow semantic parsing." In: *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, p. 335.

Myers, Seth A and Jure Leskovec (2014). "The bursty dynamics of the twitter information network." In: *Proceedings of the 23rd international conference on World wide web*. ACM, pp. 913–924.

Nenkova, Ani and Lucy Vanderwende (2005). *The impact of frequency on summarization*. Tech. rep. Microsoft Research.

Nguyen, Tu Ngoc (2017). "A Comprehensive Low and High-level Feature Analysis for Early Rumor Detection on Twitter." In:

Nguyen, Tu Ngoc, Cheng Li, and Claudia Niederée (2017). "On Early-Stage Debunking Rumors on Twitter: Leveraging the Wisdom of Weak Learners." In: *Social Informatics - 9th International Conference, SocInfo 2017, Oxford, UK, September 13-15, 2017, Proceedings, Part II*, pp. 141–158. DOI: 10.1007/978-3-319-67256-4_13. URL: https://doi.org/10.1007/978-3-319-67256-4_13.

Nichols, Jeffrey, Jalal Mahmud, and Clemens Drews (2012). "Summarizing Sporting Events Using Twitter." In: *Proceedings of the 2012 ACM international conference on Intelligent User Interfaces*. ACM, pp. 189–198.

Olteanu, Alexandra, Sarah Vieweg, and Carlos Castillo (2015). "What to expect when the unexpected happens: Social media communications across crises." In: *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*. ACM, pp. 994–1009.

Ozturk, Pinar, Huaye Li, and Yasuaki Sakamoto (2015). "Combating rumor spread on social media: The effectiveness of refutation and warning." In: *2015 48th Hawaii International Conference on System Sciences*. IEEE, pp. 2406–2414.

Peng, Sinya, Vincent S. Tseng, Che-Wei Liang, and Man-Kwan Shan (2018). "Emerging Product Topics Prediction in Social Media Without Social Structure Information." In: *Companion Proceedings of the The Web Conference 2018*. WWW '18. Lyon, France, pp. 1661–1668. ISBN: 978-1-4503-5640-4. DOI: 10.1145/3184558.3191625. URL: https://doi.org/10.1145/3184558.3191625.

Pennington, Jeffrey, Richard Socher, and Christopher Manning (2014). "Glove: Global vectors for word representation." In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543.

Perone, Christian S, Roberto Silveira, and Thomas S Paula (2018). "Evaluation of sentence embeddings in downstream and linguistic probing tasks." In: *arXiv preprint arXiv:1806.06259*.

Peters, Matthew E, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer (2018). "Deep contextualized word representations." In: *arXiv preprint arXiv:1802.05365*.

Peterson, Warren A. and Noel P. Gist (1951). "Rumor and Public Opinion." In: *American Journal of Sociology* 57.2, pp. 159–167. ISSN: 00029602, 15375390. URL: http://www.jstor.org/stable/2772077.

Phuvipadawat, Swit and Tsuyoshi Murata (2010). "Breaking News Detection and Tracking in Twitter." In: *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology-Volume 03*. IEEE Computer Society, pp. 120–123.

Procter, Rob, Farida Vis, and Alex Voss (2013). "Reading the riots on Twitter: methodological innovation for the analysis of big data." In: *International journal of social research methodology* 16.3, pp. 197–214.

Prusa, Joseph, Taghi M Khoshgoftaar, David J Dittman, and Amri Napolitano (2015). "Using random undersampling to alleviate class imbalance on tweet sentiment data." In: *2015 IEEE international conference on information reuse and integration*. IEEE, pp. 197–202.

Qazvinian, Vahed, Emily Rosengren, Dragomir R. Radev, and Qiaozhu Mei (2011). "Rumour Has It: Identifying Misinformation in Microblogs." In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. EMNLP '11. Edinburgh, United Kingdom: Association for Computational Linguistics, pp. 1589–1599. ISBN: 978-1-937284-11-4. URL: http://dl.acm.org/citation.cfm?id=2145432.2145602.

Quinlan, Ross (1993). *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann Publishers.

Radford, Alec, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever (2018). "Improving language understanding by generative pre-training." In:

Radford, Alec, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever (2019). "Language Models are Unsupervised Multitask Learners." In:

Ratner, Alex, Stephen Bach, Paroma Varma, and Chris Ré (2017). *Weak Supervision: The New Programming Paradigm for Machine Learning*. URL: http://ai.stanford.edu/blog/weak-supervision/.

Resnick, Paul, Samuel Carton, Souneil Park, Yuncheng Shen, and Nicole Zeffer (2014). "Rumorlens: A system for analyzing the impact of rumors and corrections in social media." In: *Computational Journalism Conference*.

Ries, Tonia E., Daivd M. Bersoff, Cody Amstrong, Sarah Adkins, and Jamis Bruening (Feb. 2018). *The 2018 Edelman Trust Barometer*. URL: https://www.edelman.com/trust-barometer.

Ritter, Alan, Oren Etzioni, Sam Clark, et al. (2012). "Open domain event extraction from twitter." In: *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 1104–1112.

Robertson, Stephen (2004). "Understanding inverse document frequency: on theoretical arguments for IDF." In: *Journal of documentation* 60.5, pp. 503–520.

Rosnow, Ralph L. (May 1991). "Inside rumor: A personal journey." In: *American Psychologist* 46, pp. 484–496. DOI: 10.1037/0003-066X.46.5.484.

Ruchansky, Natali, Sungyong Seo, and Yan Liu (2017). "CSI: A Hybrid Deep Model for Fake News Detection." In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. CIKM '17. Singapore, Singapore: ACM, pp. 797–806. ISBN: 978-1-4503-4918-5. DOI: 10.1145/3132847.3132877. URL: http://doi.acm.org/10.1145/3132847.3132877.

Rudra, Koustav, Subham Ghosh, Niloy Ganguly, Pawan Goyal, and Saptarshi Ghosh (2015). "Extracting Situational Information from Microblogs During Disaster Events: A Classification-Summarization Approach." In: *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. CIKM '15. Melbourne, Australia: ACM, pp. 583–592. ISBN: 978-1-4503-3794-6. DOI: 10.1145/2806416.2806485. URL: http://doi.acm.org/10.1145/2806416.2806485.

Rudra, Koustav, Siddhartha Banerjee, Niloy Ganguly, Pawan Goyal, Muhammad Imran, and Prasenjit Mitra (2016). "Summarizing Situational Tweets in Crisis Scenario." In: *Proceedings of the 27th ACM Conference on Hypertext and Social Media*. HT '16. Halifax, Nova Scotia, Canada: ACM, pp. 137–147. ISBN: 978-1-4503-4247-6. DOI: 10.1145/2914586.2914600. URL: http://doi.acm.org/10.1145/2914586.2914600.

Rudra, Koustav, Pawan Goyal, Niloy Ganguly, Prasenjit Mitra, and Muhammad Imran (2018). "Identifying Sub-events and Summarizing Disaster-Related Information from Microblogs." In: *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. SIGIR '18. Ann Arbor, MI, USA: ACM, pp. 265–274. ISBN: 978-1-4503-5657-2. DOI: 10.1145/3209978.3210030. URL: http://doi.acm.org/10.1145/3209978.3210030.

Saríyüce, Ahmet Erdem, Buğra Gedik, Gabriela Jacques-Silva, Kun-Lung Wu, and Ümit V Çatalyürek (2013). "Streaming algorithms for k-core decomposition." In: *Proceedings of the VLDB Endowment* 6.6, pp. 433–444.

Seidman, Stephen B (1983a). "Network structure and minimum degree." In: *Social networks* 5.3, pp. 269–287.

— (1983b). "Network structure and minimum degree." In: *Social networks* 5.3, pp. 269–287.

Severyn, Aliaksei and Alessandro Moschitti (2015a). "Learning to Rank Short Text Pairs with Convolutional Deep Neural Networks." In: *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '15. Santiago, Chile: ACM, pp. 373–382. ISBN: 978-1-4503-3621-5. DOI: 10.1145/2766462.2767738. URL: http://doi.acm.org/10.1145/2766462.2767738.

— (2015b). "Learning to rank short text pairs with convolutional deep neural networks." In: *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*. ACM, pp. 373–382.

Shalev-Shwartz, Shai and Shai Ben-David (2014). *Understanding machine learning: From theory to algorithms*. Cambridge university press.

Shamma, David A, Lyndon Kennedy, and Elizabeth F Churchill (2009). "Tweet the debates: understanding community annotation of uncollected sources." In: *Proceedings of the first SIGMM workshop on Social media*. ACM, pp. 3–10.

Sharifi, Beaux P, David I Inouye, and Jugal K Kalita (2013). "Summarization of twitter microblogs." In: *The computer journal* 57.3, pp. 378–402.

Shibutani, Tamotsu (1966). *Improvised news*. Ardent Media.

Shin, Jieun, Lian Jian, Kevin Driscoll, and François Bar (2018a). "The diffusion of misinformation on social media: Temporal pattern, message, and source." In: *Computers in Human Behavior* 83, pp. 278–287.

Shin, Jieun, Lian Jian, Kevin Driscoll, and François Bar (2018b). "The diffusion of misinformation on social media: Temporal pattern, message, and source." In: *Computers in Human Behavior* 83, pp. 278 –287. ISSN: 0747-5632. DOI: https://doi.org/10.1016/j.chb.2018.02.008. URL: http://www.sciencedirect.com/science/article/pii/S0747563218300669.

Shu, Kai, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu (2017). "Fake news detection on social media: A data mining perspective." In: *ACM SIGKDD Explorations Newsletter* 19.1, pp. 22–36.

Shu, Kai, Suhang Wang, and Huan Liu (2018). "Understanding user profiles on social media for fake news detection." In: *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*. IEEE, pp. 430–435.

Simonyan, Karen and Andrew Zisserman (2014). "Very deep convolutional networks for large-scale image recognition." In: *arXiv preprint arXiv:1409.1556*.

Spiegel, Uriel, Tchai Tavor, and Joseph Templeman (2010). "The effects of rumours on financial market efficiency." In: *Applied Economics Letters* 17.15, pp. 1461–1464.

Srijith, P.K., Mark Hepple, Kalina Bontcheva, and Daniel Preotiuc-Pietro (July 2017). "Sub-story Detection in Twitter with Hierarchical Dirichlet Processes." In: *Inf. Process. Manage.* 53.4, pp. 989–1003. ISSN: 0306-4573. DOI: 10.1016/j.ipm.2016.10.004. URL: https://doi.org/10.1016/j.ipm.2016.10.004.

Stewart, Avaré, Sara Romano, Nattiya Kanhabua, Sergio Di Martino, Wolf Siberski, Antonino Mazzeo, Wolfgang Nejdl, and Ernesto Diaz-Aviles (2016). "Why is it Difficult to Detect Sudden and Unexpected Epidemic Outbreaks in Twitter?" In: *CoRR* abs/1611.03426. arXiv: 1611.03426. URL: http://arxiv.org/abs/1611.03426.

Sun, Shengyun, Hongyan Liu, Jun He, and Xiaoyong Du (2013). "Detecting Event Rumors on Sina Weibo Automatically." In: *Web Technologies and Applications*. Ed. by Yoshiharu Ishikawa, Jianzhong Li, Wei Wang, Rui Zhang, and Wenjie Zhang. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 120–131. ISBN: 978-3-642-37401-2.

Sunstein, Cass R. (2010). *On Rumours: How Falsehoods Spread, Why We Believe Them, What Can Be Done*. Penguin Books Limited. ISBN: 9780141044293. URL: https://books.google.co.uk/books?id=tB6RhA5eiwkC.

Tachibana, Hideyuki, Katsuya Uenoyama, and Shunsuke Aihara (2018). "Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention." In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 4784–4788.

Tanaka, Yuko, Yasuaki Sakamoto, and Hidehito Honda (Jan. 2014). "The Impact of Posting URLs in Disaster-Related Tweets on Rumor Spreading Behavior." In: *2014 47th Hawaii International Conference on System Sciences*, pp. 520–529. DOI: 10.1109/HICSS.2014.72.

Tarnpradab, Sansiri and Kien A. Hua (2019). "Attention Based Neural Architecture for Rumor Detection with Author Context Awareness." In: *CoRR* abs/1910.01458. arXiv: 1910.01458. URL: http://arxiv.org/abs/1910.01458.

Tixier, Antoine, Fragkiskos Malliaros, and Michalis Vazirgiannis (2016). "A graph degeneracy-based approach to keyword extraction." In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1860–1870.

Tolosi, Laura, Andrey Tagarev, and Georgi Georgiev (2016). "An analysis of event-agnostic features for rumour classification in twitter." In: *Tenth International AAAI Conference on Web and Social Media*.

Unankard, Sayan, Xue Li, and Mohamed A Sharaf (2015). "Emerging event detection in social networks with location sensitivity." In: *World Wide Web* 18.5, pp. 1393–1417.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin (2017). "Attention is all you need." In: *Advances in neural information processing systems*, pp. 5998–6008.

Veyseh, Amir Pouran Ben, My T Thai, Thien Huu Nguyen, and Dejing Dou (2019a). "Rumor Detection in Social Networks via Deep Contextual

Modeling." In: *Proceedings of 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. IEEE.

— (2019b). "Rumor Detection in Social Networks via Deep Contextual Modeling." In: *Proceedings of 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. IEEE.

Vijayaraghavan, Prashanth, Ivan Sysoev, Soroush Vosoughi, and Deb Roy (2016). "Deepstance at semeval-2016 task 6: Detecting stance in tweets using character and word-level cnns." In: *arXiv preprint arXiv:1606.05694*.

Vosoughi, Soroush (2015). "Automatic detection and verification of rumors on Twitter." PhD thesis. Massachusetts Institute of Technology.

Vosoughi, Soroush, Prashanth Vijayaraghavan, and Deb Roy (2016). "Tweet2vec: Learning tweet embeddings using character-level cnn-lstm encoder-decoder." In: *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, pp. 1041–1044.

Vosoughi, Soroush, Mostafa'Neo' Mohsenvand, and Deb Roy (2017). "Rumor gauge: Predicting the veracity of rumors on Twitter." In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 11.4, p. 50.

Wang, Fei, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang (2017). "Residual attention network for image classification." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3156–3164.

Wang, Feng and David MJ Tax (2016). "Survey on the attention based RNN model and its applications in computer vision." In: *arXiv preprint arXiv:1601.06823*.

Wang, Senzhang, Zhao Yan, Xia Hu, S Yu Philip, Zhoujun Li, and Biao Wang (2016). "CPB: a classification-based approach for burst time prediction in cascades." In: *Knowledge and Information Systems* 49.1, pp. 243–271.

Wang, Tao, David J. Wu, Adam Coates, and Andrew Y. Ng (Nov. 2012). "End-to-end text recognition with convolutional neural networks." In: *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pp. 3304–3308.

Weng, Jianshu and Bu-Sung Lee (2011). "Event detection in twitter." In: *Fifth international AAAI conference on weblogs and social media*.

Woo, Jiyoung, Jaebong Son, and Hsinchun Chen (2011). "An SIR model for violent topic diffusion in social media." In: *Proceedings of 2011 IEEE International Conference on Intelligence and Security Informatics*. IEEE, pp. 15–19.

Wu, Ke, Yang Song, and Zhu Kenny Q. (2015). "False rumors detection on Sina Weibo by propagation structures." In: *2015 IEEE 31st International Conference on Data Engineering*, pp. 651–662. DOI: 10.1109/ICDE.2015.7113322.

Xing, Chen, Yuan Wang, Jie Liu, Yalou Huang, and Wei-Ying Ma (2016). "Hashtag-based sub-event discovery using mutually generative lda in twitter." In: *Thirtieth AAAI Conference on Artificial Intelligence*.

Xiong, Fei, Yun Liu, Zhen-jiang Zhang, Jiang Zhu, and Ying Zhang (2012). "An information diffusion model based on retweeting mechanism for online social media." In: *Physics Letters A* 376.30-31, pp. 2103–2108.

Xu, Wei, Alan Ritter, Chris Callison-Burch, William B. Dolan, and Yangfeng Ji (2014). "Extracting Lexically Divergent Paraphrases from Twitter." In: *Transactions of the Association for Computational Linguistics*. URL: http:

//www.cis.upenn.edu/~xwe/files/tacl2014-extracting-paraphrases-from-twitter.pdf.

Xu, Wen and He Chen (2015). "Scalable rumor source detection under independent cascade model in online social networks." In: *2015 11th International Conference on Mobile Ad-hoc and Sensor Networks (MSN)*. IEEE, pp. 236–242.

Yang, Fan, Yang Liu, Xiaohui Yu, and Min Yang (2012). "Automatic Detection of Rumor on Sina Weibo." In: *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*. MDS '12. Beijing, China: ACM, 13:1–13:7. ISBN: 978-1-4503-1546-3. DOI: 10.1145/2350190.2350203. URL: http://doi.acm.org/10.1145/2350190.2350203.

Yang, Jaewon and Jure Leskovec (2011). "Patterns of temporal variation in online media." In: *Proceedings of the fourth ACM international conference on Web search and data mining*. ACM, pp. 177–186.

Yang, Zichao, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy (2016a). "Hierarchical attention networks for document classification." In: *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pp. 1480–1489.

Yang, Zichao, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola (2016b). "Stacked attention networks for image question answering." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 21–29.

Yin, Jie, Sarvnaz Karimi, Andrew Lampert, Mark Cameron, Bella Robinson, and Robert Power (Nov. 2012). "Using Social Media to Enhance Emergency Situation Awareness." In: *IEEE Intelligent Systems* 27.6, pp. 52–59. ISSN: 1541-1672. DOI: 10.1109/MIS.2012.6.

Yu, Feng, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan (2017). "A Convolutional Approach for Misinformation Identification." In: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pp. 3901–3907. DOI: 10.24963/ijcai.2017/545. URL: https://doi.org/10.24963/ijcai.2017/545.

Zamani, Somayeh, Masoud Asadpour, and Dara Moazzami (2017). "Rumor detection for Persian Tweets." In: *2017 Iranian Conference on Electrical Engineering (ICEE)*. IEEE, pp. 1532–1536.

Zeng, Haoyang and David K Gifford (2019). "DeepLigand: accurate prediction of MHC class I ligands using peptide embedding." In: *Bioinformatics* 35.14, pp. i278–i283.

Zhang, Xiang, Junbo Zhao, and Yann LeCun (2015a). "Character-level convolutional networks for text classification." In: *Advances in neural information processing systems*, pp. 649–657.

Zhang, Xiaoming, Xiaoming Chen, Yan Chen, Senzhang Wang, Zhoujun Li, and Jiali Xia (2015b). "Event detection and popularity prediction in microblogging." In: *Neurocomputing* 149, pp. 1469–1480.

Zhang, Zili, Ziqiong Zhang, and Hengyun Li (2015c). "Predictors of the authenticity of Internet health rumours." In: *Health Information & Libraries Journal* 32.3, pp. 195–205.

Zhao, Zhe, Paul Resnick, and Qiaozhu Mei (2015). "Enquiring Minds: Early Detection of Rumors in Social Media from Enquiry Posts." In: *Proceedings of the 24th International Conference on World Wide Web*. WWW '15. Florence, Italy: International World Wide Web Conferences Steering Committee, pp. 1395–1405. ISBN: 978-1-4503-3469-3. DOI: 10.1145/2736277.2741637. URL: https://doi.org/10.1145/2736277.2741637.

Zheltukhina, Marina R, Gennady G Slyshkin, Elena B Ponomarenko, Maryana V Busygina, and Anatoly V Omelchenko (2016). "Role of Media Rumors in the Modern Society." In: *International Journal of Environmental and Science Education* 11.17, pp. 10581–10589.

Zhong, Guoqiang, Li-Na Wang, Xiao Ling, and Junyu Dong (2016). "An overview on data representation learning: From traditional feature learning to recent deep learning." In: *The Journal of Finance and Data Science* 2.4, pp. 265–278.

Zhou, Kaimin, Chang Shu, Binyang Li, and Jey Han Lau (2019a). "Early Rumour Detection." In: *Proceedings of the 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Minneapolis, MN, USA, pp. 2180–2189.

— (June 2019b). "Early Rumour Detection." In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 1614–1623. DOI: 10.18653/v1/N19-1163. URL: https://www.aclweb.org/anthology/N19-1163.

Zubiaga, Arkaitz (2018). "A longitudinal assessment of the persistence of twitter datasets." In: *Journal of the Association for Information Science and Technology* 69.8, pp. 974–984.

Zubiaga, Arkaitz, Damiano Spina, Enrique Amigó, and Julio Gonzalo (2012). "Towards real-time summarization of scheduled events from twitter streams." In: *Proceedings of the 23rd ACM conference on Hypertext and social media*. ACM, pp. 319–320.

Zubiaga, Arkaitz, Peter Tolmie, Maria Liakata, and Rob Procter (2014). "D2. 1 development of an annotation scheme for social media rumours." In:

Zubiaga, Arkaitz, Maria Liakata, Rob Procter, Kalina Bontcheva, and Peter Tolmie (May 2015). "Crowdsourcing the Annotation of Rumourous Conversations in Social Media." In: *WWW 2015 Companion*. Florence, Italy: ACM. ISBN: 978-1-4503-3473-0. URL: http://wrap.warwick.ac.uk/66846/ (visited on 11/24/2016).

Zubiaga, Arkaitz, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Peter Tolmie (2016a). "Analysing how people orient to and spread rumours in social media by looking at conversational threads." In: *PloS one* 11.3, e0150989.

Zubiaga, Arkaitz, Maria Liakata, and Rob Procter (2016b). "Learning Reporting Dynamics during Breaking News for Rumour Detection in Social Media." In: *CoRR* abs/1610.07363.

Zubiaga, Arkaitz, Elena Kochkina, Maria Liakata, Rob Procter, and Michal Lukasik (2016c). "Stance Classification in Rumours as a Sequential Task Exploiting the Tree Structure of Social Media Conversations." In: *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pp. 2438–2448. URL: http://aclweb.org/anthology/C/C16/C16-1230.pdf.

Zubiaga, Arkaitz, Maria Liakata, and Rob Procter (2017). "Exploiting Context for Rumour Detection in Social Media." In: *SocInfo*.

Zubiaga, Arkaitz, Ahmet Aker, Kalina Bontcheva, Maria Liakata, and Rob Procter (Feb. 2018a). “Detection and Resolution of Rumours in Social Media: A Survey.” In: *ACM Comput. Surv.* 51.2, 32:1–32:36. ISSN: 0360-0300. DOI: 10.1145/3161603. URL: http://doi.acm.org/10.1145/3161603.

Zubiaga, Arkaitz, Elena Kochkina, Maria Liakata, Rob Procter, Michal Lukasik, Kalina Bontcheva, Trevor Cohn, and Isabelle Augenstein (2018b). “Discourse-aware rumour stance classification in social media using sequential classifiers.” In: *Information Processing & Management* 54.2, pp. 273–290.