Rumor Identification and Verification on Social Media

by Sardar Hamidian

B.S. in Software Engineering, May 2010, Azad Tehran Markaz University
M.S. in Computer Science, December 2013, The George Washington University

A Thesis submitted to

The Faculty of
The School of Engineering and Applied Science
of The George Washington University
in partial satisfaction of the requirements
for the degree of Doctor of Philosophy

August 31, 2019

Dissertation directed by

Mona T. Diab
Professor of Computer Science

The School of Engineering and Applied Science of The George Washington University certifies that Sardar Hamidian has passed the Final Examination for the degree of Doctor of Philosophy as of June 13, 2019. This is the final and approved form of the dissertation.

Rumor Identification and Verification on Social Media

Sardar Hamidian

Dissertation Research Committee:

Mona T. Diab, Professor of Computer Science, Dissertation Director

Abdou Youssef, Professor of Computer Science, Committee Member

Robert Pless, Professor of Computer Science, Committee Member

David Broniatowski, Assistant Professor of Engineering Management and System Engineering, Committee Member

© Copyright 2019 by Sardar Hamidian
All rights reserved

Acknowledgments

I would like to thank my Ph.D. advisor Professor Mona Diab who supported me and gave me the opportunity to work with her on my Ph.D. journey. I am also grateful to members of my committee, Professors Abdou Youssef, Robert Pless, David Broniatowski, and Claire Monteleoni, for the great discussions and feedback.

I would like to thank my fellow doctoral students for their feedback, cooperation, and of course, friendship. Mahmoud Ghoneim, Hanan Aldarmaki, Ayah Zirikly, Shabnam Tafreshi, Sawsan Alqahtani, Nada Almarwani, Maryam Aminian, Ali Seyfi, Fahad Alghamdi, Pedram Hosseini, Jared Mowery, Abdelati Hawwari, and Mohamed ElBadrashiny. Special thanks to Hanan Aldarmaki, Sawsan Alqahtani, and Shabnam Tafreshi for their productive feedback in writing my thesis. Thanks to my friends and mentors Hans Sayyadi and Aria Pezeshk for their support and advice in my internship programs. In addition, I would like to express my gratitude to my fiance Armaghan Mehrabian who supported me all along in this fabulous journey.

Last but not least, I would like to thank my family: my parents and my brother for supporting me spiritually throughout these years, and just for being.

Abstract

Rumor Identification and Verification on Social Media

Social media users spend many hours each day reading, posting, and searching for news on microblogging platforms. Social media has emerged as a key channel for information. However, the inherently unmoderated features of social media render identifying misinformation ever more challenging, hence locating sound information can be elusive. This thesis assesses the problem of rumor identification, verification, and proliferation on social media platforms. Our primary objective is to identify whether the insights gained from this study can be used to build computational systems that can automatically identify and verify rumors between the emergence period and after verification. We mainly investigate rumor phenomena associated with people, entities, or events of public interest. We study the impact of different types of attributes in rumor identification, spread, and verification tasks, namely: level of committed belief, subjectivity, pragmatic, user provenance, meta-linguistic, and network propagation features. Accordingly the contributions of this thesis are as follow:

- A novel machine learning pipeline for rumor tracking by leveraging Tweet Latent Vector (TLV) instead of the conventional approaches like TF-IDF or Bag Of Words (BOW).

- More than 10,000 rumor collected and annotated automatically from 2014 to 2016. The tweet ID and rumor tag are available for research use.

- The first word-level and document level Neural Network (NN) Multi-Genre Level of Committed Belief (LCB) tagging model.

- A novel domain-independent approach for identification of unseen emerging rumor. The proposed methodology differs significantly from conventional methods

which are only applicable to identify the unfolded longstanding rumors which are known a priori.

- The thesis investigates the rumor veracity task by studying the interpretability of contextual and non-contextual signals in rumor conversations. The thesis also compares the results obtained by several conventional and neural network machine learning systems. The proposed models in this work could outperform the current state-of-the-art performance in rumor veracity classification task.

- A novel NN multi-task learning framework to attain the stance orientation of the people in rumor conversation.

Table of Contents

Acknowledgments . . . . . **iv**

Abstract . . . . . **v**

List of Figures . . . . . **xi**

List of Tables . . . . . **xv**

List of Abbreviations . . . . . **xix**

1 Introduction **1**

1.1 Terminology 3

1.2 Twitter 4

1.3 Contributions 5

1.4 Thesis Layout and Brief Overview of Chapters 6

2 Background **8**

2.1 Introduction 8

2.2 Rumor in Psychology 8

2.3 Rumor on Social media 9

2.3.1 Rumor Related Corpora 10

2.3.2 Rumor Tracking and Identification 15

2.3.3 Rumor Veracity 20

2.4 Summary 25

3 Rumor Detection and Opinion Classification on Twitter **26**

3.1 Rumor Detection and Opinion Classification (RDC) 26

3.2 Problem Definition 26

3.3 Dataset 28

3.4 Experimental Conditions 30
3.4.1 Machine Learning Frameworks 31
3.4.2 Feature Extraction 31
3.4.3 Experimental Design 35
3.4.4 Results 37
3.5 TLV Approach 38
3.5.1 Silver Data 40
3.5.2 Features 40
3.5.3 Experimental Design 42
3.5.4 Experiments and Evaluations 43
3.5.5 Results 43
3.6 Summary 45

4 Committed Belief Tagging 46

4.1 Introduction 46
4.2 Related Work 48
4.3 Approach 49
4.3.1 Dataset 49
4.3.2 LCB Tagging for (*Propositions*) 51
4.4 Experimental Setup 55
4.4.1 Training 55
4.4.2 Datasets 56
4.4.3 Experiments 56
4.4.4 Results and Evaluation 59
4.4.5 Summary 63
4.5 Single-step versus Joint LCB 63
4.6 Experimental Setup 63
4.6.1 Datasets 63

4.6.2 Experiments 64
4.6.3 Results and Evaluations 66
4.6.4 Error Analysis 66
4.7 Summary 67

5 Detecting Rumors on Early Stages in Social Media 68

5.1 Introduction 68
5.2 Related Work 69
5.3 Approach 70
5.4 Rumor Analysis 71
5.4.1 Pre-processing 72
5.4.2 Features Extraction 74
5.4.3 Time-series Rumor Analysis 79
5.5 Rumor Detection 86
5.5.1 Experimental Design 87
5.5.2 Baseline 87
5.5.3 Experiments 87
5.5.4 Evaluation 89
5.6 Summary 94

6 Analyzing the Rumor Veracity on Social Media 96

6.1 Introduction 96
6.2 Problem Statement 97
6.2.1 Data 97
6.2.2 Reaction to True versus False Rumor in Early Hours 98
6.2.3 Reply, Response and Re-Tweet in Early Hours 99
6.3 Approach 100
6.3.1 Pre-processing 100

6.3.2 Feature Extraction . . . 101
6.3.3 Experimental Setup . . . 102
6.3.4 Feature Interpretability . . . 102
6.3.5 Insight from Selected Features . . . 105
6.4 Rumor Veracity Classification . . . 107
6.4.1 Data . . . 107
6.4.2 Experimental Setup . . . 109
6.4.3 Machine Learning . . . 111
6.4.4 Results and Evaluation . . . 112
6.4.5 Multi-Task Learning (MTL) . . . 118
6.5 Task 2: Stance . . . 118
6.5.1 Data: SemEval 2019 . . . 119
6.5.2 Approach . . . 121
6.5.3 Model Training . . . 124
6.5.4 Result and Evaluation . . . 125
6.6 Summary . . . 126
7 Conclusions & Future Directions . . . 127
7.1 Conclusions . . . 127
7.2 Problems and Limitations . . . 129
7.3 Research Contributions . . . 129
7.3.1 Publications . . . 131
7.4 Future Directions . . . 132
Bibliography . . . 133

List of Figures

2.1 Tree-structured Recursive Neural Networks for identifying the veracity of the source 24

3.1 The overall pipeline for RDC 26
3.2 Tweet distribution for the Palin rumor collection within a 6 month period 34
3.3 The Average F-measure and precision of SRDC and TRDC classifications employing each group of features: Content, Pragmatic, Network 38
3.4 The overall accuracy in different experiments with and without preprocessing 39
3.5 The overall pipeline for TLV 39
3.6 The CB and NCB score for the Obama Dataset in 2010, 2014, and 2016 44

4.1 (A)The embedding of each word is generated by employing the output of a character embedding using a Bidirectional Long Short Term Memory (Bi-LSTM) layer concatenated with pre-trained embeddings for each word. (B) The word embedding from (A) is fed to the Bi-LSTM network for the LCB task. The proposed model contains a Bi-LSTM followed by a Linear chain Conditional Random field (CRF) classifier. 49
4.2 The word embedding is fed to the multi-task Bi-LSTM network for LCB and GD tasks. The proposed model contains a shared Bi-LSTM with two Bi-LSTM layers of each task following corresponding CRFs and a Softmax classification layer for LCB and GD, respectively. 54

4.3 The different sections of the proposed two-step framework for LCB tagging illustrated here. The right network is a binary sequence tagging network for Proposition (prp)/Event identification task. The left network gets the prediction output of the prp/Event identifier model. Both networks using the concatenation of Character embedding trained by Bi-LSTM and Glove pre-trained word embedding. 64

5.1 This plot shows, the KDE of different types of tweets rumors versus nonrumor conversations. The x-axis is indicative of the response time . . 73

5.2 The time-series model for feature extraction of different events (E_i) and subsequent rumour and non-rumour conversations (C_i) on three types of tweet including the Reply (RP), Response (RS), and Re-tweets (RT). . 73

5.3 The degree of reported belief in the source of the stories of the PHEME dataset. 81

5.4 Correlation matrix of the proposed set of features in T=0 on the source of conversations. 81

5.5 Neural Network architecture for rumor identification task 89

5.6 Tweet types, showing the performance of the model using the features extracted from different types, e.g., ret (Re-tweets), rep (Reply), res (Response), and all features combined. Frame versus Cumulative shows the performance of the model using the features extracted from related time frames, cumulatively, and all features, which is the combined set. . 91

5.7 The confusion matrix of different classifiers which are trained by certain category of features. 92

5.8 The left figure shows the performance (F-Micro) per feature category for different types of tweet in different time frames. Right figure shows the performance per event in various time frames for different types of tweet 94

6.1 This plot shows the kernel density estimation of the false (red) versus true (blue) in the 2 hours of rumor propagation for the top five events. The x-axis shows the time elapse of the replied tweets (hours). The y-axis is the estimated density value of KDE. The top five stories are those with the highest number of tweets in the PHEME data set. 99
6.2 This plot shows the kernel density estimation of the false (red) versus true (blue) in the first 75% (2 hour interval) of rumor propagation for different tweet types. The x-axis indicates the response time difference to the source of the conversation (hours). The y-axis is the estimated density value of KDE. 99
6.3 Examples from the rumor conversations and source tweets with a veracity value. 108
6.4 Performance of the proposed model on development and test set for each platform by the features extracted over the time from the conversation. The y-axis showing the i time index in T_i. For example, T_1 refers to the first second or only the features extracted from the source of the conversation, and T_2 refers to the [0-6] time frame. 112
6.5 Performance of each category of features on test set extracted over the time from the rumor conversation. The y-axist shows the i time index and the x-axis shows the F-macro. 113
6.6 Performance of each category of features per class in different time frames. 114
6.7 Confusion matrices on training and testing across platforms. 116
6.8 Performance of different conventional machine learning models applied in the all the experiments conducted for rumour veracity tasks on dev and test set. 117
6.9 The Neural Network architecture including the single and multi-task learning pipeline with the cross-target self attention mechanism 118

6.10 Examples of tree-structured of rumor thread discussing the veracity of a rumour, where the label associated with each tweet is the target of the stance classification 120
6.11 The distribution percentage of stance and verification tags on Twitter and Reddit dataset.“TaskB_Source" (exp. False_Source) indicate the verification tag of the source of conversation. 121

List of Tables

2.1 List of publicly available Rumor (RU) and Fake News (FN) dataset with their stats 14

2.2 The annotation information of the publicly available corpora. 14

3.1 List of rumor examples and their corresponding queries used to collect data from Twitter [**?**]]Vahed 28

3.2 Rumor Detection Annotation Guidelines 28

3.3 Final list of tweets that could be extracted using Twitter API 29

3.4 Final list of used features '*' marked features are the appended set of features 32

3.5 Number of features and labels used in single step and two step classifications 36

3.6 F-measure results of SRDC and TRDC methods employing 15 features and VAR11 features 37

3.7 List of Tweets in Silver Data 40

3.8 Labels Used in Rumor Retrieval and Rumor Type Classification for Silver Data 40

3.9 Precision and Recall Of RR task by Employing TLV+Content Unigram, RDC, and V11 is reported as Mean Average Precision (MAP) 44

4.1 $LDC_{proposition}$ training and evaluation set statistics. The training/dev set was split into 90% for training and 10% for development maintaining the distributional proportions across LCB categories. 56

4.2 Training and evaluation set statistics of belief categories in Discussion Forum (DF), Newswire (NW), and Mix corpora for LDC_{event}. The training/dev set was split into 90% for training and 10% for development maintaining the distributional proportions across LCB categories. 57

4.3 Results obtained by different experimental setup for LCB task including the (Ch)aracter Embedding, (Pre)trained Word Embedding (WE), and Contextual Representation (CR) on $LDC_{(Pr)oposition}$ and $LDC_{(Ev)ent}corpora$. 59
4.4 Contrasting Results per LCB tag in P15 model and our best condition neural network LCB model, WE_{Ch}+$WE_{Pre+trained}$+CR, on $LDC_{Proposition}$. 61
4.5 The results of single-genre and multi-genre experiments separating out DF and NW datasets in LDC_{Event} corpus 61
4.6 Results of employing the MTL on LCB Event task (LCB) and Genre Detection on $LDC_{Event}corpus$ 62
4.7 Overall F-Measure results per LCB category by leveraging the MTL framework for LCB events task (LCB) and Genre Detection on $LDC_{Event}corpus$ 62
4.8 Results per LCB categories by applying single-step and two-step approaches on LDC_prp and LDC_ev corpora. 64
4.9 Leveraging larger dataset LCB_prp on LCB_event using a two-step approach. F (Beta=1)-measure as the harmonic mean between (P)recision and (R)ecall. 65
4.10 List of top N errors for each task in single-step and two-step method on the two corpora- 67

5.1 Statistics for the PHEME dataset on Rum/NRum for each of the 5 events. 71
5.2 Different category of features. 75
5.3 Descriptive statistics on pragmatic features from the source of the rumor and nonrumor conversations as well as K-S tests of the differences in the distributions of these measures. 80
5.4 Descriptive statistics on pragmatic features from the reacted tweets to the rumor and nonrumor conversations as well as K-S tests of the differences in the distributions of these measures. 80

5.5 The Logistic regression models estimating the likelihood of a source of conversation being Rumor (0) or Non-Rumor (1) by using the variables collected from different type of tweets in different time frames T=[0, 6, 30, 90 minutes, 3 days] (In-Frame (F) and Cumulative (C) features filtered by time T_i and by the significance at 5% level) 84
5.6 List of top features in time-series experiments for different tweet types with corresponding rumor (R) and non-rumor (NR) tags. 86
5.7 The overall list of significant source (src) in-frame (F) and cumulative (C) features from different time-series. 87
5.8 The Performance of the proposed list of features in Rumor identification task 90
5.9 The performance of different classifiers in rumor identification task . . . 90
5.10 The performance of each category of features on rumor and non-rumor cases 90

6.1 The overall statistics of the PHEME data on rumor/nonrumor and rumor veracity tags for each event. The first five events are the largest and relatively balanced data sets of this corpus. 98
6.2 The importance of network features in the rumor veracity task in different time frames. All the time-series features in the table were filtered only to demonstrate the significant features on the basis of 5% significance. Red and gray cells are correlated with false and true, respectively. 103
6.3 The importance of user provenance features in the rumor veracity task in different time frames. 103
6.4 The coefficient of the significant features from the pragmatic category in rumor veracity identification in different time frames of rumor propagation 104
6.5 The importance of different platform specific features in rumor veracity tag in different time frames of rumor propagation 105
6.6 List of the significant categories with the veracity tag in different time frames 105

6.7 List of the significant features in all-time series. The True (T) or False (F) indicates the correlation of the corresponding feature with the rumor veracity tag 107
6.8 Number of the threads and tweets for the task B of the SemEval2019 which was about the rumor veracity identification on Reddit and Twitter. This data is the combination of training, development, and test set after removing the cases which were tagged as 'Removed' by the task organizers. 108
6.9 Number of source (Src) conversations and replies (Rep) on Reddit and Twitter in the training, development, and evaluation sets. 109
6.10 The best performance achieved by the proposed methodology in SemEval 2019 rumor veracity detection task. Per class performance is based on macro averaged F-measure. 111
6.11 Number of threads and tweets for stance classification on Reddit and Twitter. This data is the combination of training, development, and test set. 119
6.12 The result of the conventional and NN machine learning approaches in rumor veracity task 125
6.13 The performance of applying the MTL and attention mechanism on the stance classification task. 126

List of Abbreviations

BOW Bag Of Words

Bi-LSTM Bidirectional Long Short Term Memory

CRF Conditional Random Fields

DF Discussion Forum

GD Genre Detection

LDC Linguistic Data Consortium

LCB Level of Committed Belief

LSTM Long Short Term Memory

MTL Multi Task Learning

NN Neural Network

NW Newswire

ROB Reported Belief

NCB Non Committed Belief

CB Committed Belief

NA Non Attributable Belief

NLP Natural Language Processing

OALD Oxford Advanced Learner's Dictionary

RDC Rumor Detection and Classification

SRDC Single-Step Rumor Detection and Classification

TLV Tweet Latent Vector

TRDC Two-Step Rumor Detection and Classification

Chapter 1: Introduction

Social media platforms continuously generate a massive amount of data. Nowadays, novel breaking news appears first on microblogs like Facebook, Twitter, and Reddit before making it through to traditional media outlets. Hence, microblogging websites are rich sources of information which have been successfully leveraged for the analysis of sociopragmatic phenomena, such as beliefs, opinions, and sentiments in online communication. Among these social media outlets, Twitter emerges as one of the most popular microblogging platforms. According to [Gottfried and Shearer, 2016], more than 59% of Twitter users rely on it as a news reference. In fact, the study found that 11% of the US adult population used Twitter for online news. Twitter serves as one of the foremost media outlets for Natural Language Processing Natural Language Processing (NLP) research, in which practitioners rely on deriving various sets of features to leverage content, user demographics, and network structure. However, the unprecedented existence of this massive data acts as a double-edged sword: we get unreliable information from such sources, and it is a challenge to control the spread of misinformation generated either maliciously or even inadvertently. The information seeker is inundated by an influx of data. The problem is exacerbated when many believe that information online in digital form is inherently accurate and trustworthy, even though it is well known that a lot of information online is false or untrue [Allcott and Gentzkow, 2017]. Factually inaccurate and unverified information or news can harmfully impact an individual or a society. As was observed in the recent US Presidential election, unverified rumors were more likely to be shared by people than verified information [Silverman, 2016], and the influence of disinformation on the election outcome is still under investigation. The influence of rumor could appear in other domains as well. For example, a false rumor that President Obama had been

injured in an explosion in the White House caused stocks to briefly plunge [Domm, 2013].

It is challenging to distinguish reliable information from false information, especially if the data appear to be compelling, correctly formatted, confidently said, and well structured [Wathen and Burkell, 2002] [Rieh and Belkin, 1998].

Despite the popularity of research in addressing online misinformation, most of the current approaches only do well when it comes to long-standing rumors which are known a priori. In this thesis, we study the significant signals for identifying *emerging* rumors. Namely we focus on four significant aspects of online information: *Provenance* (the origin of the content); *Network and propagation*, (temporal and diffusion pattern) [Silverman, 2013]; *Platform* (the meta information that can be extracted from the specific social media platform); and *Content* (linguistic and pragmatic information that can be extracted from the context of the posts). Based on our analysis, each of these four major signals plays a crucial role in addressing rumor identification and verification.

We are in particular interested in new pragmatic features extracted from the posts' content. One of the main contributions of this work is studying the impact of novel pragmatic features such as Level of Committed Belief (LCB). We propose an automated pipeline to measure LCB in post content in the rumor conversations. Capturing the belief from the context is a challenging task but could help us go beyond the propositional meaning and understand the cognitive stance of the writer about a certain topic. Speaking/writing with Confidence– also referred to as firmness or strength of belief [Bandura and Wessels, 1994]– could cause a specific story go viral in social media. In contrast, hedging, skepticism, and low certainty, repel the audience regardless of its accuracy. People like to hear confident statements; as former President Bill Clinton once said, Americans would initially prefer someone strong and wrong over someone weak and right.

Problem Definition:

The main objective of this thesis is investigating rumor detection and verification in social media, and how rumor emerges and propagates. To that end, we leverage novel supervised and unsupervised learning within conventional and advanced machine learning technologies. We explicit modeling of linguistic modality, committed belief, and genres coupled with network propagation, social media specific characteristics in addition to pragmatic and content features as devices.

1.1 Terminology

In this dissertation, we will discuss the various problems in different environments. As such, some of the common terms must be defined. These terms are often used interchangeably, but they differ greatly in function and content. Here, we have put together a glossary using the definitions from the *Oxford Advanced Learner's Dictionary (OALD)* with an explanation to bring some clarity to confusing related terms.

- **Misinformation**

  Misinformation is false or inaccurate information; it could deliberately intend to deceive or could include disinformation to deliberate propaganda with unintentional errors.

- **Disinformation**

  Disinformation is false information which is intended to mislead, especially when issued by a government organization to a rival power or the media. This differs from misinformation, for disinformation is always distributed intentionally and, quite often, strategically. The English word is a translation of the Russian, dezinformatsiya, which is derived from the title of a KGB propaganda department.

- **Rumor** The ground truth definition of a rumor is a statement whose true value is uncertified even if its ideological or partisan origins and intents are clear. Any rumor could remain unverified for a long time and become a fact or false information after being verified by an eligible source. A rumor may spread misinformation, or disinformation about an event or entity. Despite the similarity, the inference of the word rumor should be differentiated from disinformation, misinformation, false information or any other terms which imply more falsehood and intention than uncertainty.

1.2 Twitter

We treat tweets as an object. Each tweet could be classified in one of the following categories.

- **Tweet**

  The general tweet is a message containing text, and/or a photo, and/or a video, where it appears on the sender's profile page and on the main timeline.

- **Re-Tweet**

  A Retweet is a re-posting of another tweet and a quick way to share the Tweet with all the followers of the user. Sometimes people on Twitter type "RT" or "RT USER NAME" at the beginning of a post to show that they are re-posting someone else's content.

- **Reply**

  A reply is when a user responds to another person's Tweet on the sender's profile page.

- **Mentions**

  A Tweet containing another account's Twitter username, using the "@" symbol.

For example: “I agree with @USER". Anyone on Twitter who follows the sender of a mention will see the post in their timeline.

1.3 Contributions

This thesis contributes the following to the field of social media and computational linguistics in online discourse:

- Contributes Rumor Detection and Classification (RDC) as a novel machine learning pipeline for the rumor tracking task and investigates the importance of content features on rumor tracking and opinion classification [Hamidian and Diab, 2015];

- Investigates the application of TLV instead of the conventional approaches like TF-IDF and Bag of Word in a two-way Rumor Retrieval (RR) task. This component uses the latent representation of the tweet and leads to state-of-the-art performance in the rumor retrieval task;

- Contributes a novel dataset which was collected, pre-processed, and annotated with more than 10,000 rumors from 2014 to 2016. The tweet ID and rumor tag are available for research use [Hamidian and Diab, 2016];

- Introduces the first NN Multi-Genre sequence tagging LCB model. This tagger has since generated great research interest in applying it to other tasks, such as rumor identification, stance classification, and rumor veracity (See Chapter 4)[Rambow et al., 2016];

- Contributes a novel domain-independent approach for identification of unseen emerging rumors. The proposed methodology differs significantly from conventional methods which are only applicable to identify the unfolded longstanding

rumors which are known a priori. The methodology is based on leveraging linguistic and meta-linguistic signals from the content, source, and network propagation information in rumor conversation (See Chapter 5). [Sardar Hamidian, 2019b].

- Investigates the rumor veracity task by studying the interpretability of contextual and non-contextual signals in rumor conversations (See Chapter 6). The thesis also compares the results obtained by several conventional and neural network machine learning systems. The proposed models in this work outperform current state-of-the-art performance in rumor veracity classification in SemEval 2019 [Sardar Hamidian, 2019a].

- Provides a novel neural network approach by applying a multi-task learning framework to attain the stance orientation of the people in rumor conversation. The stance signal could significantly boost the overall performance of identifying the unverified type of rumor in the related rumor veracity tasks.(See chapter Section 6.4.5) [Sardar Hamidian, 2019a].

1.4 Thesis Layout and Brief Overview of Chapters

This thesis comprises 7 chapters which will be briefly described as follows. Chapter 2 studies background related work about rumor in different fields and the history of closely related articles on identification, propagation, and verification of rumor on social media. In Chapter 3, we lay out our rumor detection and opinion classification approach, including supervised machine learning frameworks and analysis of the use of several feature sets. Chapter 4 of this thesis includes the multigenre LCB tagging pipeline on the discussion forum and newswire data. In Chapter 5, the problem of rumor identification and propagation is explicitly discussed for novel unseen rumors at different time frames during their spread. In Chapter 6, we investigate the rumor

veracity problem on different social media platforms and propose an automated domain independent pipeline for identifying the veracity of rumor conversations on social media. We also introduce a new pipeline to classify stance orientation of the replied tweets in the rumor conversations. Chapter 7 concludes the thesis with all the observations and lessons learned. We will have a close look at the limitations of the proposed contributions in the different sections of this work. Finally, we will conclude with suggested directions for future work.

Chapter 2: Background

2.1 Introduction

The primary objective of this thesis is to investigate the rumor phenomena in different social media platforms. We are looking for certain natural language pragmatics in addition to typical meta-linguistic attributes like network propagation and content-based features associated with rumor. Our ultimate goal is to build a predictive pipeline to identify existing as well as emerging rumors on social media. Related works to this thesis have studied numerous aspects of identification, propagation, and credibility of online information mostly by relying on network, user-specific as well as content-based features. For further insight, We also overview different fields, including psychology, journalism, and philosophy, which studied the phenomenon of rumor even before the emergence of social media platforms.

The overview of this chapter is as follow, we first briefly talk about some early studies by different psychologists and then mainly focus on closely related works from the field of NLP that concentrate on information propagation and trustworthiness on different social media platforms, and especially on Twitter. In the final section of this chapter, we also summarize and compare the publicly available corpora in this domain. We delve into the approaches applied for collection, annotation, and development along with summaries and representation of key features of each corpus.

2.2 Rumor in Psychology

Psychologists have studied the phenomenon of rumor from various angles. First studies were carried out in 1902 by German psychologist and philosopher, William Stern, and later in 1947 by his student Gordon Allport, who studied how stories get affected in their lifecycle [Takahashi and Igata, 2012]. In 1994, Robert Knapp

published “A Psychology of Rumors”, which consists of a collection of more than a thousand rumors that were propagated during the World War II and that was the first time list of word of mouth rumor became available. According to his work, the rumor is what was transmitted by word of mouth and bore information about a person, an event, or a condition, which fulfilled the emotional desires of the public [Knapp, 1944]. In 1948, Allport and Postman [Allport and Postman, 1947] introduced a new angle in studying the rumor and studied how one rumor reflects leveling, sharpening, and assimilation behavior in its propagation. A more recent study by Nicholas [DiFonzo and Bordia, 2007] assessed how rumors start and spread and can be managed across media outlets and within organizations. According to his book, a rumor could attract attention, induce emotion, incite involvement, and affect attitudes and actions. He also named fact-finding, relationship-enhancement, and self-enhancement as the three psychological motivations for rumor transmission. Related studies in political communication conducted by Harsin Harsin et al. [2006a] presented the idea of the "Rumor Bomb." For Harsin, a "Rumor Bomb" spreads the notion of the rumor into a political communication concept. In other research, (Kumar and Geethakumari) [Kumar and Geethakumari, 2014] examine the use of theories in cognitive psychology to estimate the propagation of misinformation and propaganda across social networks.

2.3 Rumor on Social media

There are several studies about the behavior of misinformation on social media, how they are distinguished, and how social media users react to them. Most of the studies on misinformation study the phenomena of rumor on Twitter, Facebook, Reddit, and Sina (China’s most popular social media platform). Twitter is the most popular one since it has an infrastructure which allows researchers to be able to access the data, network information, and meta information of all the users through Twitter APIs.

Moreover, a large number of monthly active users and unlimited access to the data, have made Twitter even more popular [Ozturk et al.]. Based on the applied approach and problem definition, the related works are divided into three main sections. In the first section, the related works which contribute the new corpora for the task of rumor identification and veracity classification are discussed. In the second section, we discuss the related work on rumor identification, and in the last section, the rumor veracity related works are discussed. In the last two sections, the related works are arranged based on the applied approach in each work.

2.3.1 Rumor Related Corpora

The main challenge to detecting the misinformation lies not in building the algorithm itself, but in finding the right data with which to train that algorithm. Despite the increasing number of studies on addressing misinformation and propagation of unverified information on social media, research in publicly available corpora is scarce. In this section, we set out to collect, summarize, and compare the publicly available corpora on misinformation, rumor, and factuality. We delve into the approaches applied for collection, annotation, and development along with summaries and representation of key features of each corpus.

2.3.1.1 LIAR

LIAR [Wang, 2017] is one of the earliest publicly available datasets for fake news detection related tasks[1]. It includes 12.8 K manually labeled short statements from the popular fact-checking website, PolitiFact. The dataset is split into three sets, including train, test, and validation sets. Instead of simply labeling each statement as either true or false, there is a range from true to pants-fire (barely-true, half-true, mostly-true, and false) from which the labels are chosen.

[1]https://www.cs.ucsb.edu/~william/ data/liar_dataset.zip

2.3.1.2 FakeNewsNet

FakeNewsNets[2] is one of the recent publicly available datasets for fake news detection[Shu et al., 2017]. Like most of the datasets, ground truth labels of news articles are extracted from two well-known fact-checking websites, BuzzFeed and PolitiFact. One feature that makes FakeNewsNet a little more interesting than the other datasets is the social engagement information it has. Even though the information of users is totally anonymous, each news article comes with the number of users who share the news articles and their corresponding followers. Such a feature makes this dataset potentially useful for those who want to not only train a model for detecting fake news but to study how misinformation spreads in social media. This dataset is part of an ongoing project and new news articles are being added to this data resource.

2.3.1.3 Ma

This dataset [Ma et al., 2016] is one of the largest publicly available corpora for rumor on Twitter and Weibo. The author relied on www.snopes.com as the ground truth and obtaining the relevant keywords for the data collection on Twitter. More than 778 events are obtained of which 64% were a rumor, In addition, other publicly available datasets were added to balance the number of rumor and non-rumor classes to 498 and 494 respectively. The tweet ID, event ID, and associated rumor tag are included in the dataset. The author has not put the tweet content nor type of event in this dataset because of the Twitter policy, but the majority of tweets are retrievable using the tweet ID. In a similar approach for Weibo data, the author relied on Sina community management center as the ground truth to collect the set of verified misinformation. The resulting corpus consists of 2313 rumor and 2351 non-rumors.

[2]https://github.com/KaiDMML/FakeNewsNet

2.3.1.4 MediaEval

The aim of collecting this corpus [3] was to study the challenges of computation verification of online content as fake or real. The author created the first corpus of tweets, including the images and tweet content related to the Boston Bombing and Hurricane Sandy storm. Later This dataset was extended to 16 events with more than 16,000 tweets. The online version of the corpus does not contain the tweet content but Tweet IDs, which can be used to retrieve the tweet information. [Boididou et al., 2014] Boididou et al. relied on both images and keyword to collect the relevant tweets with an image of the two events. The author did not explicitly name what online resources were used for collecting the fake or real images nor the intuition behind the used keywords. One of the main drawbacks of only relying on keywords is that the relevant tweets which do not contain those keywords are missing. Most of the annotation pipeline is automated; however, the irrelevant tweets which were only 5% of the total tweets in the first corpus were checked and removed manually. The author did not specify the manual process in data annotation pipeline.

2.3.1.5 Rumor-Citation (Kaggle)

This dataset[4] is a collection of posts and claims from three popular fact-checking websites including Snopes, Emergent, and Politifact. Each instant of this dataset comes with a link to the data source, claim, and many other features including the claim-label which indicates the degree to which a claim is True or Fact. Despite some similarities in the way these websites classify online information, heterogeneous schemes among three resources make it less uniform in comparison with other available datasets for rumor and misinformation.

[3] https://github.com/MKLab-ITI/image-verification-corpus/tree/master/mediaeval2015
[4] https://www.kaggle.com/arminehn/rumor-citation

2.3.1.6 PHEME

There are a few benchmark datasets on misinformation, hoaxes, and rumor that have been proposed by PHEME in recent years. We review the PHEME rumor relevant datasets that have been widely used in rumor detection tasks. It is worth mentioning that despite the commonality of the tweets and events in different PHEME datasets the annotations vary by the use case.

2.3.1.7 PHEME rumor scheme dataset (PHEME-RumorScheme)

This dataset contains 330 rumor threads in two languages (297 in English, and 33 in German) including the 4,842 tweets of 9 newsworthy events [Zubiaga et al., 2016b]. All the tweets have been annotated for support, certainty, and evidentiality. The general approach in collecting the dataset is set to find the emerging rumor rather than the conventional method of collecting the widespread rumors known in a priori. To collect the unknown rumors, the newsworthy events had been selected by the journalist that could potentially spark the propagation of rumors. Then the candidate conversational rumor with corresponding conversations collected for annotation. The general information of the corpus is tabulated in Table 2.1 and Table 2.2

2.3.1.8 PHEME dataset for Rumour Detection and Veracity Classification (RSV)

This dataset[5] is an extension to the first PHEME datasets of rumor containing tweets from nine events [Kochkina et al., 2018]. Three levels of annotation exist in this dataset, including the stance, rumor tag, and veracity. However the stance annotation only available on a subset of threads which were described in RumorEval dataset [Zubiaga et al., 2016a]. As opposed to most of the available studies which rely on fact-checking resources or non-expert annotators, this dataset is labeled as true, false,

[5]https://figshare.com/articles/PHEME_dataset_of_rumours_and_nonrumours/4010619

Dataset	Data source	Date	Lang	Size	Dataset type
MediaEval	Twitter	2015*	En	14000 (Tw, Img)	FN
Qazvinian	Twitter	2011*	En	8900 (Tw)	Ru
Hamidian	Twitter	2016*	En	7500 (Tw)	Ru
Ma	Twitter Weibo	2016	En	1.1 M (Tw) 3.8 M (We)	Ru
Kaggle	Snopes Emergnet Politifact	2017	En	21,930 Claims	RU+FN
RumorScheme (PHEME)	Twitter	2016	En DE	4,842 Tweet	RU
RumorEval (PHEME)	Twitter	2016	EN	5,802 Tweet	RU
RSV (PHEME)	Twitter	2018	EN	105,354 Tweets	RU
LIAR	PolitiFact	2017	EN	12,800 Statements	FN
FakeNewsNet	BuzzFeed PolitiFact	2017	EN	422 News articles	FN

Table 2.1: List of publicly available Rumor (RU) and Fake News (FN) dataset with their stats

Dataset	Annotation	Annotation categories
MediaEval	Fact-checking websites	{real "fake"}
Ma	www.Snopes.com	{rumor "not-rumor"}
Qazvinian	Manual	{Rumor "Not-RumorËndorse"Deny"neutralündetermined"}
Hamidian	Auto	{Rumor "Not-RumorËndorse"Deny"neutralündetermined"}
Kaggle	Snop.com emergnet.com politifact.com	{Unverified "False"True"}
RumorScheme	Manual	{rumour"non-rumour"}{rumour-type"}{ertainty}{tance}
RumorEval	Manual	{rumour"non-rumour"}
RSV	Manual	{rumour"non-rumour"}{rumour-type"}
FakeNewsNet	Fact-checking websites	{Fake "Real"}

Table 2.2: The annotation information of the publicly available corpora.

and unverified by professional journalists. To the best of our knowledge, this dataset is the only publicly available dataset which contains all the classes mentioned above in a single corpus.

2.3.1.9 Discussion

We discussed the annotation schemes and applied methodologies behind the publicly available copra in this domain. To the best of our knowledge, the datasets mentioned above are the only publicly available resources which are completely accessible for the researchers to use. It is worth noting that might exist other resources that are

available upon request, but we only discussed those which are publicly and freely available. We compare the available resources in this domain from different angles and compare these resource by three main features. First, the size of the corpus, two, the annotation quality, and three, the existence of metadata. One of the key methodologies which are used in most of the available corpora to date is reliance on publicly available fact-checking resources. However, there are also some studies which proved the existence of different types of bias in these resources. Considering the datasets that are mentioned above, the PHEME and FakeNewsNet copra are the best options to pick for the researchers in this domain. The main features which make these corpora unique are the quality of annotation and the availability of the meta information.

2.3.2 Rumor Tracking and Identification

In this section, we mainly focus on machine learning approaches which leveraged standard corpus and investigated the rumor phenomena from the following perspectives. First, user information or provenance, second, network or propagation, and the last one is the linguistic content. We divide the related work of rumor identification and rumor veracity and discuss them separately. The associated work for each section is also separated based on the used features in the proposed approach. After explaining the related works in each section, we finalized the section with a discussion about each methodology.

- **Rumor has it: Identifying Misinformation in Microblogs** [Qazvinian et al., 2011a]

Method Description:

One of the most relevant works and pioneers in studying the misinformation online,

which addresses rumor detection on Twitter using content-based as well as microblog-specific meme features. This work addressing the rumor identification task on long-standing rumors which are known a priori. The author relied on a supervised machine learning approach with mainly the content based features and a standard dataset which consists of five famous rumors like Obama is a Muslim and Sarah Palin got divorced.

Evaluation:

[Qazvinian et al., 2011a] suggests label-dependent features in creating their User-based (USR) and URL features, which is only possible by having the input data labeled for being a rumor or not. In other words, labeled data is used for creating the language model (LM) with USR and URL features, and the trained LM is then used for extracting the value of each feature. The proposed model in this work showing the promising performance for rumor tracking of long-standing rumor but perform poorly in the novel emerging rumor. One of the shortcomings of the provided dataset in this work is the missing meta information of the data and missing information from the dataset.

- **Predicting information credibility in time-sensitive social media** [Castillo et al., 2013]

Method Description:

In a similar approach but with explicit analysis, [Castillo et al., 2013] leveraged both information cascade and content features of the tweets by applying a supervised mechanism to identify credible and newsworthy content. Castillo conducted a thorough analysis of information propagation behavior using the re-tweets collected one hour after the earthquake of Chili. According to his work "confirmed truth" or the rumors which are verified as truth after the tweet burst are more likely to be affirmed in re-tweets than false rumors which had a relatively high number of tweets in the "question" category. Castillo designed an experiment to identify newsworthy topics automatically.

Three-way classification designed in this phase including NEWS/CHAT/UNSURE but the best result achieved by removing the UNSURE from the experiment. Sixty-eight features are aggregated (many of which are based on previous works) in four categories including message-based, topic-based, user-based, and propagation based features. The results show that the text-only feature category leads to the highest performance in the newsworthy classification task. Noticeably the text-only category got the same result as the full feature category classifier. The performance of different learning schemes also experimented in which Bayes Network-based and Random forest performed better than Naive Bayes and Logistic Regression. Among the proposed features "User with the high number of followers and post," "URL to the site with a description" is correlated with credibility and on the other hand " negative sentiment," "question and Exclamation marks" have a significant correlation with Non-credible cases. The author extended the proposed work to Spanish and studied the same features from the newsworthy and credibility tasks. In this comparison, the credibility features remained as significant as for English experiments but some newsworthy features differ from the English newsworthy tests.

Evaluation:

This work extends the previous studies and proposes a new set of features and learning scheme for the newsworthy and credibility analysis on twitter however the assessment was only limited to a single event. The fair evaluation of the proposed methodology could be assessed in other events and also other social media platforms.

- **Analysing How People Orient to and Spread Rumours in Social Media by Looking at Conversational Threads** [Zubiaga et al., 2016b]

Method Description:

This work introduces a new scheme for annotating the social media data collected from 330 threads discussing 9 different newsworthy events. The topics are selected by journalist and the annotation done by a combination of experts and non-experts.

The analyses assessed the rumor diffusion on social media with support and denial of the recipient in different time frames. According to their findings true news getting resolved faster than false news on social media also they show the social media users have more tendency to support rumor whose veracity is yet to be verified. According to this study, social media users do reasonably good in distinguishing true from false after they are debunked or corroborated. One of the findings of this work is that true rumors resolved significantly faster the false rumors in twitter as 2 hours for true and 14 hours for the false. Another finding of this paper is that the tweets that are tweeted the least are those denying the rumor regardless of being true or false.

Evaluation:

This work investigates the reaction to a different type of rumor in different time series of their propagation but nothing specified about the significance of this information and a degree to which they could be helpful address the problem of rumor on social media. The author also did not mention anything about controlling for confounding attributes in their experiments. We assume that some of the findings of this work could be challenged by more analysis.

- **Rumor Detection over Varying Time window** [Kwon et al., 2017]

  Method Description:

This is one of the recent works, which discusses the rumor phenomena in various time frames of propagation and investigate the effect of linguistic, network and temporal features in rumor phenomena. Their findings suggest that people with moderate influence are more likely than people with high influence to attend in rumor conversation. they consider influencer as people with a high number of follower, friends, and tweets and notice that the user feature of non-rumor event in majority percentiles are more than rumors. Which suggest the used features can be used to determine rumor and not rumor events. In Linguistic feature category [Kwon et al., 2017] used different categories and subcategories to capture psychological features and negation

from the content of the tweets such as writing pattern, negation, the psychological process including (social, affect, cognitive, perceptual, tentative, swearing) and writing pattern like negation, home, tense. According to this study in the initial stage, the rumor could be distinguished by linguist features like negation and skepticism. Linguistic feature lead to highest performance in this study by achieving 0.8-0.86 f-measure. As opposed to a network feature, which required a longer time to become predictive. Only the features representing the specific direction of information flow, unique users whose source is outside the network and the rumored content being neglected are important for classification.

Evaluation:

The main contribution of this work is the thorough analysis of the different category of features to identify rumor in different propagation phases. They suggest that linguistic features are promising at the early stage of rumor propagation as opposed to network and temporal features that distinguish rumor from non-rumor in a long-term window. The author did not explicitly talk about the rumor and not-rumor rumor events and if the proposed list of features could distinguish rumor and non-rumor on the events with similar entities or topics. It might be the case that event with specific topics included a certain range of linguistic features that could down perform in other topics.

- **Identifying rumors and their sources in social networks**[Seo et al., 2012a]

  Method Description:

This work monitors the social network as a connected graph to identify rumors and their corresponding sources by observing which of the monitoring nodes (individuals who are assigned and agreed to report if they received the information) receive the given information and which do not. In this approach, the monitoring nodes which are exposed to the rumor are in the M+ category, and those who are not exposed are in M-. A member of M+ node with the highest number of connection to other monitoring nodes in M+, shortest total distance to other M+ nodes, and longest

distance to M- members are selected as the source of rumors.

Evaluation:

The main advantage of this approach is being context-independent to identify the rumor resources, but considering the massive social media networks number of the candidate, sources could raise and assigning the adequate number of monitoring nodes would be expensive.

2.3.2.1 Discussion

We selected the most insightful works for this section, which have inspired many other studies in this domain. There are also a significant number of related works which we will be briefly discussed in a different section of this thesis (See Chapter 5 and Chapter 6). Most of the related works in rumor tracking domain [Qazvinian et al., 2011a, Hamidian and Diab, 2015, 2016] relying on the content based features by levering the standard dataset and focusing on the dense representation of the documents. Supervised approaches performed significantly better than unsupervised ones, and the extracted signals from the content are more important than the network or event the source information. Only leveraging the content features lead to a state of art performance in the rumor tracking task. Considering the recent improvements of dense representations in the sentence level, and the promising performance of the neural network approach in text classification there is massive trend in applying NN for rumor tracking tasks and most of the recent work in this domain could outperform the old conventional machine learnings in rumor tracking baselines [Ma et al., 2016, Zubiaga et al., 2018, Cao et al., 2018].

2.3.3 Rumor Veracity

In this section, we review the most insightful works for the rumor veracity task, which have inspired many other studies in this domain. In the discussion section, the

proposed methodology in each work is evaluated, and the section is concluded with a summary of the proposed approaches from other related works for identifying the veracity of rumor or fake news analysis on social media.

- **Automatic detection and verification of rumors on Twitter** [Vosoughi, 2015a]

Method Description:

[Vosoughi, 2015a] proposes his two-step rumor detection and verification model on the Boston Marathon bombing tweets. The Hierarchical-clustering model is applied for rumor detection, and after the feature engineering process, which contains linguistic, user identity, and metalinguistic features, he adopts the Hidden Markov model to find the veracity of each rumor. Soroush also analyses the sentiment classification of tweets using the contextual information, which shows how tweets in different spatial, temporal, and authorial contexts have, on average, different sentiments.

Evaluation:

This work is one of a few publicly available dissertations in rumor identification and veracity, which is an excellent source for researchers in this domain. The main advantage of the proposed methodologies in this thesis is that author did a great job investigating the impact of different category of linguistic and meta-linguistic signals in addition to applying an unsupervised methodology for rumor identification task. It is worth noting that there are only a few works available which address rumor identification problem applying unsupervised methodologies. The main shortcoming of this work is that the author did not experiment with the strength of the proposed mechanism on early identification of rumor.

- **The spread of true and false news online** [Vosoughi et al., 2018]

Method Description:

This work is one of the most recent studies in this field. The main goal of this work

is to compare the differences in the spread of true and false news on social media. Different statistics have been measured in this study. Overall the statistics in this work show that false information is more likely to be viral in information cascades than true information, and also spread faster, farther, more in-depth and also more broadly than the true information for all the categories of information. In Political cascades, both true and false information tends to be more viral, and also spread faster and deeper. Furthermore, news about politics, urban legends, and entertainment spread to most people, while rumors about politics and urban legends spread the fastest and are the most viral. In this work, they have also compared different users regarding the rumor and not rumor. Five different features have been compared in their experiment, including the number of followers, valid profile, User engagement, number of, and account age. In one of the interesting observations, no difference between the gender distribution of false and true rumor. According to the results in this work, a relatively small percentage of users are responsible to start a large number of rumors. According to his work, false stories inspired fear, disgust, and surprise in replies, true stories inspired anticipation, sadness, joy, and trust.

Evaluation:

The main advantage of this work is comprehensive support and statistical analysis on a massive dataset. However, they relied on the fact-checking resource as ground through without considering the fact the there is huge bias, especially when it comes to the political domain, in the fact-checking resource. Another factor that could be more covered in this work is the impact of different type of tweets in their evaluation for example retweets, replies or mentions in the true and false discourse. Nothing specified about controlling for the type of tweets in this work.

- **All-in-one: Multi-task Learning for Rumour Verification**[Zubiaga et al., 2018]

Method Description:

This work proposes 4 steps including (1) rumor detection, determining whether a claim is worth verifying rather than the expression of an opinion; (2) rumor tracking, collecting sources and opinions on a rumour as it unfolds; (3) stance classification, determining the attitude of the sources or users towards the truthfulness of the rumour, and (4) rumour verification, as the ultimate step where the veracity value of the rumor is predicted. These steps are discussed in different steps of the rumor life cycle. In comparison to other tasks identifying the rumor, veracity is the relatively an important task and the ultimate goal of the proposed framework. Among the four steps of rumor resolution, three of them are the classification tasks. Rumor detection is the task which distinguishes rumors (unverified pieces of information) from non-rumors (all other circulating information). This work applies a sequential and MTL approach. In the sequential approach, each thread or conversation decided into linear branches.

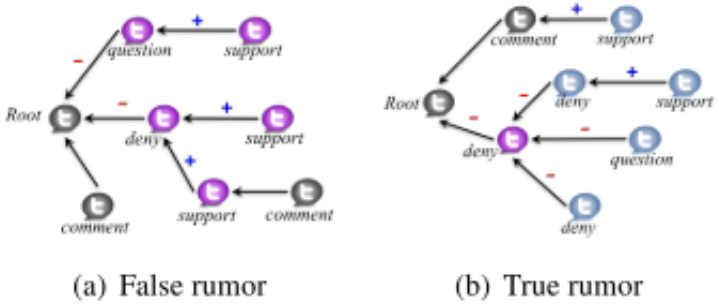
Evaluation:

Applying the MLT for three main tasks of rumor including the rumor tracking, detection, and veracity classification in addition to stance classification was done for the first time in this work. The author considered the main and objective of this task as the rumor veracity and noting have been specified about the impact of the proposed methodology on the other auxiliary tasks. One of the shortcomings of this work is that the proposed architecture could have been clarified in more details since the current documentation makes the replication of this work impossible. However, the application of MTL for the mentioned tasks above is so insightful as many studies start to apply MTL for rumor tasks after this work published.[Ma et al., 2018a, LIAO et al., 2018].

- **Rumor Detection on Twitter with Tree-structured Recursive Neural Networks** [Ma et al., 2018b]

Method Description:

This work presents a neural rumor detection approach based on recursive neural



(a) False rumor (b) True rumor

Figure 2.1: Tree-structured Recursive Neural Networks for identifying the veracity of the source

networks (RvNN) to bridge the content semantics and propagation clues. The proposed model has been used mostly on semantic and syntactic parsing. Each node of the structure represents the semantic of a post. The propagation tree at Figure 2.1 rooted from a source of the conversation rather than the parse tree of a single sentence, and each tree node representing a post in the conversation instead of individual words. This work indicates that Twitter has self-correct behaviors as if someone posts a piece of wrong information and gets deny in response the following tweets will support the denial as its shown from the following figure pot in the paper.

Evaluation:

One of the key insight and strength of this work is the proposed mythology for modeling the network and codding the structural information from underlying posts in the conversations. Considering the fact the user information are key signals in the rumor veracity integrating the extra information such as user properties to the proposed model in this work could enhance representation learning and detect rumor spreaders at the same time.

2.3.3.1 Discussion

Identifying the rumor veracity is more challenging than the rumor identification task. One of the main reasons is the novelty of the stories and patterns that a new

rumor show when they emerge. Most of the current approaches are looking for the generalizable signals both from the content and conversation of rumor to address this problem. From the content, [Vosoughi, 2015a] relied on the new pragmatics which can be extracted from the content of the posts, including the speech act events. From the network, [Vosoughi et al., 2018] proposed the novelty and also investigated the profile information. Another direction is to extract auxiliary information from the conversation, which provides an extra hint for identifying the veracity of the source of conversation. [Zubiaga et al., 2018, Ma et al., 2018b] are among this category. Overall our observation of the relevant work in rumor veracity tasks showing that the combination of pragmatic in addition to conversation feature is key to obtain some insight about the veracity of the source of the rumor.

2.4 Summary

In this chapter, we selected some of the pioneer and insightful works in the filed of rumor tracking, identification, and veracity classification. Despite the relevance of these problems, there is an enormous difference in the concept and approaches that can be applied for each task. Most of the current approaches on rumor tracking and identification rely on content-based signals, besides, to advance machine learning approach while most of the approaches on rumor veracity are more toward learning a new insight and adding more generalizable signals which can signify the veracity of rumor and also be applicable when it comes to unseen events. One of the lessons that we learned from the related work is the importance of applying the natural language inference (NLI) and also lexicon semantics including the sentiment and subjectivity for both tasks and specifically the rumor veracity problem. There is still a gap for improvement and investigating both problems that we are trying to address in the thesis.

Chapter 3: Rumor Detection and Opinion Classification on Twitter

3.1 Rumor Detection and Opinion Classification (RDC)

In this chapter, we present two main pipelines for rumor identification and classification tasks on Twitter. In both pipelines, we trained the supervised classification models by exploiting the standard dataset. For the first task, the novel set of features is proposed, and their impact is explicitly discussed. In the second pipeline instead of the BOW and the proposed list of features, the TLV as the representation of tweet was applied for rumor identification task. The proposed models could outperform the baseline and the current state-of-the-art model in rumor detection and classification tasks on the same dataset. The TLV model could reach better performance with higher precision and recall in the rumor detection task. Whereas, applying the RDC with three categories of features performed better on the rumor classification task. All the experiments were conducted with various levels of preprocessing and machine learning models as a precursor to the classification as well as grouping of features. Figure 3.1 illustrates the RDC pipeline with corresponding classification labels.

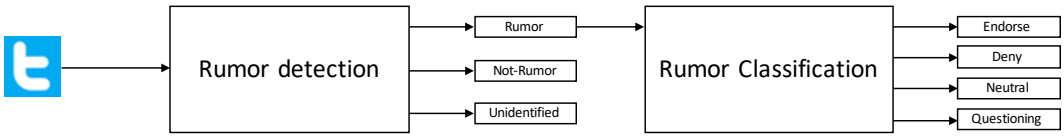


Figure 3.1: The overall pipeline for RDC

3.2 Problem Definition

Considering that we have a set of related tweets that are about the same topics (Mostly controversial topics). Our objective can be categorized in two-fold: (1) Identifying the

controversial tweet which mostly spread false information (Rumor Detection) (2) and classifying the identified rumor's opinion regarding that rumor. (Rumor Classification) Following two tweets are both related to Obama and Islam but only the first one is about the "Obama is muslim" rumor.

- (Rumor) Obama fooled many, but not all!!! Obama a radical Islam!! Lover of farraconn, kill white people!! Lover of alynsky student of corruption, destroyer of America!! Favored our enemies, Iran, Cuba, Muslim brotherhood, Soros, guns/drugs in our country, uranium to
- (Not-Rumor) He wrote an article saying Hillary and Obama perfer Islam to Christianity. WTF are you talking about?

After identifying the rumors we are looking for those opinions that endorse, deny, questioning or are neutral about that rumor. We call this task rumor classification as we classify the opinion of the writer regarding the certain rumor. For example, following four examples are instances of each category about Obama is muslim rumor. All the instances are about the same rumor but showing different opinion regarding the controversial topic.

- (Rumor)(Endores) Check this video out – Obama Admits He Is A Muslim http://t.co/sTnWGAn via youtube
- (Rumor)(Denies) glennbeck so what are you saying? Liberals don't believe in God? So Obama is a racist Muslim and half the population is God-less? #turd"
- (Rumor)(Questioning) Why Obama is not a Muslim http://nblo.gs/8pKGx
- (Rumor)(Neutral)TV drove viewers to Internet to explore Obama-Muslim rumours

3.3 Dataset

Data collection process done by people at University of Michigan (Vahed et.al, (2011)) [Qazvinian et al., 2011a] (V11) using the public Twitter data API for five different 'established' rumors as listed in Table 3.1. The handcrafted regex and keywords were utilized to retrieve the largest set of tweets related to the rumors. Table 3.1 shows a list of the utilized regex for tweet collection process with corresponding #Tweets with the veracity status. The general annotation guidelines are presented in Table 3.2.

Rumor Reference	**Regular Expression Query**	**Status**	**# of tweets**
Is Barack Obama muslim?	Obama & (muslim | islam)	False	4975
Michelle Obama hired many staff?	staff & (michelle obama)	Partly True	299
Cell phone numbers going public?	(cell|cellphone|cell phone)	Mostly False	215
Sarah Palin getting divorced?	palin & divorce	False	4423
Air France mid-air crash photos?	(air.france|air france)	False	505

Table 3.1: List of rumor examples and their corresponding queries used to collect data from Twitter [Qazvinian et al., 2011a]

0	If the tweet is not about the rumor
11	If the tweet endorses the rumor
12	If the tweet denies the rumor
13	If the tweet questions the rumor
14	If the tweet is neutral
2	If the annotator is undetermined

Table 3.2: Rumor Detection Annotation Guidelines

The following examples illustrate each of the annotation labels from the Obama rumor collection.

0: 2010-09-24 15:12:32 , nina1236 , Obama: Muslims 2019 Right To Build A Manhattan Mosque: While celebrating Ramadan with Muslims at the White House, Presi... http://bit.ly/c0J2aI

11: 2010-09-28 18:36:47 , Phanti , RT @IPlantSeeds: Obama Admits He Is A Muslim http://post.ly/10Sf7 - I thought he did that before he was elected.

12: 2010-10-01 05:00:28 , secksaddict , barack obama was raised a christian he attended a church with jeremiah wright yet people still beleive hes a muslim

13: 2010-10-09 06:54:18 , affiliateforce1 , Obama, Muslim Or Christian? http://goo.gl/fb/GJtsJ

14: 2010-09-28 22:22:40 , OTOOLEFAN , @JoeNBC The more Obama says he's a Christian, the more right wingers will say he's a Muslim."

2: 2010-10-05 17:37:04 , zolqarnain , Peaceful Islam- Muslims Burn CHURCH in Serbia: http://wp.me/p121oH-1ir OBAMA SILENT #politics #AACONS #acon #alvedaking #women #news #tcot

Table 3.3 shows statistics for the annotated tweets corresponding to each of the five rumors. The original dataset as obtained from [Qazvinian et al., 2011a] did not contain the actual tweets for both Obama and Cellphone rumors, but they only contained the tweet IDs. Hence, we used the Twitter API for downloading the specific tweets using the tweet ID. Accordingly, the size of our dataset is different from that of [Qazvinian et al., 2011a] amounting to 9000 tweets in total for our experimentation.

Rumor	**0**	**11**	**12**	**13**	**14**	**2**	**Total**
Obama	945	689	410	160	224	1232	3666
Michelle	83	191	24	1	0	0	299
Palin	86	1709	1895	639	94	0	4423
Cellphone	92	65	3	3	3	0	166
Air France	306	71	114	14	0	0	505
Mix	1512	2725	2452	817	321	1232	9059

Table 3.3: Final list of tweets that could be extracted using Twitter API

3.4 Experimental Conditions

We approached RDC in a supervised manner and investigated the effectiveness of multi-step classification with various sets of features and preprocessing tasks versus a single step detection and classification approach. In the single-step classification for RDC, we performed detection and classification simultaneously as a 6-way classification task among the six classes in the labeled data, as shown in Table 3.2, by retrieving the tweets as Not Rumor(0), Endorses Rumor(11), Denies Rumor(12), Questions Rumor(13), Neutral(14), and Undetermined tweets(2). In the two-step classification set up, an initial 3-way classification task is performed among the following groups of fine-grained labels (0, Not Rumor), (2, Undetermined tweet), and the compound (11-14, Rumor) labels. This is followed by a 4-way classification step for the singleton labels, (11, Endorsing the Rumor), (12 Denys the Rumor), (13, Questions the Rumor), and (14, Neutral about the Rumor). In the second step, we took out class 0 and 2 tweets from the training dataset and only classified the tweets from the test dataset, which had been classified as a rumor in the first step. The underlying motivation of our effort in designing the single-step and two-step classification is to investigate the performance of each technique in order to solve two problems. First, classifying tweets as 'Rumor' and 'Not Rumor', which can assist users to distinguish the type of tweets. Second, classifying the rumor type that the tweet endorses, denies, questions or is neutral. Although in both problems we investigated the rumor, these two problems are different. Our two-step model pipeline is dynamic in a way that the output of the first step (Rumor Detection) is the input dataset for the next step (Rumor Type Classification). We also designed a new set of pragmatic features along with updating the set of features in Twitter and network-specific category, which could boost the overall performance in our pipeline.

3.4.1 Machine Learning Frameworks

For our experiment, we applied J48, a discriminative classifier that utilizes decision trees and supports various types of attributes. WEKA platform [Hall et al., 2009] is used for training and testing the proposed models in our pipeline.

3.4.2 Feature Extraction

We experimented with content, network, and social meme features. We extended the number of features by including the pragmatic attributes. We employed all the features proposed in [Qazvinian et al., 2011a] in addition to developing more pragmatic attributes as well as additional network features. For network and meme features, we explicitly modeled source and timestamped information and for pragmatic features, we proposed NER, Event, Sentiment, and Emoticon. Table 3.4 listed all the features for the RDC task and marked the new features with "*".

3.4.2.1 Content Features

This set of features is developed using tweet content. We applied various preprocessing granularity levels to measure the impact of preprocessing on the RDC task.

Unigram-Bigram BOW Similar to the content lexical features proposed in [Qazvinian et al., 2011a] we used a BOW feature set comprising word unigrams and bigrams. We employed the WEKA's String To Word Vector along with N-gram tokenizer for creating this feature set with the TF-IDF weighting factor as the matrix cell content corresponding to each feature. We also generated the lemma form of the words in the tweets using WordNet [Miller, 1995] lemmatization capability. Accordingly, we created four feature sets: unigram tokenized word form, unigram lemma form, bigram tokenized word form, and bigram lemma form. Following Example which is selected from the Michelle Obam's Rumor, is an explicit sample for processing in our

	ID	Value
Twitter and Network Specific	* Time	Binary
	* Hashtag	Binary
	Hashtag Content	String
	URL	Binary
	Re-tweet	Binary
	*Reply	Binary
	User ID	Binary
Content	Content Unigram	String
	Content Bigram	String
	Pos Unigram	String
	Pos Bigram	String
Pragmatic	*NER	String
	*Event	String
	*Sentiment	String
	*Emoticon	Binary

Table 3.4: Final list of used features '*' marked features are the appended set of features

framework.

- **Before Preprocessing**

  Separating fact from fiction regarding the size of Michelle Obama's staff (The Yahoo! Newsroom) URL

- **After Preprocessing**

  separate fact regard fiction size michelle obama staff newsroom yahoo URL

3.4.2.2 Pragmatic Features

In an extension to the features proposed by [Qazvinian et al., 2011a], we further explored the explicit modeling of pragmatic features to detect favorable and unfavorable opinions toward specific subjects (such as people, organizations). Applying this set of features offers enormous opportunities for detecting the type of rumors [Nasukawa and Yi, 2003].

Sentiment There are a wide variety of features for sentiment classification on Twitter datasets that have been studied in various publications. We believe that polarity of

a tweet could be an informative factor to extract user's opinion about each rumor. For tagging the sentiment polarity of a tweet, we applied the Stanford Sentiment system [Socher et al., 2013]. We preprocessed the data by removing punctuations, URL, "RT2", and lowercased the content. Each tweet is tagged with one of the following sentiment labels; ***Very Positive, Positive, Neutral, Negative, or Very Negative***.

Emoticon Another pragmatic cue is Emoticon. Studies on modeling and analyzing microblogs, which explicitly use emoticon as a feature, show its impact on classification [Agarwal et al., 2011]. We used the list of popular emotions. We manually designated and labeled the list of entries as either expressing Positive (2), Negative (1), or Neutral (0) emotions.

Named-Entity Recognition (NER) We employed Twitter NLP tools [Ritter et al., 2011] to explicitly extract information about named-entities, such as Location, Person, Organization, etc. In this work, we show how modeling NER has an explicitly positive impact on performance.

Event Extracting the entity *Obama* and the event phrase *praises in connection with Muslims* is much more informative than simply extracting *Obama*. We utilized the same Twitter NLP tools [Ritter et al., 2011] for tagging event labels.

3.4.2.3 Network and Twitter Specific Features

Relying on Twitter specific memes, we expanded features listed in [Qazvinian et al., 2011a] by adding time and network behavior features, such as Reply.

Time It is quite remarkable that social networks spread the news so fast. In a similar task to [Starbird et al., 2014] we analyzed the process of rumor expansion on

Twitter in our dataset. Both the structure of social networks and the process that distributes the news lead to a piece of news becoming viral instantaneously. We labeled and ranked all the days based on the number of tweets posted in a day. We modeled the tweet creation-time attribute. We also observed that more than 90% of rumors are posted during the five busiest days in the collected dataset. Figure 3.2 shows the results of tabulating time frequency of the rumors in the Palin rumor dataset and how the number of rumors changed within a six month period. Accordingly, we designated two labels for the time feature: Busy Day or Regular Day, depending on what type of day tweets were (re)tweeted.

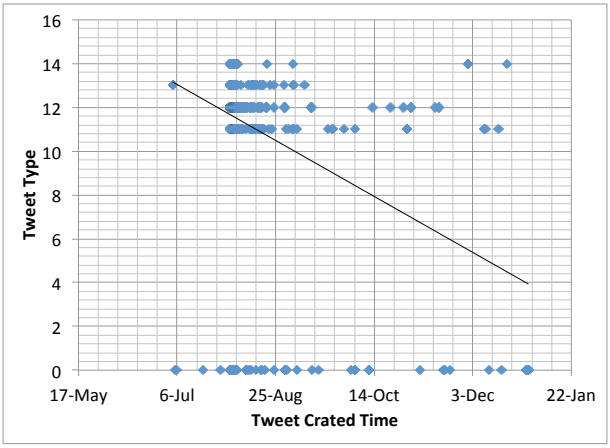


Figure 3.2: Tweet distribution for the Palin rumor collection within a 6 month period

Reply, Re-tweet, User ID Replying and retweeting in microblogs are revealing factors in judging user's trustworthiness when it comes to relaying information [Thomas et al., 2011]. For example, *User A* is more likely to post a rumor than *User B* if *User A* has a history of retweeting or replying to *User C* who also has a rumor spreading history. Investigating the credibility of users is an expensive and almost impossible task, but it is doable when we only want to investigate a specific story. For example, knowing the limited number of users who have a history of posting rumors could be a hint to detect the large number of users that follow, retweet or reply to those tweets.

Hashtag Hashtags serve as brief explanations of the tweet content [Deveaud and Boudin, 2013]. We extracted hashtags in the labeled dataset. Tweets with no hashtags are assigned a value 0 to their hashtag feature dimension, and tweets containing hashtag(s), received a 1 in the hashtag dimension. Additionally, we added all the observed hashtags as feature dimensions, thereby effectively identifying the tweets that share the same hashtag. For compound hashtags, we used a simple heuristic. If the hashtag contained an uppercase character in the middle of the hashtag word, then we split it before the uppercase letter. For instance, ***#SarahDivorce*** is separated into two hashtags and converted to **Sarah** and **Divorce**. We then modeled both compounds and separated hashtags as hashtag feature dimensions.

URL Twitter users share URLs in their tweets to refer to external sources as an authentic proof (a source of grounding) to what they share. All URLs posted in tweets are shortened to 22 characters using the Twitter t.co service. Analyzing the URL is an expensive task and requires a huge source of information to verify the content of the shared URL. We excluded all URLs, but we modeled their presence as a binary feature.

3.4.3 Experimental Design

All the experiments are designed, performed, and evaluated based on various experimental settings and conditions, all elaborated in this section.

3.4.3.1 Data

We experimented with three datasets: the two largest rumor sets, Obama and Palin, and a mixed dataset (MIX) which comprises all the data from the five rumors. We split each of the three datasets into 80% train, 10% development, and 10% test.

3.4.3.2 Experimentation Platform

All experimentations were carried out using the WEKA-3-6 platform [Hall et al., 2009].

3.4.3.3 Baselines

We adopted two baselines: Majority and limiting the features to the set of features proposed in [Qazvinian et al., 2011a], which are Content, Hashtag-Content, URL, Re-tweet, and User ID. As the name indicates, the Majority baseline assigns the majority of label from the training dataset to all the test data.

3.4.3.4 Experimental Conditions and Evaluation Metrics

We had two main experimental conditions: Single-Step Rumor Detection and Classification (SRDC) and a Two-Step Rumor Detection and Classification (TRDC). We employed the set of 15 features listed in Table 3.4. Information about SRDC and TRDC is illustrated in Table 3.5. In the development phase multiple settings and configurations were performed on the development dataset for tuning, then the models that achieved the highest performance were used in the test set. Evaluating the performance of the proposed technique in rumor detection should rely upon both the number of relevant rumors that are selected (recall) and the number of selected rumors that are relevant (precision). Hence, we calculated F-measure ($Beta = 1$) as the harmonic mean of precision and recall due to its bias as an evaluation metric. Table 3.9 shows the F-measure value for the different settings on the test set.

1st Step		2nd Step
Method	Labels	Labels
(SRDC) 6-way classification	(0)(11)(12)(13)(14)(2)	
(TRDC) 3-way (1st step) | 4-way (2nd step) classification	(0)(2)(11-14)	(11)(12)(13)(14)

Table 3.5: Number of features and labels used in single step and two step classifications

3.4.4 Results

In this section, the impacts of different experimental conditions are investigated.

3.4.4.1 SRDC and TRDC

By studying the results in Table 3.6, it can be observed that TRDC significantly outperforms SRDC, since TRDC achieves an F-measure of 82.9% compared to 74% in SRDC for the MIX dataset, and 85.4% for the Obama dataset compared to a 71.7% in SRDC. By comparing the F-measure with the Majority baseline and the features proposed in [Qazvinian et al., 2011a](VAR11) as the second baseline, we could explicitly see how applying the proposed methodology, and set of features enhances the overall performance in certain rumors, and also leads to acceptable performance in the MIX dataset.

Dataset	Method	Our 15 Feat.	VAR11 Feat.
	Majority	0.30	
	SRDC	**0.743**	**0.748**
MIX	TRDC	**0.83**	**0.83**
	Majority	0.33	
	SRDC	**0.717**	0.705
1-3 Obama	TRDC	**0.854**	0.844
	Majority	0.46	
	SRDC	**0.754**	0.748
Palin	TRDC	**0.79**	0.70

Table 3.6: F-measure results of SRDC and TRDC methods employing 15 features and VAR11 features

3.4.4.2 Impact of Feature Set

In this experiment, we assessed the performance of different groups of features individually. Figure 3.6 shows the average F-measure and precision of SRDC and TRDC by employing the Content, Pragmatic and Network sets of features. As shown in Figure 3.6, employing the Content set of features yields the overall best precision. In

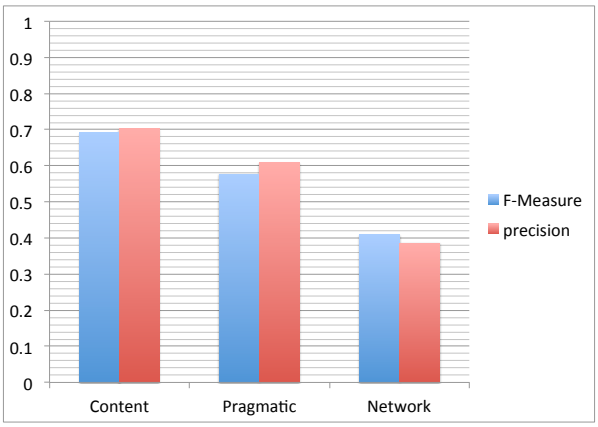


Figure 3.3: The Average F-measure and precision of SRDC and TRDC classifications employing each group of features: Content, Pragmatic, Network

contrast to the other features, the network feature set had a minimum impact on our classification.

3.4.4.3 Impact of Preprocessing

As mentioned above, we applied various levels of preprocessing to the content of tweets such as stemming, lemmatization, punctuation removal, lowercasing, and stop words removal. We measured the impact of applying such preprocessing versus no preprocessing. Figure 3.4 illustrates that accuracy doesn't benefit from preprocessing and results in the loss of valuable information.

3.5 TLV Approach

TRDC and V11 results indicate that content features outperform other features in the rumor detection task. In this experiment, Rumor Retrieval (RR) pipeline is implemented with a new set of features. In order to overcome the missing word and short length tweet issue, TLV representation [Guo and Diab, 2012] is proposed. Moreover, we apply the belief feature to capture the level of committed belief of the author toward the proposition in the tweet. For more analysis, the current dataset is

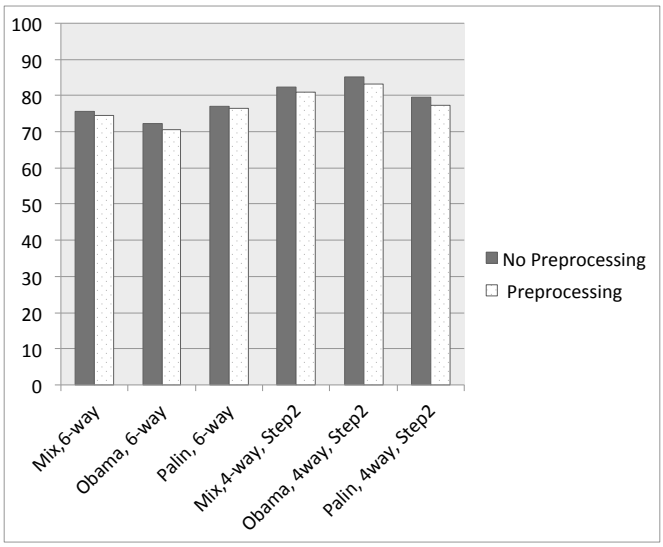


Figure 3.4: The overall accuracy in different experiments with and without preprocessing

extended by collecting more tweets about "Obama is Muslim" rumor by employing the same data collection pipeline as V11. The collected data are labeled using the TRDC component which led to highest performance in RDC task. We further study the belief change on different opinion on the extended dataset and noticed that endorsing opinion about "Obama is Muslim" rumor show more Committed Belief (CB) than denying opinion about that rumor. Figure 3.5 illustrates the TLV pipeline with corresponding classification labels. As it is shown the TLV is a coarse-grained approach with a fewer number of classes in comparison to RDC where non-rumor and unidentified labels are combined so as for deny, neutral, and questioning in rumor classification step.

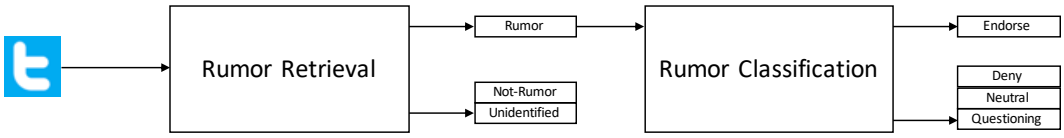


Figure 3.5: The overall pipeline for TLV

3.5.1 Silver Data

We applied the same regex which were used at Table 3.1 for the Obama rumor and collected more than 7000 tweets from 2014 and 2016. We named the new data as silver-data and use them to investigate how the belief stance of users has changed toward the "Obama is Muslim" rumor from 2010 to 2016. Table 3.3 shows statistics for the Silver-Data. We labeled the silver-data as 0(Non-Rumor), 11(Believe), and merged 12 (Deny-12, Doubtful-13, and Neutral-14). For tagging the silver data, we used the original Obama dataset as the train dataset. Table 3.8 shows what labels are being used for the rumors retrieval and silver-data creation experiment.

	0 non-rumor	11 Believe	12 (Deny/ Doubtful/ Neutral)	Total
Obama2014	2940	3055	678	3738
Obama2016	1250	856	379	2485

Table 3.7: List of Tweets in Silver Data

1st Step		2nd Step
Method	Labels	Labels
(2-way, 2 step)	(0,2)(11-14)	(11)(12,13,14)

Table 3.8: Labels Used in Rumor Retrieval and Rumor Type Classification for Silver Data

3.5.2 Features

In designing the new set of features for the Rumor Retrieval (RR) task, we considered two key points. First, addressing the missing words and length issue in Twitter TLV and second, extracting a feature that implies the user's belief about each rumor. We also present and conduct RR experiment applying RDC features as one of our baselines. We designed and employed a new set of features in RDC which are tagged by "*" in Table 3.4. Untagged features represent the features that are used in V11.

3.5.2.1 Tweet Latent Vector (TLV)

The main intuition behind TLV is to create the latent vector representative of each tweet since, in most of the tweets, there are too few observed words to tell us what the sentence is about. We assume that the semantic space of both the observed and missing words make up the complete semantic profile of a sentence. We propose the TLV feature by applying the Semantic Textual Similarity (STS) model proposed by [Guo and Diab, 2012] [Guo et al., 2014], which built on the WordNet+Wiktionary+Brown+training dataset. STS preprocess each short text by tokenization and stemming, then changes the preprocessed data by removing infrequent words and TF-IDF weighting, and finally uses the model to extract the latent semantics, which is represented as a 100-dimension vector.

3.5.2.2 Level of Committed Belief

For the belief feature, we investigate the level of committed belief for each tweet. We relied on the Werner et al. [Werner et al., 2015a] belief tagger to tag the LCB toward the proposition in a tweet. The feature values are set to a binary 0 or 1 for each CB, NCB, NA, and ROB corresponding to unseen or observed. The following example illustrates how the belief feature values are created.

- Did yall <NA>know</NA> 1 in 5 people <CB>thought</CB> obama is a Muslim
- Feature Values : CB:1 NCB:0 NA:1 ROB:0

3.5.2.3 Content Unigram

Similar to the content lexical features proposed in RDC and V11 we use the BOW feature set comprised of word unigrams. The feature values are set to a binary 0 or 1 for the word unigram vector representative of each tweet.

3.5.3 Experimental Design

All the experiments are conducted and evaluated based on various experimental settings. We utilized different datasets, features, and machine learning approaches, which are elaborated in this section.

3.5.3.1 Data

We conduct our experiments with two datasets: for the RR experiment, we use the mixed dataset (MIX) which comprises all the data from the five rumors. We split each of the three datasets into 80% train, 10% development, and 10% test. For the belief investigation experiment, we only rely on the Obama dataset. After tagging the silver data by applying the TRDC model, we randomly select 400 rumors (200 believer-11 and 200 denier-12) from 2010(Gold Data), 2014, and 2016(Silver-data), and investigate how tweet writer's beliefs about the Obama rumors have changed in recent years.

3.5.3.2 Baseline

For the RR experiment, we adopt three baselines: Majority, RDC features, and the V11 model. The Majority baseline assigns the majority label from the training dataset to all the test data. In the RDC baseline, we perform the RR experiment by relying on the features that are proposed in RDC and shown in Table 3.4. We performed the RR experiment with different models in Weka platform and chose the SMO, which yield to the highest result in this experiment. We also compared our results with V11, which reported the results as Mean Average Precision (MAP).

3.5.3.3 Machine Learning Tools

For the experiments, we employ SVM Tree Kernel model, which was proposed by Alessandro Moschitti [Moschitti, 2004]. In another experiment, we perform the RR

task by applying RDC features, which are illustrated at Table 3.4.

3.5.4 Experiments and Evaluations

We implement two main experimental pipelines: Rumor Retrieval (RR) and Belief investigation. Content and TLV features are employed for the RR task, and then we conduct our experiment in two different phases. In the development phase, we utilized development data for tuning. Then the model, which could reach the highest performance, is used in the test dataset. Evaluating the performance of the proposed technique in rumor detection should rely on both the number of relevant rumors that are selected (recall) and the number of selected rumors that are relevant (precision) since both of them are presented in this work. In another experiment, we investigate the belief change in the Obama rumors. We define two scores for analyzing the belief for the rumor poster/ writer. T_{iCB} and T_{iNCB} are defined for each rumor in the Obama dataset. Each of $T_{iBeliefTag}$ corresponds to a number of seen tag in each tweet. We calculate the belief scores for each Obama rumor dataset separately. We apply formula 1 on believer (11) and denier (12) rumor in the Obama datasets.

$$\frac{\#R_{11}BeliefTag}{\#R_{12}BeliefTag} \tag{3.1}$$

3.5.5 Results

In this section, the impact of the different experimental setup is discussed. We first elaborate on each experiment and then compare our methodology with the baselines.

3.5.5.1 Rumors Retrieval

We perform the RR task by applying two sets of features and compare the results with the three baselines. We perform the RR experiment by employing the gold dataset to detect Not-Rumor(0 and 2) and Rumor(11, 12, 13, and 14) in one-step two-way

Data	Method	RDC(pr,rec)	V11	TLV
	Majority	0.51,0.71	—	—
MIX	RR	0.94,0.94	**0.965**	**0.972,0.99**
	Majority	0.27,0.52	—	—
Obama	RR	**0.91,0.91**	—-	**0.971,1.0**

Table 3.9: Precision and Recall Of RR task by Employing TLV+Content Unigram, RDC, and V11 is reported as Mean Average Precision (MAP)

classification experiment. For the RDC baseline, we applied all the 15 features listed in Table 3.4. We investigated the performance of different classifiers including J48(Decision Tree), Naive Base(NB,) and SMO and picked SMO which has outperformed the others. In a similar experiment for the TLV task we employe TLV and Content features by applying the SVM Tree Kernel model, which lead to 0.972, 0.99 for the MIX and 0.971, 1.0 (precision and Recall) for the Obama gold dataset. Table 3.9 shows how we outperform the other baselines (Majority, RDC, and V11) by employing the proposed features.

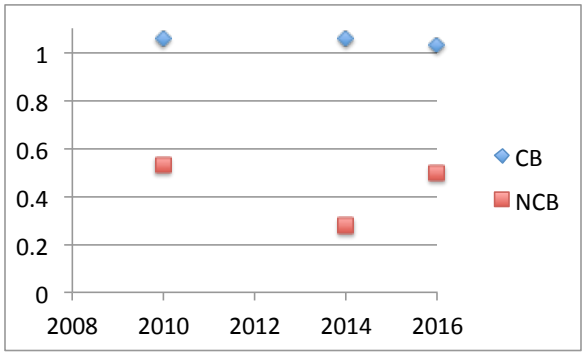


Figure 3.6: The CB and NCB score for the Obama Dataset in 2010, 2014, and 2016

3.5.5.2 Belief Analysis

We propose Equation (3.1) to measure the belief score for the Obama dataset in different years. Then we investigate how the Committed Belief (CB) and Non-Committed Belief (NCB) have changed among rumor believers as well as deniers from 2010 to 2016.

Figure 3.6 shows the Committed Belief and Non-Committed Belief scores among the three datasets. Scores above one mean that the number of the committed belief words in rumor believers is more than in rumor deniers. It is interesting to see that the belief score for all three years is higher than one. A simple interpretation of that would be, in all 2010, 2014, and 2016, people who were rumor believers in "Obama Being Muslim" show more committed belief than those who deny Obama being Muslim. On the other hand, we see the NCB ratio, which is less than one for the same years. NCB means when SW presents a weak belief towards something. Having below one for NCB could be interpreted as deniers showing weak belief toward the fact that Obama is not a Muslim in 2010, 2014, and 2016. It is important to state that by receiving more data, we can attain more accurate behavior belief.

3.6 Summary

We studied the impact of applying Tweet Latent Vector TLV instead of conventional methods like the bag of word or TF_IDF on the problem of rumor retrieval on Twitter. We also investigated the impact of different level of preprocessing in the rumor detection tasks. We observed in the experiments that preprocessing the tweets for rumor tracking yields to lower performance which was caused by removing the punctuation from the content. Our analysis also proved that applying the dense representation of tweet is a better solution to represent the short-length documents and tweet in particular. From the Machine learning point of view, applying TLV is computationally more efficient and less prone to overfitting.

Chapter 4: Committed Belief Tagging

4.1 Introduction

One useful modality marker for modeling the pragmatics of natural language use as well as the cognitive states of speakers/writers (SW) is identifying their Level of Committed Belief (LCB). Identifying the Level of Committed Belief (LCB) is of relevance to opinion mining, veridicality, and certainty assessment of one's opinion stance [Wiebe et al., 2005]. LCB could also contribute to identifying people's attitudes toward events: Are people hedging or committed to a specific view of a proposition or event? Also, LCB may help in calibrating the level of confidence someone has regarding an expressed opinion, determining the veracity of piece of information, or understanding the power relation in linguistic communication. For example, [Hamidian and Diab, 2016] used the LCB definition and the tagging model of [Werner et al., 2015a] on rumors and measured the opinion stance of a group of people about certain rumors on social media. Given a sentence such as "John is in Paris." vs. "John might be going to Paris." indicate the LCB of the speaker/writer (SW). In the former, the SW is committed to the proposition, while in the latter, the SW is not committed, s/he is hedging. At first blush, it might be thought that simply modeling tense and lexical items, LCB could be identified and tagged, however, if the first utterance were: "I am not sure John is in Paris.", then the LCB tag would change to non committed belief, indicating the role of syntax on the value judgement.

Most of the works to date addressed the problem of LCB tagging using supervised learning approaches mainly leveraging handcrafted features relying on feature engineering techniques. In contrast with previous approaches and more in line with current trends in NLP, we adopt a neural network framework for feature representation, however inspired by the previous feature engineering approaches. Despite the success

of feature engineering based techniques, identifying which features to explicitly model is complex and time consuming. The problem is exacerbated when we need to solve the problem for multigenre data. To that end, we leverage word distributional representations that are sensitive to morphological variants (different spellings) and unseen words, i.e. out of vocabulary words. We use both character and word embeddings in our models coupled with the power of pre-trained embeddings to mitigate out of domain scenarios. The LCB task consists of two conceptual subtasks: identifying the scope of a proposition, and classifying each proposition as Committed Belief (CB), Non-Committed Belief (NCB), Reported Belief (ROB), and Non-Attributable Belief (NA). We adopt a joint model in which the proposition heads are chosen and classified simultaneously. We employ a recurrent neural network model (RNN), namely an Long Short Term Memory (LSTM) implementation, to address the problem of LCB Tagging. Similar to previous state of the art work, we model LCB tagging as a sequence tagging problem. We use two of the most recently annotated training data sets for LCB. Both corpora use similar labels for LCB, but apply them to different linguistic spans. The first corpus is collected from English discussion forum threads (DF) and annotates LCB on the head of the proposition in context. The second dataset is collected from both Discussion Forum (DF) and Newswire (NW) data, but the scope of the annotation is on specific event categories.

Our contributions can be summarized as follows: 1) We systematically compare the performance of the proposed model on LCB tagging with the current state-of-the-art; 2) Our work is the first to apply a neural network based approach to the problem of LCB sequence tagging; 3) We show that explicitly modeling genre variation as an auxiliary task leads to better domain generalization resulting in state of the art performance.

4.2 Related Work

Initial work on identifying LCB and factuality was published by [Diab et al., 2009], releasing the LU corpus. The LU corpus comprises annotations of propositional belief in context. It was devised as a three way annotation for LCB: Committed Belief (CB), reflecting that the SW firmly believes in a proposition; Non-Committed belief (NCB), reflecting an SW's weak belief in a proposition; and, Non-Attributable belief (NA), indicating that the SW is not (or could not be) expressing a belief in the proposition (e.g., desires, questions, etc.). Apart from the annotated corpus, they also proposed a supervised system to predict LCB by leveraging features like Part of Speech (POS) tags, word type features, AlphaNumeric, and shallow syntactic features such as chunk information. The system set the stage for state of the art performance. Follow on work by [Prabhakaran et al., 2010] improved over [Diab et al., 2009] results by adding more hand crafted lexical and syntactic features in a supervised framework using Support Vector Machines (SVM) and Conditional Random Fields (Conditional Random Fields (CRF)). Work by [Werner et al., 2015a] followed the [Prabhakaran et al., 2010] work and adopted another supervised approach to the LCB problem as well as applying their methods to both the LU corpus and the Factbank [Saurí and Pustejovsky, 2009] corpus. The main contribution of that work was a new set of features and an in-depth comparison of machine learning techniques for this problem. Finally, [Prabhakaran et al., 2015] proposed a novel dataset with new definition and annotation scheme for LCB sharing the three LCB tags of CB, NCB, and NA as defined in [Diab et al., 2009] but also adding another tag Reported Belief (ROB) which annotates the case where the SW reports someone else's stated belief, regardless of their own belief. The annotated data consisted of English text from discussion forum threads. For building a system, they proposed a new set of features including hedge features in addition to what was offered by [Prabhakaran et al., 2010]. Their

proposed model, to date, is the state of the art for LCB tagging on propositions.

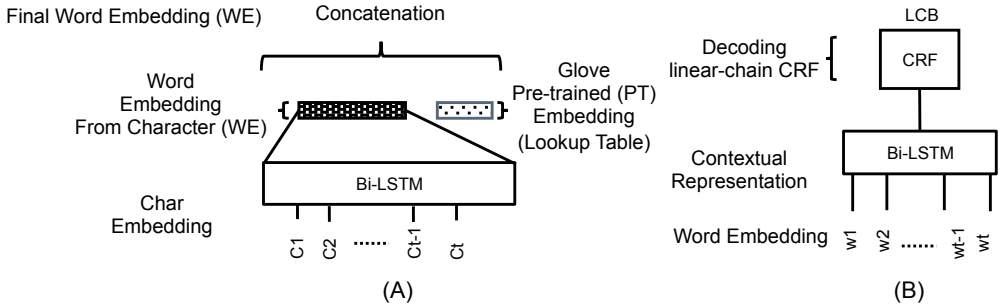


Figure 4.1: (A)The embedding of each word is generated by employing the output of a character embedding using a Bi-LSTM layer concatenated with pre-trained embeddings for each word. (B) The word embedding from (A) is fed to the Bi-LSTM network for the LCB task. The proposed model contains a Bi-LSTM followed by a Linear chain Conditional Random field (CRF) classifier.

4.3 Approach

Following the previous work in [Prabhakaran et al., 2015], we adopt a supervised approach to LCB tagging. We propose a neural network framework. First, we investigate the proposed pipeline on a single-genre corpus whose annotation marks the belief on the head of a proposition. Next, we study the performance of our proposed approach on a multi-genre corpus where the LCB annotations are targeting specific categories of events. We go beyond previous approaches by examining the significance of genre detection as an auxiliary task for LCB tagging.

4.3.1 Dataset

We leverage two Linguistic Data Consortium (LDC) corpora in our experiments.

4.3.1.1 $LDC_{Proposition}$

This corpus comprises English discussion forum (DF) threads, where the annotations mark LCB on the head of clausal propositions. The four-way belief categories are described as follows:

- Committed Belief (CB): the SW firmly believes in the proposition. For example, *Mark will prove he is innocent*, or, even stronger, *I know that Mark will prove he is innocent in court*;

- Non Committed Belief (NCB): the SW exhibits a weak belief about the proposition. For example, *Mark may prove that he is innocent* or *I doubt he can prove he is innocent*;

- Reported Belief (ROB): the SW is only relaying the belief or lack thereof of someone else via reported speech. For example, *The CNN anchor reported that Mark won't be able to prove he is innocent*;

- Non Attributable Belief (NA), the SW is not (or could not be) expressing a belief in the proposition (e.g., desires, questions, etc.) This includes the expressions of desire *Some people wish Mark couldn't prove he is innocent*, questions *Will Mark prove he is innocent?*.

4.3.1.2 LDC_{Event}

This corpus is the belief and sentiment corpus annotating discussion forum (DF), and newswire (NW) genres. LCB annotations reflect the SW belief in an event in context rather than a proposition, namely, the annotations mark LCB on specific events (event-target) types: Life, Movement, Business, Conflict, Contact, Personnel, Transaction, and Justice. Similar to $LDC_{Proposition}$ all the mentioned type of events are labeled with the four-way belief categories (CB, NCB, ROB, NA). The following example illustrates the difference in annotation units between LDC_{Event} and $LDC_{Proposition}$.

LDC_{Event}:

- My dad **<CB> died </CB>** a couple of years ago, and I was alone at his **<CB> funeral </CB>**, so I've been through the process, and can share my experience.

$LDC_{Proposition}$:

- My dad **<CB> died </CB>** a couple of years ago, and I was alone at his funeral, so I've been through the process, and **<CB> can </CB> <CB> share </CB>** my experience.

This example is a sentence chosen from both selected corpora. Despite the similarity in LCB categories, these two corpora are distinctive because of the difference in annotation objectives. LDC_{Event} targets specific categories of events as opposed to $LDC_{proposition}$ where the annotation marks all the clausal propositions. the two annotation schemes are applied to a subset of the DF corpus, however, to date, there has not been an attempt to merge the two annotation frameworks. Hence we will treat them as two separate data sets. We adopt the same basic neural architecture for both data sets.

4.3.2 LCB Tagging for (*Propositions*)

We address this problem as 5-way (CB, NCB, ROB, NA, and O) multi-class classification task on the word level by leveraging the $LDC_{Proposition}$ corpus for all the experiments.

The following examples illustrate the LCB tags in different cases:

1. He <CB> asked </CB> my sister to <NA> go </NA> to the party.
2. I'm not sure if he <NCB> asked </NCB>.
3. He <CB> told </CB> me that she could <ROB> go </ROB> to the party.
4. Is she <NA> going </NA> to the party? In the first example, the "go" is labeled NA because, even though the SW is committed to the belief that his sister was asked to go to the party, the SW is not expressing a belief about whether she is going to the party or not. In the second example, "asked" is labeled as NCB, because of "Not sure" which implies uncertainty. In the third example, "go" is tagged as ROB, since it is reporting someone else's stated belief. Finally, in example 4, "going" is tagged as

NA, since it is questioning the proposition of going to the party, therefore, "going" being the head of the proposition clause is tagged as NA. As is shown, the LCB task is challenging, since it requires deep vocabulary understanding and thorough interpretation of context.

4.3.2.1 Proposed Neural Network Architecture

Given the success of recurrent neural networks (RNN) on language problems especially for sequence tagging, we adopt an LSTM RNN framework for our problem as illustrated in Figure 4.1. We model the lexical and contextual information as follows:

- Character and Word Embeddings: For each word we need to employ a dense representation. We use a Bi-LSTM network as it shown in Figure 4.1.A to encode the morphology and word embeddings from characters. A Bi-LSTM at the character level is applied, but it is worth noting that we could apply any other network architecture here. Each character C_i of a word $W = [C_1, C_2, ..., C_t]$ is passed to a Bi-LSTM as a sequence of characters and the output of the last hidden layers are concatenated yielding a vector $W_{Character}$. Intuitively the concatenated fixed size vectors $W_{Character}$ capture word morphology. $W_{Character}$ is concatenated with a pre-trained word embedding from GloVe $W_{pre-trained-Glove}$ to get the final word representation.
  W_{final}=$W_{Character}$+$W_{pre-trained-Glove}$.

- Contextual Encoding: Once the word embedding is created we use Bi-LSTM layer as illustrated in Figure 4.1.B to encode the contextual meaning from the sequence of word vectors $W_1, W_2, ..., W_t$ and consequently obtain another sequence of word vectors from the concatenation of hidden states $h_1, h_2, ..., h_t$ for each LSTM layer. The concatenation of W_t and h_t encodes word level syntax, semantics and contextual information.

- Decoding: in this phase the vector representative of each word W is associated with h which captures all the information from characters, meaning, and context. Considering the number of tags as k and the vector length as l we compute the vector of scores for each tag as $S = W\ .\ h + b\ \ which\ W \in R^{k \times l}$ and $b \in R^k$. $S[i]$ refers to the probability score for each of the belief categories i for a word w.

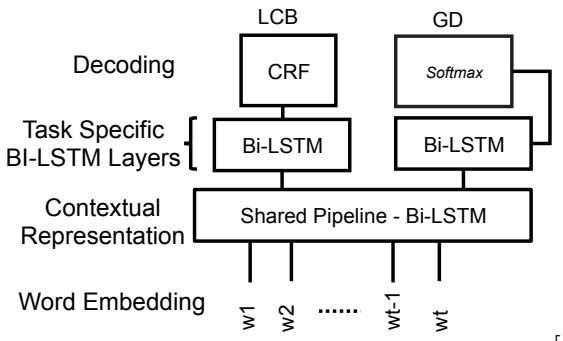
- Prediction and Loss: after decoding we use the cross-entropy loss function $L = -\log P(\tilde{y}))$ which $P(\tilde{y})$ is the probability of the correct sequence $\tilde{y}$.

- Tagging: we use Linear-chain CRF for sequence tagging.

As opposed to CB, other belief categories such as *(NA) and (NCB)*, appear more frequently in DF compared to NW. Yet we know that there is a high number of shared topics across DF and NW corpora, one interpretation of this observation is that SWs in NW data show more certainty than they do in DF. Accordingly, we in this work we tease apart the impact of genre variation on the LCB task. We hypothesize that knowing the data genre explicitly being modeled will have an impact on LCB tagging. Consequently for LCB events, we model the problem in a joint multitask learning framework where genre detection is set up as an auxiliary task to LCB sequence tagging.

4.3.2.2 Multi-task learning (MTL) Network

The simple definition of Multi-task Learning (MTL) is to share representations between related auxiliary tasks, enabling the model to generalize better on the original task [Caruna, 1993] [Yang and Hospedales, 2016]. Our main objective is to improve the proposed model for LCB tagging by explicitly leveraging an auxiliary task such as genre detection (GD). We propose using a hard-parameter-sharing pipeline. Figure 4.2 demonstrates the adopted neural network model with the Multi Task Learning (MTL) framework. We employ the word embedding output form Figure 4.1.A and use that as



[t]

Figure 4.2: The word embedding is fed to the multi-task Bi-LSTM network for LCB and GD tasks. The proposed model contains a shared Bi-LSTM with two Bi-LSTM layers of each task following corresponding CRFs and a Softmax classification layer for LCB and GD, respectively.

the input to the MTL pipeline at Figure 4.2. Figure 4.1.B. is modified with following task-specific Bi-LSTM layer and loss function to be used in the MTL pipeline.

- Task-Specific layers: Contextual representation of each word along with the last hidden state encoding the sentence level meaning are passed to the task-specific Bi-LSTM layers.

- Prediction and Loss; after decoding we use the Linear-chain CRF for LCB sequence tagging and Softmax for Genre Detection (GD). We use the cross-entropy loss function $L_{task_i} = -\log P(\tilde{y}))$ where $P(\tilde{y})$ is the probability of the correct sequence $\tilde{y}$ for $task_i$. The overall loss values are calculated and passed to the optimization function using the following formula $\sum_{i=1}^{t} loss_{task_i}$ where t is the total number of the tasks ($t = 2$).

4.3.2.3 Auxiliary Task: Genre Detection (GD)

Language consists of much more than just content [Pavlick and Tetreault, 2016]; choice of word, entities, hedging, and many other factors play critical roles in changing the tone of a sentence. Given the observed distributional correlations of the LCB tags, we hypothesize that data genre will have an impact on LCB tagging. We cast the

problem of genre detection (GD) as a two way classification problem, for every input sentence we predict a tag of either NW or DF. However, the corpora that we have are not tagged for genre on the sentence level. As mentioned earlier, we adopt a supervised approach to LCB tagging in general. Hence we need labeled training data for Genre Detection (GD). In the spirit of distant supervision, we make the simplifying assumption that if a sentence comes from a NW corpus, then every sentence from that corpus is tagged with a NW label and likewise for DF sentences. n the Our intuition is that domain or genre could be correlated with the degree of belief. We used the shared pipeline with a task-specific Bi-LSTM and a Softmax classification layer, as shown in Figure 4.2. For each sentence, we use the output states that contain the forward and backward final states of Bi-LSTM as the input to the Softmax layer.

4.4 Experimental Setup

4.4.1 Training

For all of the pipelines, the network is trained with back propagation using Adam [Kingma and Ba, 2014], Root Mean Square Propagation (RmsProp), and Stochastic Gradient Descent (SGD) optimization algorithms. The parameters get updated in every training epoch. The character and Glove pre-trained embedding size [100,200,300] are examined with batch size 20 with 100 epochs. The training is stopped after no improvements in five consecutive epochs to ensure the convergence of the models. The highest performance on the development set was achieved under the following parameters: hidden size of Bi-LSTM (100); optimization (RMSprop); initial learning rate (0.003); L2 ($Lambda = 0.1$); character and word embedding size (300,100); dropout size (0.3).

Training/Dev Set				
Genre	CB	NCB	ROB	NA
DF	79995 (56%)	3890 (3%)	7150 (5%)	52205 (36%)
Evaluation Set				
DF	8730 (50%)	583 (3%)	941 (5%)	7299 (42%)

Table 4.1: $LDC_{proposition}$ training and evaluation set statistics. The training/dev set was split into 90% for training and 10% for development maintaining the distributional proportions across LCB categories.

4.4.2 Datasets

In all the experiments, we use the same datasets for training and evaluation provided in previous studies. Table 4.1 and Table 4.2 illustrate the statistics of genre and belief categories and their corresponding percentages in the training/development and evaluation sets for both $LCB_{Proposition}$ and LDC_{Event} corpora, respectively.

Tagging Scheme The task of LCB is assigning a belief label to every word in a sentence. A single belief tag could span more than one token. Therefore, we use an IOB scheme, in which every token is labeled B_{LCBTag} if the token is the beginning of an LCB tag of CB, NCB, ROB, NA, I_{LCBTag} if it is inside an LCB tag, or O otherwise. This results in 13 tags.

4.4.3 Experiments

We investigate the effectiveness of the different components of the proposed network at Figure 4.1 on $LDC_{Proposition}$ and LDC_{Event}, in addition to systematic comparison with the current state-of-the-art method on $LDC_{Proposition}$. We devise the following experimental conditions:

Training/Dev Set				
Genre	CB	NCB	ROB	NA
DF	3263 (67%)	111 (2%)	43 (0.8%)	1489 (30%)
NW	1421 (86%)	21 (1%)	13 (0.1%)	181 (11%)
Mix	4684	132	56	1670
Evaluation Set				
DF	1002	13	7	295
NW	379	5	2	77
Mix	1381	18	9	372

Table 4.2: Training and evaluation set statistics of belief categories in DF, NW, and Mix corpora for LDC_{event}. The training/dev set was split into 90% for training and 10% for development maintaining the distributional proportions across LCB categories.

4.4.3.1 Baseline: State of the art to data using feature engineering (P15)

[Prabhakaran et al., 2015] (P15) proposed a supervised pipeline with a wide range of lexical, syntactic, and hedging features using a CRF model for sequence tagging. We view this as our strong baseline model. We use the same training, development, and evaluation set as in P15 from $LDC_{Proposition}$ to obtain a fair comparison.

4.4.3.2 Measuring the Impact of explicit Character and Contextual representations modeling

We aim to investigate the impact of character embedding, pre-trained word embedding, and contextual representation (CR) of the proposed Bi-LSTM CRF network. We adopt four different settings:

***WE*$_{Pretrained}$**

We use a simple network where we only use pre-trained word embeddings ($WE_{Pretrained}$) as our word representations followed by the CRF output layer.

***WE*$_{Char}$**

In this experiment, we aim to measure the impact of using an explicit character

encoding layer in our network. We use the character level (WE_{Char}) representations with and without $WE_{Pretrained}$.

***WE*$_{Char}$, *WE*$_{Pretrained}$, Contextual Representation (*CR*)**

In this setting, all the components are combined.

4.4.3.3 Examining the Impact of the GD auxiliary task in multi-genre LCB events

The primary objective of this experiment is to measure the impact of using GD as an auxiliary task to the LCB task using the MTL framework (Figure 4.2). All the experiments conducted with and without the GD as the auxiliary tasks. It is worth mentioning that this experiment is only relevant to the LDC_{Event} dataset as it is a multi-genre corpus and cannot be applied to single-genre corpus $LDC_{Proposition}$ since the genre information would not provide any additional signal to the LCB model in the latter case.

4.4.3.4 Comparison between Multi-Genre and Single-Genre LCB

Adding additional data from another genre is not always helpful since it could add some noise and consequently harm the generalization power of the model. Our objective is to investigate if the proposed multi-genre MTL pipeline performs better than the single-task with a mixed genres set up. Employing the LDC_Event dataset, we decoupled the DF and NW training sets with corresponding development and evaluation sets and used the Figure 4.2 pipeline with only single task (LCB) for training. In another experiment, we merged the training sets of NW and DF dataset, resulting in a mixed data set, and tested the model on the evaluation set of each genre separately.

4.4.4 Results and Evaluation

We conducted two sets of experiments to assess the impact of applying the NN approach and Multi-task learning with the auxiliary task on the LCB. Both sets with sub-experiments are conducted in two phases. In the development phase, the development data were used for tuning, and in the evaluation phase, the tuned model with the highest performance was used for evaluation. It is worth noting that the majority of the words will not be tagged with any of the four belief classes. (Recall that most words have none of the four belief classes). In all the experiments we report F (Beta=1)-measure as the harmonic mean between (P)recision and (R)ecall. The best performing configuration and corresponding F-Measures for the experimental setup discussed in the previous section are illustrated in Table 4.3, Table 4.6, and Table 4.5

Exp. Cond.	LDC_{Pr}	LDC_{Ev}
Baseline P15	69.1	–
WE_{Pre}	44.97	48.46
WE_{Ch}	52.67	46.0
WE_{Ch}+WE_{Pre}	60.24	57.55
WE_{Ch}+WE_{Pre}+CR	75.09	58.76
WE_{Ch}+$WE_{Pre+trained}$+CR	**75.94**	**59.82**

Table 4.3: Results obtained by different experimental setup for LCB task including the (Ch)aracter Embedding, (Pre)trained Word Embedding (WE), and Contextual Representation (CR) on $LDC_{(Pr)oposition}$ and $LDC_{(Ev)ent}corpora$.

4.4.4.1 Ablation Study

Table 4.3 presents the comparison of the best F-Measure for various experimental setups for the LCB task on $LDC_{Proposition}$ and LDC_{Event} corpora. The results reflect the single task with no GD considerations for the LCB on the LDC_{Event} data.

- **P15** leveraged handcrafted features, such as hedging, part of speech, dependency parser, sentiment, and WordNet with SVM classifier. Overall F-Measure of P15 is 69.1 on the $LDC_{Proposition}$. P15 did not study the LCB on events problem.

- $\boldsymbol{WE_{Pretrained}}$ is the simplest model by using the pre-trained word embedding and CRF which leads to lowest performance on $\boldsymbol{LDC_{Proposition}}$ in our experiments, achieving 44.97% and 48.46% on $\boldsymbol{LDC_{Proposition}}$ and $\boldsymbol{LDC_{Event}}$ corpora respectively. It is interesting to see that the pre-trained word embedding performed relatively better in $\boldsymbol{LDC_{Event}}$. We assume that an alignment in the genre of the pre-trained word embeddings with the NW present for Linguistic Data Consortium (LDC) event corpora could be the reason.

- $\boldsymbol{WE_{Char}}$ reflects the results of leveraging character embeddings with no pre-trained word embeddings. We note that for $\boldsymbol{LDC_{Proposition}}$ this condition yielded better results than using pretrained word embeddings alone as in the previous condition, from 44.97% going up to 52.67%. However we see a drop in performance for LCB event tagging dropping from 48.46% to 46.0%. One reason for this observation could be the fact that $\boldsymbol{LDC_{Event}}$ is a relatively small corpus.

- $\boldsymbol{WE_{Char}}$+$\boldsymbol{WE_{pre}}$ significantly improved the results in both LCB tasks on both data sets, as we reached up to 60.24% and 57.55% F-Measure for $\boldsymbol{LDC_{Proposition}}$ and $\boldsymbol{LDC_{Event}}$, respectively.

- $\boldsymbol{WE_{Char}}$+$\boldsymbol{WE_{pre}}$+$\boldsymbol{CR}$

  Encoding the contextual representation layer significantly improves the LCB task. We conducted all the experiments with and without training the pre-trained word embedding. The best results were obtained by using all the features combined yielding 75.94% on $\boldsymbol{LDC_{proposition}}$ task and 59.82% on the $\boldsymbol{LDC_{Event}}$ task by the proposed pipeline illustrated in Figure 4.1.

	P15			$\boldsymbol{LDC_{Proposition}}$		
Class	Pr	Re	F1	Pr	Re	F1
CB	68.9	77.9	73.1	**75.01**	**84.25**	**79.36**
NCB	52.9	29.7	38.0	**58.98**	**41.75**	**48.89**
ROB	43.8	15.6	23.0	**55.67**	**34.97**	**42.96**
NA	80.1	62.0	69.0	**82.39**	**74.97**	**78.50**
Overall	72.0	66.4	69.1	**76.20**	**76.46**	**75.94**

Table 4.4: Contrasting Results per LCB tag in P15 model and our best condition neural network LCB model, $\boldsymbol{WE_{Ch}}$+$\boldsymbol{WE_{Pre+trained}}$+$\boldsymbol{CR}$, on $\boldsymbol{LDC_{Proposition}}$.

4.4.4.2 Results Per Belief Category in the LCB Proposition task

We observe the improvement in both precision and recall of all the belief classes. However, the improvement of ROB and NCB, which rarely occur in this corpus, is significant. We compared the results of these two categories in other experiments and noticed that adding the contextual representation feature (CR) has played a vital role in this improvement as without CR the F-Measure of NCB and ROB in $\boldsymbol{WE_{Char}}$+$\boldsymbol{WE_{pre}}$ experiment were 41.45 and 32.3 respectively.

Train	Test	F-Measure
NW	NW	57.96
DF	DF	52.03
Mix	NW	58.53
	DF	**67.38**

Table 4.5: The results of single-genre and multi-genre experiments separating out DF and NW datasets in $\boldsymbol{LDC_{Event}}$ corpus .

4.4.4.3 Multi-genre LCB

Table 4.5 illustrates the performance on the LDC events task when using within genre training and test vs mixing the genres of the data in a single task, with the best experimental setting combining character, pretrained and contextual encoding, framework. Mix is a combination of the two genres DF and NW.

The best performance is obtained by the multi-genre Mix condition which is trained

on the combined dataset. It achieved 67.38% and 58.53% F-Measure on DF and NW respectively. The mixed condition also did remarkably better on the DF data improving results by almost 16% absolute 52.03%. However, NW F-Measure improvement is only a marginal 0.57% absolute. This suggests that modeling knowledge from NW helps generalization to other genres such as DF, but DF is not informative for NW.

4.4.4.4 Impact of Auxiliary task on LCB Event Task

Table 4.6 illustrates the results of both the single task LCB Event versus the multi task learning framework proposed. In the single task setting there is no auxiliary task for GD. As shown, the joint task model (LCB+GD) achieves the highest F-Measure for the LCB task, yielding an F-measure of 61.53% compared to the 59.82% in the single LCB setting. Results in Table 4.7 show that NA and CB exhibit more improvement than ROB and NCB. This is probably attributed to the fact that NCB and ROB are extremely rarely observed in the training data. NCB and ROB tags make up only 4% of the total LCB tag annotations.

Task	Pr	Re	F-Measure
LCB (Single Task)	65.83	55.12	59.82
LCB + GD	64.27	**59.59**	**61.53**

Table 4.6: Results of employing the MTL on LCB Event task (LCB) and Genre Detection on $\boldsymbol{LDC_{Event}corpus}$

Tasks	CB	NCB	ROB	NA
LCB	65.03	16.67	4.88	47.64
LCB+GD	66.46	17.24	**9.76**	**50.15**

Table 4.7: Overall F-Measure results per LCB category by leveraging the MTL framework for LCB events task (LCB) and Genre Detection on $\boldsymbol{LDC_{Event}corpus}$

.

4.4.5 Summary

We show that employing a neural architecture for the problem of LCB tagging outperforms a conventional feature engineering approach. We also presented the first results on the challenging task of LCB tagging for events. We are planing to extend this work to other available datasets in this domain including the FactBank [Saurí and Pustejovsky, 2009] and LU [Diab et al., 2009]. Furthermore, our results show that applying the pre-trained word embedding perform better on multi-genre corpus than a single genre. Based on the experimental results, we assume that employing the pre-trained embedding on a homogeneous domain as training set could be useful in LCB problem. Hence, we will further investigate the impact of other pre-trained word embedding in multi-genre LCB problem. We also studied the impact of genre on LCB, and our studies showed that genre detection could be an auxiliary task in belief problem. We are interested to examine other auxiliary signals from homogeneous tasks that can help us specially on rarely occurring cases like ROB and NCB.

4.5 Single-step versus Joint LCB

In this section we examine and compare the single-step and two-step pipeline for the LCB tagging problem. Our ultimate goal is to device a new tagging model, that given a sentence and the events identifies the level of committed belief for the events.

4.6 Experimental Setup

In this section, we explicitly discuss the experimental setup of different experiments.

4.6.1 Datasets

In all the experiments, we use the same datasets for training and evaluation provided in previous section. Table 4.2 illustrates the statistics of different LCB categories on

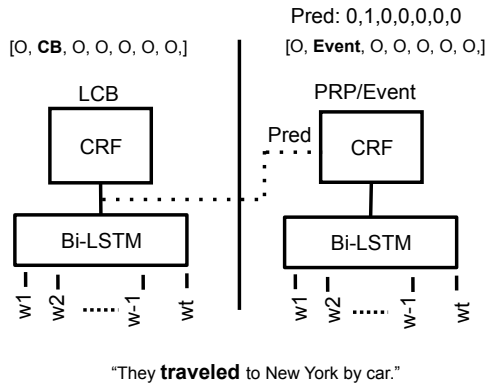


Figure 4.3: The different sections of the proposed two-step framework for LCB tagging illustrated here. The right network is a binary sequence tagging network for Proposition (prp)/Event identification task. The left network gets the prediction output of the prp/Event identifier model. Both networks using the concatenation of Character embedding trained by Bi-LSTM and Glove pre-trained word embedding.

	LDC_proposition							LDC_event						
Approach	**CB**	**NA**	**NCB**	**ROB**	**F**	**P**	**R**	**CB**	**NA**	**NCB**	**ROB**	**F**	**P**	**R**
Baseline	65.15	63.3	7.01	2.09	59.02	63.4	58.06	-	-	-	-	-	-	-
single-step	**79.42**	**78.14**	48.28	38.16	**75.24**	**76.69**	75.33	71.44	41.48	19.61	10	**56.95**	**68.88**	49.6
Two-Step	77.01	71.12	**48.93**	**41.56**	74.15	72.78	**76.37**	56.72	**42.25**	**23.73**	**21.28**	52.62	54.66	**51.08**

Table 4.8: Results per LCB categories by applying single-step and two-step approaches on LDC_prp and LDC_ev corpora.

both $\boldsymbol{LCB_{Proposition}}$ and $\boldsymbol{LDC_{Event}}$ training set.

4.6.2 Experiments

We investigate the effectiveness of the proposed two-step method on $\boldsymbol{LDC_{Proposition}}$ and $\boldsymbol{LDC_{Event}}$ we also compare the proposed approaches in this work with the current state-of-the-art method on LCB task. We devise the following experimental conditions:

4.6.2.1 Baseline: State-of-the-art Performance

[Prabhakaran et al., 2015] trained an SVM (Quadratic Kernel) using a wide range of lexical, syntactic, and hedging features. We compare our work to their since they provide state-of-the-art performance and show a strong baseline. For a fair comparison, we also use the same training, development, and evaluation set as in [Prabhakaran

Method	P	R	F
single-step	69.72	50.03	57.61
Pred Ev+PRP	51.44	40.52	44.99
Gold Ev+PRP	**81.31**	**70.89**	**75.5**

Table 4.9: Leveraging larger dataset LCB_prp on LCB_event using a two-step approach. F (Beta=1)-measure as the harmonic mean between (P)recision and (R)ecall.

et al., 2015].

4.6.2.2 Single-step VS. Two-step Method

The primary reasons behind this experiment are to assess the performance of the proposed Two-step method versus a single-step method. In the single-step method, we train a model using a 5-way classification method, while in the Two-step method, we first train a sequence tagging model to identify proposition or event and then in another step classifying the LCB categories using the predicted information of the primary model. We conduct all the experiments including the current state-of-the-art on two corpora.

4.6.2.3 Two-Step Approach for LCB_event

One of the main motivations of the two-step approach is to share the component between different tasks which share similar annotation definition. In LCB problem, We leverage the larger dataset (LCB_prp) and testing on smaller LCB_event dataset. LCB_prp and LCB_event have different span definitions but sharing similar LCB annotation. We conduct two experiments and compare the results with a single-step model trained on LDC_ev. 1) Ev+PRP in which the binary event detection model trained on LDC_event data and coupled with a LCB_prp tagging model to test on LCB_event. b) Gold ev+prp in which the actual gold event label of the data used with LCB_prp tagging to test LCB_event. The idea behind this experiment is to

test, given an ideal event detection model how well the model performs using the LCB_prp as the classifier for LCB_event.

4.6.3 Results and Evaluations

Table 4.8 showing the overall and per category performance of different approaches on LDC_event and LDC_prp corpora. According to the table, applying a single-step approach to LCB problem lead to better performance in comparison with the Two-step approach. However, the two-step approach lead to better performance per category for the NCB and ROB in LDC_prp and LDC_event. We investigated this phenomenon and noticed that most of the errors caused by unidentified events from the first step of the two-step approach. We further studied the performance of Two-step VS. single-step in -Table 4.9 and showing if a model can be trained and shared between datasets which have similar annotation definition. According to the table, a model which was using the gold event location with LCB_prp model outperforms a single-step method by more than 15%. This proves our primary assumption to leverage datasets with similar annotation definition in order to gain better overall performance.

4.6.4 Error Analysis

We manually analyzed the prediction of the prp/event identification and LCB classification in the single-step and two-step experiments on the development set.-Table 4.10 showing list of the most common errors in prp/event identification and LCB classification. According to this table model verbs are one of the main challenges of the LCB task maybe because of the number of times they appear as a non-event is significantly higher than an event. Another interesting finding is that the number of false positive cases for modal verbs significantly drop using the two-step learning method.

	List of common misclassified words
PRP	What, what, Is, All, Work, Spending, be, tax,sorry
EVENT	Saying, said, depositors, made, say, take, deposit
LCB	can,make,could,have,want,Considering, need

Table 4.10: List of top N errors for each task in single-step and two-step method on the two corpora-

4.7 Summary

In this chapter, we discuss the new neural network frameworks for level of committed belief (LCB) sequence tagging component. Our hypothesis is that the information of genre could be correlated with different level of formality of language and correlated with belief, and subjectivity. For example, people language in the discussion with informal level of lanaguage are more opinionated and show a lower level of committed belief in comparison to the news wire domain, which is more subjective with a high degree of belief. We proposed a multitask learning architecture by leveraging genre detection as the auxiliary task for LCB. The analysis showed that leveraging the genre information yielded better performance in the LCB task. It is was an interesting observation to see the news wire data showing more degree of reported belief and committed belief as opposed to the discussion forum, which is more toward Non-committed belief and interrogative. We experimented the impact of a single step and two step pipeline in this problem. In the first task, we conducted the event/Proposition detection and belief classification at the same time, but in two-step experiment, we separate the tasks, and the output of the first task is used as an input to the second task. We observe from the results that the single-step approach performs slightly better than the two-step approach, however, applying two-step approach add new functionality to the LCB tagger which is limiting to tagger to only identify the degree of belief for the given event or proportions.

Chapter 5: Detecting Rumors on Early Stages in Social Media

5.1 Introduction

The popularity of social media as a main source of news is irrefutable. According to the Pew research center, 68% of the US adult population in 2018 relied on a social media platform to seek news online [1]. However, unlike other media outlets, social media lacks supervision, and the abundant social media platforms serve as fertile ground for the emergence of rumors in the content. Our definition of a rumor is a statement that has not been verified at its posting time. The statement could turn out to be true or false, or it could even remain unverified [Qazvinian et al., 2011b, Castillo et al., 2011]. Propagation of rumor could cause widespread damage [Weeks and Garrett, 2014], potentially harming individuals or even endangering the lives of millions of people in a society. Therefore, identifying and intervening against the spread of rumors in their early phases of proliferation is vital to mitigate any consequent negative events. In this work, we studied the behavior of rumors on social media and investigate rumor and nonrumor events in different time frames of propagation. Our main hypothesis in this work was that rumors possess unique linguistic and metalinguistic characteristics that make them separable from nonrumor data. In fact, in rumor-laden conversations we noted different degrees of belief in comparison to nonrumor conversations; e.g., the source tweets in rumor-laden conversations tend to rely more on reporting events than do nonrumor cases. Accordingly, we investigated this hypothesis by studying the content of tweets. We leveraged a rich set of pragmatic features such as subjectivity, readability, and level of committed belief. Moreover, we further examined typical linguistic and metalinguistic features used to study rumors such as network characteristics [Seo et al., 2012b, Liu and Xu, 2016], linguistic

[1] https://www.journalism.org/2018/09/10/news-use-across-social-media-platforms-2018/

content [Vosoughi, 2015b, Qazvinian et al., 2011b], user provenance [Zubiaga et al., 2016a, Hassan et al., 2018], and social media features [Yu et al., 2019]. Assuming a conversational setup, we studied the importance of the different types of tweets in such conversations, e.g., Reply, Response, and Re-tweet. Our analysis confirmed the significance of the proposed list of features in various experiments.

5.2 Related Work

One of the key and primary challenges in studying rumor is to establish a sound meaning of this phenomenon. In this work, we emphasized our adherence to the widely accepted meaning of rumor as an unverified piece of information in circulation, which may turn out to be true or false. The phenomena of rumor have been widely studied from different perspectives, ranging from the psychology [Rosnow and Foster, 2005, Ajmal et al., 2018], to political communication [Harsin et al., 2006b],to social media [Zubiaga et al., 2016b], and so forth.

There are several studies about the behavior of rumor on the social media, but the works most related to ours have addressed this problem from Natural Language programming (NLP) and social media perspective. For example, one of the earliest works was reported in Qazvinian et al. [2011b]; it addressed a rumor-tracking task on longstanding rumors (rumor/not-rumor/undetermined) on Twitter by using the content-based as well as the microblog-specific meme features. In another study, [Castillo et al., 2013] leveraged both the content information and the information cascade feature by applying a supervised mechanism to identify credible and newsworthy content. In a recent study, [Vosoughi, 2015a] proposes his two-step rumor detection and verification model on the Boston Marathon bombing tweets. [Vosoughi, 2015a] proposed the hierarchical clustering model in addition to content-based and user-specific features. He proposed the speech act features as a new pragmatic attribute for identifying rumor.[Zubiaga et al., 2016a] is the work closest to ours that proposed

a new data set for this task, leveraged context preceding a tweet with a sequential classification pipeline, and showed that preceding context exploited as a sequence can considerably boost the rumor detector's performance on the conversation level. Later [Zubiaga et al., 2016b] extended the first data set and in the further analysis showed that true rumors resolved more quickly than false or unverified rumors. According to that work, the rumors in their unverified stage show significant burst regarding the number of re-tweets in comparison to rumors proven as true or false. Later [Kotteti et al., 2018] proposed the time series approach and studied the significance of different machine learning schemes by using only the temporal feature of the tweets in the conversation. Overall, most of the related work on the rumor identification task used network propagation, or content-based features in combination with user–related, or temporal features. To add to the current works, we propose a new set of content-based features including the level of committed belief, subjectivity, and readability. Beyond that, we propose a novel feature-extraction theme to encode the information from different time series of information diffusion. We also update the previous works by designing a novel neural network pipeline. The key advantages of our approach are

• less dependence on the context or event and greater robustness in unseen rumors.

• The significance of the proposed list of features in different time frames of rumor discussion

• Using only the certain type of tweets in the rumor identification task, which not only reduces the complexity but also improves the performance of the model.

• State-of-the-art performance in identifying rumor in early stages of rumor diffusion.

5.3 Approach

The main goal of this work was to identify rumor on Twitter at the conversation level. We leveraged the PHEME data set, which is a standard data set of rumor and

Events	#Threads	#Tweets	Rum	NRum
Charlie Hebdo	2,079	38,268	458	1,621
Sydney siege	1,221	23,996	522	699
Ferguson	1,143	24,175	284	859
Ottawa shooting	890	12,284	470	420
Germanwings-crash	469	4,489	238	231
Total	5,802	103,212	1,972	3,830

Table 5.1: Statistics for the PHEME dataset on Rum/NRum for each of the 5 events.

nonrumor conversations collected from Twitter. Each thread consists of multiple replied tweets. We first studied the significance of the proposed set of features from both the source and the corresponding conversation and then apply the proposed features in a supervised machine learning framework toward the task of rumor detection.

PHEME Dataset: PHEME Data set: The PHEME data set comprises annotations corresponding to rumors (Rum) and nonrumors (NRum) threads pertaining to five events [Zubiaga et al., 2016a]: the shooting at Charlie Hebdo, the Ferguson unrest, the crash of a German-wings plane, the hostage situation in Sydney, and the shooting in Ottawa. Overall statistics for the data set are given in Table 5.1.[2]

5.3.0.1 Feature

It was imperative for us to understand what was going on in different layers of the conversation. Hence, a new set of features was designed in four different categories to capture a different range of information from the source and replied tweets in the conversation. Table 2 provides a list of the proposed categories in this work.

5.4 Rumor Analysis

Our main goal in the rumor analysis section was to study the behavior of rumor and its spread by leveraging the proposed set of features in different time series of rumor

[2]The extended version of the PHEME dataset includes four other stories, but we only use the first five events which have relatively higher number of annotated data.

propagation. This section describes the preprocessing, feature extraction and analysis of the applied features in the experiments.

5.4.1 Pre-processing

In the preprocessing phase, the types of tweets in Twitter conversations were distinguished R = {Reply; ReTweet; Response}. The second step of preprocessing was to separate tweets in different time frames using the temporal information of the reacted tweet to the source of conversation.

5.4.1.1 Tweet Type

The tweet type could be an informative signal of the source event popularity, newsworthiness, and also legitimacy [Metaxas et al., 2014, Wu and Shen, 2015].
Using the Twitter API specification [3] about the tweet types, we defined each type as follows: Reply Mention, if the tweet mentions more than one user from the tweet conversation, including the source of the tweet (exp. Reply to @user1 @user2 I agree....); Re-tweets, in which the author reposts (RT @user ....) the source tweets with or without some modification; and Response, when the tweet does not mention any other user in the content and only leaves some comments.

5.4.1.2 Time-Series

In the next step, we divided reacted tweets in each conversation into different time series. We considered the time gap of the reacted tweets with respect to the source of conversations (the time gap indicates the time difference between the reacted tweets and the source of the conversation, calculated by using the created_time attribute of the tweets), and we set the time intervals $\boldsymbol{T = \{0, 6, 20, 80\ minutes, 3days\}}$. For each Event $\boldsymbol{E_i}$ in the PHEME dataset that consists of $\boldsymbol{C_i}$ rumor and nonrumor

[3] https://help.twitter.com/en/using-twitter/retweet-faqs

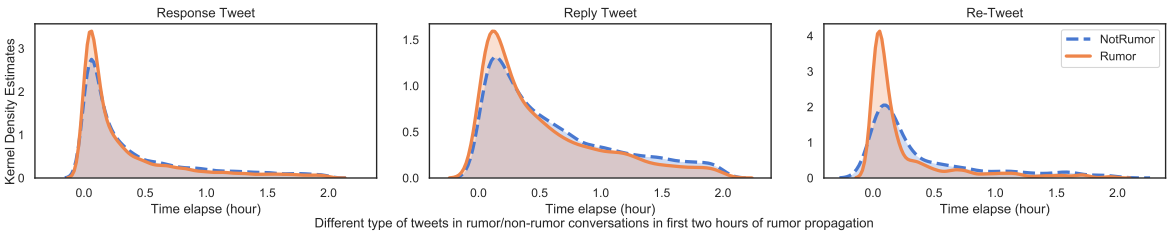


Figure 5.1: This plot shows, the KDE of different types of tweets rumors versus nonrumor conversations. The x-axis is indicative of the response time

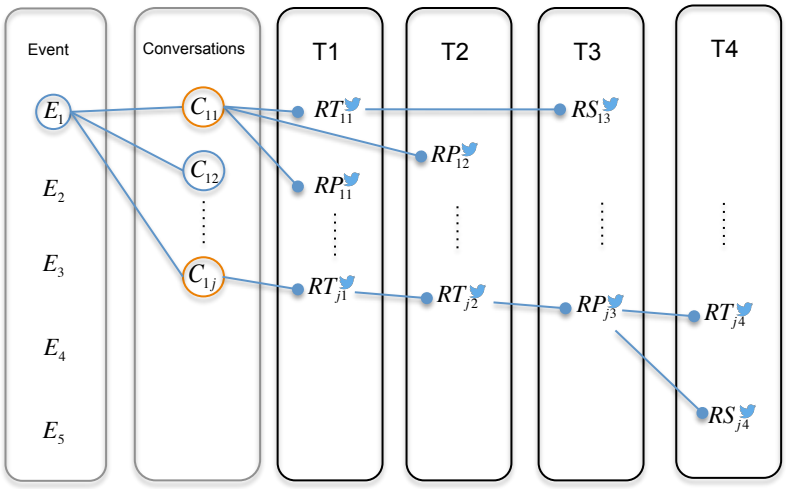


Figure 5.2: The time-series model for feature extraction of different events ($\boldsymbol{E_i}$) and subsequent rumour and non-rumour conversations ($\boldsymbol{C_i}$) on three types of tweet including the Reply (RP), Response (RS), and Re-tweets (RT).

conversations we have $\boldsymbol{Tweet_{ij}}$ where $\boldsymbol{i}$ is associated with the time series index in $\boldsymbol{T}$ and $\boldsymbol{j}$ associated with the tweet type index in $\boldsymbol{R}$.

Figure 5.1 shows the density estimation of different tweet types in the first two hours of rumor propagation, which retained more than 75% of the tweets posted in all the events. By looking at the KDP, it is observed that the Re-tweet ratio (M-W test = 2585[4], p 0.0) in rumor conversation was significantly higher than nonrumor conversations, and was similar for Reply (M-W test = 9507, p 0.0) and Response tweets (M-W test = 9317, p 0.0).

[4]Mann-Whitney U statistic [Mann and Whitney, 1947]

5.4.2 Features Extraction

Figure 5.2 demonstrates the timeseries scheme which is designed to derive features for each conversation of the events. The associated conversational-based vectors $\boldsymbol{C_i}$ for each event is built from the following extracted features.1) the source of the conversations. 2) different types of tweet associated with the source of the conversation in each time frame $\boldsymbol{T_{fi}}$. e.g., $\boldsymbol{T_{f2}}$ indicates the time frame between 6 to 3o minutes and $\boldsymbol{F_{rpf2}}$ is related to the features extracted from the replied tweets in the same time frame of the certain conversation. 3)different type of tweets cumulatively $\boldsymbol{T_{ci}}$ from the time the source tweet was posted until that time frame. e.g., $\boldsymbol{T_{c3}}$ indicates the time between zero and 30 minutes and $\boldsymbol{F_{rpc2}}$ is related to the features extracted from the replied tweets in the first 30 minutes of the conversation.

5.4.2.1 Feature Categories

5.4.2.2 Network

Rumors travel faster and deeper in information cascades [Vosoughi et al., 2018] and understanding the flow and transition of information in the conversation can help decipher their behavior. Network features are designed to code propagation information of the rumor in the information tree [Zeng et al., 2016, Wu and Shen, 2015]. All the features in this category were extracted from the reacted tweets in each conversation. For a given conversation, the network feature was calculated by obtaining the average of the values of the listed features for each tweet type in a certain time frame. For instance, for a conversation that gets 100 Retweets in the first 6 minutes (0.1 hour), the value of the react rate for Re-tweets $\boldsymbol{f_{rt1}}$ is Log(100/0.1), or similarly, for the max depth feature of the Replied tweets in the first 30 minutes, $\boldsymbol{f_{rp2}}$, we measured the maximum depth among all the replied tweets in that conversation.

Category	*Where it came from*
Network	
Time Gap	The time deference between the replied and source tweet
Level	The level of the tweet in the conversation tree in certain time
Max depth	The max depth of the conversation in the certain time
React rate	Log of the rate of reactions in a certain time frame.
User provenance	
Description	If user-defined string describing their account.
Verified	When true, indicates that the user has a verified account
Followers	Log(The number of followers this account currently has)
Friends	Log(The number of users this account is following)
Listed count	Log(The number of public lists that this user is a member of)
User URL	User profile URL other than Twitter
Statues Count	Log(The number of Tweets issued by the user)
Tweeting Pace	Tweeting ratio since the created time of the account
Profile age	Age of the account
Social media specific	
Hashtag	The existence of hashtag
URL	The existence of non Twitter URL
Top 500 URL*	If URL is one of the top 500 URL
URL Credibility*	URL credibility label as fake, bias, satire, or ...
Media Existence	Represents a single photo, video or animated GIF
Content	
Linguistic Content	The n-gram content Representation
Pragmatics	
Coarse terms*	Number of Coarse terms
Subjectivity*	The subjectively score of the post
Sentiment	The sentiment score of the post
Readability*	The readability score of the post
Belief*	The LCB score on (CB, NA, ROB, and NCB)

Table 5.2: Different category of features.

5.4.2.3 User Provenance

In this category of features, we used the metadata of the user profile that was only applicable on Twitter data. The main purpose of this category is to measure the user popularity and influence on social media. We extracted user provenance features for all the people in the conversation who gave a Re-tweet, Reply, or Response to the source of conversation. We coded all the information of the people who were involved in the conversation in different time frames of rumor propagation.

5.4.2.4 Social Media Specific

Twitter users rely on hashtags, media, or URLs to describe a topic or a theme or for relating to other resources to support their opinion. We updated the proposed features in this category [Zeng et al., 2016, Hamidian and Diab, 2015] by checking the legitimacy of the URLs inside the tweet content. We checked whether URLs were among the top 500 most popular domains (excluding Twitter) by the time the conversations had all been posted and also tagged each URL as fake, satire, conspiracy, reliable, political, unreliable, clickbait, hate, junksci, and state, using a corpus of prelabeled domains [5].

5.4.2.5 Linguistic Content

Linguistic content–based features were extracted from the raw text of the tweet. We removed all the hashtags, URLs and user mentions. The linguistic content features are mainly derived from the words n-gram (n=1,2). We used the TF-IDF value of each n-gram for the source of the tweet in the conversations. It is worth noting that this feature was applied only to the source of conversations

Pragmatic features; These were designed mainly to identify the subjectivity, readability, sentiment, and belief of the people in the discussion regarding the particular proposition. We assumed that the belief of the people concerning the source of the conversation is a strong signal to identify rumorous conversation. Some of the proposed features in this category were inspired by previous works [Vosoughi et al., 2018, Castillo et al., 2011, Hamidian and Diab, 2016, Zubiaga et al., 2018] that discussed the impact of certainty, factuality, sentiment, and speech act events in rumor phenomena. We introduced the level of committed belief (LCB), which measures the writer's degree of belief about the proposition or event. We leveraged the LCB tagger model proposed by [Werner

[5]https://github.com/several27/FakeNewsCorpus

et al., 2015b] for this purpose. The LCB tagger is a sequence tagger that identifies the event in the text and tags the level of committed belief as committed belief (CB) when the writer strongly believes in the proposition, noncommitted belief (NCB) when the writer reflects a weak belief in the proposition, and nonattributable belief (NA) when the writer is not (or could not be) expressing a belief in the proposition (e.g., desires, questions, and so forth), and reported belief (ROB) when the writer uses event reporting, such as said, told, according to, and so forth, to report on someone else's behalf (e.g., CNN reported that the election was a fraud). In order to extract the LCB on the tweet level we used the following formula:

$$\boldsymbol{LCBScore_i} = \frac{\boldsymbol{\#LCB_i}}{\boldsymbol{\#Event}} \tag{5.1}$$

Where $\boldsymbol{i}$ is the index of the LCB tag in $\boldsymbol{\{CB, NA, ROB, NCB\}}$ and # is the total number of cases in the tweet. For each LCB tag the score is a value between 0 and 1. The following example is selected from the Ottawa Shooting tweets.

Ottawa SUN: Man 42 years old, $\boldsymbol{LCB_3}$ **arrested** in April south Ottawa $\boldsymbol{LCB_3}$ **shooting**

Using the -Equation (5.1) in this example $\boldsymbol{LCB_3}$ corresponds to the ROB tag and is equal to 1; other LCB scores for CB, NA, and NCB are set to values close to 0.

HateSpeech - Curse Words; For hate speech we relied mainly on a lexicon of more than 1,300 offensive English terms5 and checked the existence of these terms in the source and replied tweets. [6]

Sentiment and Subjectivity; For the sentiment and subjectivity, we used the [Loria et al., 2014] API. This tool returns the sentiment (polarity, subjectivity) score at the

[6]https://www.cs.cmu.edu/~biglou/resources/

sentence level. The polarity score is a float value range [-1.0, 1.0] and the subjectivity is in the range [0.0, 1.0] where 0.0 is very objective, and 1.0 is very subjective. High subjectivity indicates mostly that it is an opinion and not a piece of factual information.

Readability Readability is another modality that we discuss in this problem. Readability conveys the degree of complexity of the language for the reader; this feature has been recently addressed in online sharing to estimate the number of shares for fake and real information [Hosseini et al., 2019]. Here, we discuss readability in identifying rumor and spread of rumor. For this purpose, we used the [Hüning, 2005] package that provides statistics from the text to determine the estimated school grade level required to understand the text readability in a document; a higher grade indicates the complexity of reading.

5.4.2.6 Insight from Pragmatic Features

Table 5.3 gives the statistics of the proposed pragmatic features extracted from the content of the source of conversations. Each column shows the average value of the corresponding feature for each class. According to the results, nonrumor tweets are more subjective than rumor tweets (D=0.119 P-Value 0.0). One interpretation of this is that rumors are more oriented toward being factual and nonrumor tweets are more toward being opinionated. As shown in -Table 5.4, this is also true to a lesser degree in the reacted tweets; as expected, the reacted tweets in both rumor and nonrumor conversations are more opinionated and subjective in comparison with the source, which is not surprising. Regarding the readability or ease of reading, rumor conversations are slightly easier to read than nonrumor conversations (8th grade versus 9th grade (D=0.078 P-Value 0.0)). Our assumption is that this could impact the

spread of rumor, as discussed in [Hosseini et al., 2019]

The comparison of sentiment between rumor and nonrumor conversations shows that rumor conversations were more neutral and nonrumor conversations about the same stories were more positive. This was valid also in the reacted tweets in the conversations, as reacted tweets to nonrumor conversations were more positive than to rumor ones (D = 0.029 P-Value 0.0). The average hate speech score of the source of conversation was not statistically significant, but the hate speech average in the reacted tweets to rumor was higher than to the nonrumor one. Reported belief occurs when a writer uses the reported proposition and conveys a low degree of certainty. According to the statistics, ROB in both rumor source and reacted tweets in rumor conversations was higher than in nonrumor ones; this was also observed in all the stories of the PHEME data set shown in Figure 3. CB or committed belief is another LCB feature that showed a high degree of belief of the writer and had higher value in rumor conversations than in nonrumor ones. Interestingly, the reaction to rumor tweets has a lower level of committed belief. From this could be inferred that the degree of belief in the source might impact the degree of belief in the reactions in rumor and nonrumor conversations. Considering the K-S test result, NCB or noncommitted belief is not a significant feature in this category. We attribute this to the fact that the number of features in the NCB category were not big enough across the data set. NA or nonattributable belief indicates desire, request, or questions. The NA score of the source of nonrumor conversations was almost two times more than that of the source of rumor conversations (D = 0.038 P-value 0.0). This was also valid for reactions but with a small difference.

5.4.3 Time-series Rumor Analysis

To analyze the importance of different category of features we applied the entire dataset for the training. For the $\boldsymbol{T_{i,j}}$ experiment we extracted all the in-frame and

	Mean		***Std***		***KS-TEST***
Feature	***Rumor***	***Non-Rumor***	***Rumor***	***Non-Rumor***	
Subjectivity	0.244	0.311	0.287	0.329	D=0.119 P-Value 0.0
Readiblity	8.817	9.363	2.7	2.91	D=0.078 P-Value 0.0
Sentiment	0.003	0.044	0.202	0.278	D=0.103 P-Value 0.0
Hate Speech	0.463	0.443	0.686	0.686	D=0.014 P-Value 0.95
ROB	0.088	0.026	0.213	0.12	D=0.110 P-Value 0.0
CB	0.719	0.673	0.37	0.39	D=0.055 P-Value 0.0
NCB	0.01	0.006	0.081	0.057	D=0.005 P-Value 0.9
NA	0.107	0.198	0.248	0.319	D=0.144 P-Value 0.0

Table 5.3: Descriptive statistics on pragmatic features from the source of the rumor and nonrumor conversations as well as K-S tests of the differences in the distributions of these measures. .

	Mean		**Std**		**KS-TEST**
Feature	**Rumor**	**Non-Rumor**	**Rumor**	**Non-Rumor**	
Subjectivity	0.345	0.358	0.338	0.338	D=0.023 P-value 0.0
Readability	5.748	5.99	3.8	3.74	D=0.031 P-Value 0.0
Sentiment	0.012	0.03	0.29	0.301	D=0.029 P-Value 0.0
Hate Speech	0.354	0.333	0.650	0.637	D=0.014 P-Value 0.0
ROB	0.0711	0.035	0.232	0.164	D=0.044 P-Value 0.0
CB	0.499	0.52	0.424	0.418	D=0.030 P-Value 0.0
NCB	0.0169	0.0153	0.097	0.09	D= 0.002 P-Value 1.0
NA	0.276	0.297	0.369	0.371	D=0.038 P-value 0.0

Table 5.4: Descriptive statistics on pragmatic features from the reacted tweets to the rumor and nonrumor conversations as well as K-S tests of the differences in the distributions of these measures. .

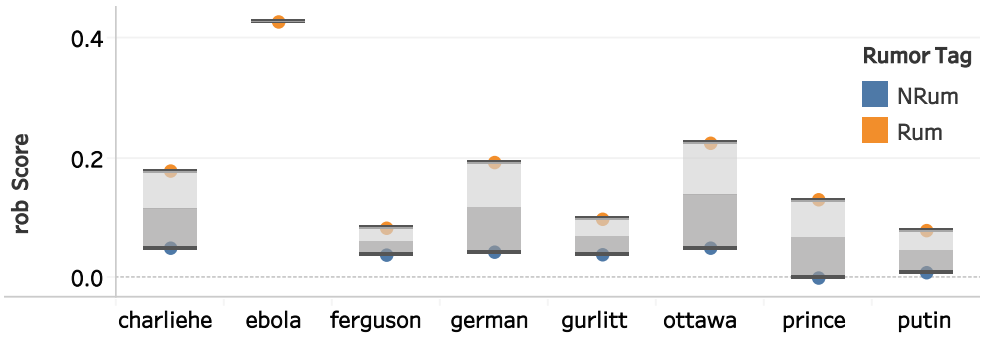


Figure 5.3: The degree of reported belief in the source of the stories of the PHEME dataset.

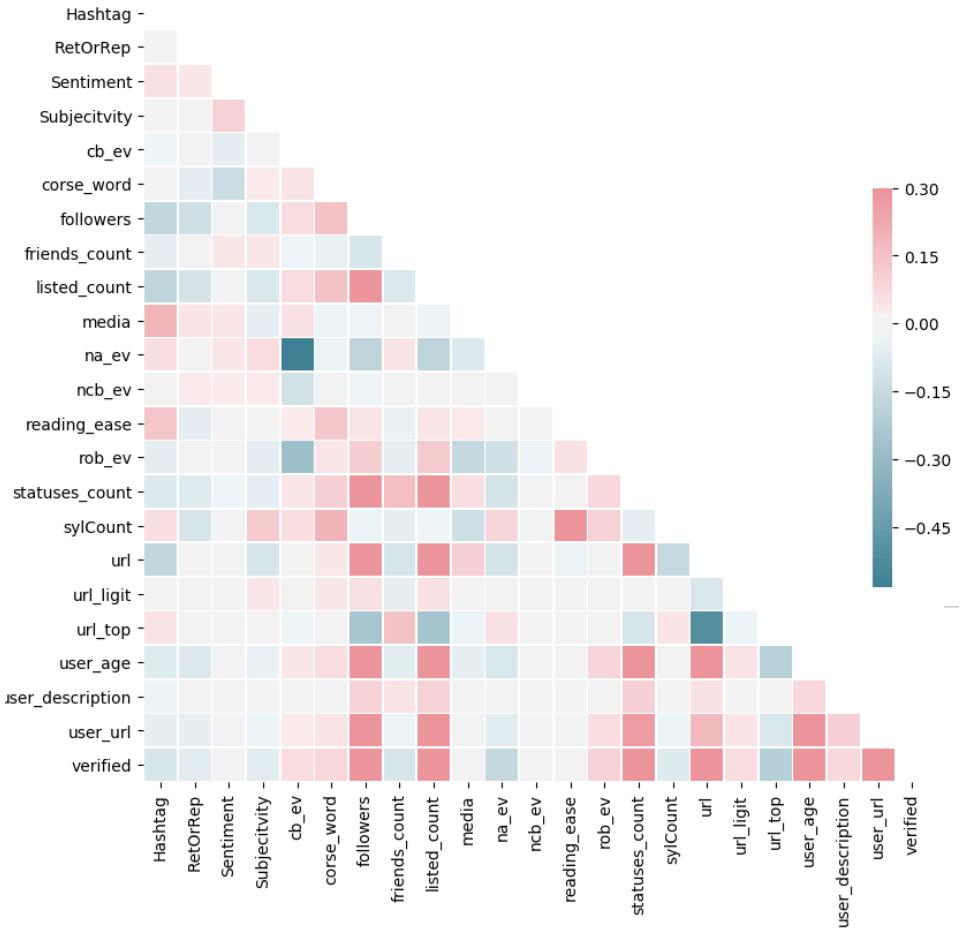


Figure 5.4: Correlation matrix of the proposed set of features in T=0 on the source of conversations.

cumulative features from $\boldsymbol{T_0}$, $\boldsymbol{T_0 i}$, $\boldsymbol{T_i j}$, and $\boldsymbol{T_0 j}$ time frames; e.g., for T = (6,30] experiment we used all the features from T = 0, T = (0,10], T = (6,30], and T = [0,30].

Feature	Coef.	Std.Err.	z	P>|z|	[0.025	0.975]
ROB	-1.9988	0.2002	-9.9829	0	-2.3912	-1.6064
NCB	-0.8525	0.4024	-2.1185	0.0341	-1.6413	-0.0638
URL	-0.4395	0.0676	-6.4968	0	-0.5721	-0.3069
Statuses_count	-0.4203	0.0539	-7.7912	0	-0.526	-0.3145
CB	-0.292	0.0963	-3.0314	0.0024	-0.4809	-0.1032
User_url	-0.1777	0.0847	-2.0995	0.0358	-0.3437	-0.0118
Readability	0.0696	0.0123	5.6515	0	0.0454	0.0937
Friends count	0.1377	0.0349	3.9408	0.0001	0.0692	0.2062
Followers	0.1391	0.0352	3.954	0.0001	0.0702	0.2081
Media	0.4978	0.0555	8.9657	0	0.389	0.6067
Hashtag	0.5813	0.0567	10.2588	0	0.4702	0.6923
Sentiment	0.5818	0.1091	5.333	0	0.368	0.7956
Subjectivity	0.6169	0.0863	7.1449	0	0.4477	0.7862
NA	0.672	0.126	5.3348	0	0.4251	0.919

Source of the conversation $\boldsymbol{T_1}$=0

-Figure 5.4 showing the correlation of the variables on the source of conversations in $\boldsymbol{T_1}$. For all the analysis we first control for confounding variable and then remove the correlated variables with more than 80% similarities. For feature selection we rely on Lasso [Tibshirani, 1996] and Feature ranking using the recursive feature elimination [Guyon et al., 2002] algorithm. For further adjustment of confounding variables and addressing of multicollinearity, we applied principle component analysis (PCA) with the same number of components as features. The output of PCA was fed to a logistic regression model for training. All the time series results given in Table 5 were filtered to demonstrate only the in-frame significant features based on 5% significant rate. We discuss the significance of different categories in the following subsections.

5.4.3.1 Interpretiblity

We compared the P-value, standard error, and z score of all the features in different experiments. In the features extracted from the source of conversations, ROB, URL, and status count were among the significant features in all five time frames. Considering

Feature	Coef.	Std.Err.	z	P>|z|	[0.025	0.975]
CB on Replied Tweets	0.7731	0.1987	3.8897	0.0001	0.3835	1.1626
Media In Re-tweets	2.0553	0.5197	3.9547	0.0001	1.0367	3.074
ROB in Reacted Tweets	-0.8229	0.2161	-3.8082	0.0001	-1.2464	-0.3994
Hashtag in Response	-1.3701	0.3872	-3.5389	0.0004	-2.1289	-0.6113
Hashtag in Replies	-1.3261	0.4079	-3.2513	0.0011	-2.1255	-0.5267
Hashtag in Reacted Tweets	1.5563	0.5077	3.0653	0.0022	0.5612	2.5514
CB in Reacted Tweets	-0.8281	0.2735	-3.0278	0.0025	-1.3641	-0.292
NA in Re-tweets	1.1371	0.3815	2.9805	0.0029	0.3894	1.8848
React Rate	0.2136	0.0727	2.9381	0.0033	0.0711	0.3561
NA in Replies	0.691	0.2393	2.888	0.0039	0.2221	1.1599
Hashtag in Replies	-1.2845	0.4461	-2.8794	0.004	-2.1588	-0.4102
Top 500 URL in Replies	-0.0011	0.0004	-2.8642	0.0042	-0.0019	-0.0004
Media in Reacted Tweets	-2.0748	0.7852	-2.6424	0.0082	-3.6138	-0.5359
CB in Response	0.6275	0.2419	2.5935	0.0095	0.1533	1.1017
Re-tweet Rate	-1.5301	0.5944	-2.5741	0.01	-2.6951	-0.3651
Reply depth	0.4909	0.1969	2.4928	0.0127	0.1049	0.8769
Na in Reacted Tweets	-0.7911	0.3334	-2.3732	0.0176	-1.4445	-0.1378
Response depth	0.868	0.3798	2.2855	0.0223	0.1236	1.6124
Media in Replies	1.425	0.644	2.2127	0.0269	0.1628	2.6872
Overall Depth	-0.947	0.4414	-2.1454	0.0319	-1.8122	-0.0818
ROB in Re-tweets	0.9193	0.4315	2.1306	0.0331	0.0736	1.7649

$\boldsymbol{T_{2f}}$,$\boldsymbol{T_{2C}}$=(0, 6 Minutes]

Feature	Coef.	Std.Err.	z	P>|z|	[0.025	0.975]
ROB in Response (F)	-0.8749	0.1659	-5.2741	0	-1.2	-0.5497
Uset Age Retweets (F)	-0.6417	0.2274	-2.8224	0.0048	-1.0873	-0.1961
Top 500 URL in Reply (F)	-0.0059	0.0016	-3.8078	0.0001	-0.009	-0.0029
Hashtag (F)	1.9528	0.5344	3.6541	0.0003	0.9054	3.0003
CB in Re-tweets (F)	2.1742	0.9084	2.3934	0.0167	0.3938	3.9546
CB in Re-tweet (C)	-2.7493	0.9768	-2.8146	0.0049	-4.6637	-0.8348
Hashtag in Reply (C)	-0.8651	0.3328	-2.5992	0.0093	-1.5174	-0.2128
Max Depth (C)	-0.081	0.0264	-3.0661	0.0022	-0.1328	-0.0292
Top 500 URL in Reply (C)	0.006	0.0016	3.7205	0.0002	0.0028	0.0091
User age in Retweets (C)	0.5169	0.2307	2.2403	0.0251	0.0647	0.969
Response Rate(C)	1.2842	0.2102	6.1083	0	0.8721	1.6962
Reply Rate (C)	1.4259	0.3265	4.367	0	0.7859	2.0658
Hashtag in Response (C)	2.0051	0.6521	3.0746	0.0021	0.7269	3.2833
Hashtag in Retweet (C)	-2.6752	0.6286	-4.2559	0	-3.9072	-1.4432

$\boldsymbol{T_{3F}}$=(6, 30 Minutes], $\boldsymbol{T_{3C}}$=(0, 30 Minutes]

Feature	Coef.	Std.Err.	z	P>|z|	[0.025	0.975]
Hashtag in Retweet (F)	1.4992	0.258	5.8108	0	0.9935	2.0048
ROB in Reply (C)	-2.6682	0.6796	-3.9264	0.0001	-4.0001	-1.3363
ROB in Retweet (C)	-1.8464	0.6895	-2.6778	0.0074	-3.1978	-0.4949
Hashtag (C)	-1.4344	0.3359	-4.2704	0	-2.0927	-0.776
ROB in Respons (C)	-0.9289	0.3048	-3.0476	0.0023	-1.5263	-0.3315
Hashtag in Reply (C)	1.2098	0.2116	5.7162	0	0.795	1.6246
ROB in all tweets (C)	1.2235	0.5714	2.1413	0.0323	0.1036	2.3434

$\boldsymbol{T_{4F}}$=(30, 90 minutes], $\boldsymbol{T_{4C}}$=(0, 90 Minutes]

Feature	Coef.	Std.Err.	z	P>|z|	[0.025	0.975]
Depth in Replies (F)	-0.2312	0.0307	-7.5297	0	-0.2914	-0.171
Hashtag in Replies (F)	1.9034	0.7658	2.4855	0.0129	0.4025	3.4044
Sentiment in Retweets (F)	2.8389	0.7283	3.8979	0.0001	1.4114	4.2664
Depth Replies (C)	0.4289	0.0631	6.7983	0	0.3052	0.5525

$\boldsymbol{T_{5F}}$=(90, 3 days], $\boldsymbol{T_{5C}}$=(0, 3 days]

Table 5.5: The Logistic regression models estimating the likelihood of a source of conversation being Rumor (0) or Non-Rumor (1) by using the variables collected from different type of tweets in different time frames $\boldsymbol{T}$=[0, 6, 30, 90 minutes, 3 days] (In-Frame (F) and Cumulative (C) features filtered by time $\boldsymbol{T_i}$ and by the significance at 5% level)

the coefficient sign, all these features were correlated with rumors. subjectivity, media, NA score, and the existence of a hashtag in the content were the other features, which extracted from the source tweets and were correlated with the nonrumor class. Among the feature extracted from $\boldsymbol{T_2f}$ and $\boldsymbol{T_2c}$ the hashtag existence in response tweets and the existence of top 500 URLs in the replied tweets are significant and both of them are correlated with being rumor. Looking at the URLs 'BBC,' 'CNN,' 'Youtube,' 'CBC,' and 'WSJ' were among the top five URLs used in replied tweets. we assume that people use legitimate URLs to support their response to the source of rumors more than in nonrumor conversations. Among the features extracted from $\boldsymbol{T_3f}$ and $\boldsymbol{T_3C}$ the most significant features were the existence of a hashtag in Re-tweets, top 500 URLs in replies, hashtag in replies, and ROB in all types of tweets are the most significant features. From the features extracted from $\boldsymbol{T_4c}$ and $\boldsymbol{T_4f}$, ROB in all type of tweets from the Response tweets is significant. This could be interpreted that in the early hours of rumor propagation people start using the reported verbs to support or deny the source of conversation. In other words, rumor conversations are more likely to trigger people using reported belief in their language. e.g., *I just read that France Interior minister* $\boldsymbol{<ROB>}$ ***said*** *operation to detail Charlie Hebdo suspects is now underway northeast of Paris.*

From the significant features of $\boldsymbol{T_{4c}}$ and $\boldsymbol{T_{4c}}$ the ROB in replied tweets and the existence of hashtags in all the tweet are correlated with rumors, whereas ROB in all the tweets posted since T4c is correlated with being nonrumor. In the last round of experiments for T5, the depth of replies is correlated with rumor and the depth of all tweets is correlated with nonrumor cases.

Moreover, sentiment in Re-tweets and hashtags in replies are correlated with nonrumor cases. The summary of the top features is listed in Table 5.6. Table 5.7 shows the significance of each category in different time frames. Each check mark indicates that there is a feature from that category that was significant in that time frame.

Top Features	Rumor tag
ROB in time frame $\boldsymbol{T_1}$	R
URL in time frame $\boldsymbol{T_1}$	R
Statuses_count in time frame $\boldsymbol{T_1}$	R
Subjectivity in time frame $\boldsymbol{T_1}$	NR
Media in time frame $\boldsymbol{T_1}$	NR
NA in time frame $\boldsymbol{T_1}$	NR
Hashtag in time frame $\boldsymbol{T_1}$	NR
Hashtag in response tweets in time frame$\boldsymbol{T_{2f}}$	R
Top 500 URL in Replied tweets in time frame $\boldsymbol{T_{2f}}$	R
Top 500 URL in Replied tweets in time frame $\boldsymbol{T_{3f}}$	R
Hashtags in Replied tweets in time frame $\boldsymbol{T_{3c}}$	R
Top 500 URL in Replied tweets in time frame $\boldsymbol{T_{3c}}$	NR
Hashtag in time frame $\boldsymbol{T_{3f}}$	NR
ROB in all the tweets in time frame $\boldsymbol{T_{3c}}$	R
ROB in Replied tweet in time frame $\boldsymbol{T_{3c}}$	NR
Hashtag in Re-tweet in time frame$\boldsymbol{T_{3f}}$	R
ROB in time frame $\boldsymbol{T_{4c}}$	NR
Hashtag in time frame $\boldsymbol{T_{4c}}$	R
ROB in Replied tweets in time frame $\boldsymbol{T_{4c}}$	R
Depth of tree in replied tweets in time frame $\boldsymbol{T_{5f}}$	R
Depth of tree in time frame $\boldsymbol{T_{5c}}$	NR
Sentiment in Re-tweet in time frame $\boldsymbol{T_{5f}}$	NR
Hash in Replied tweets in time frame $\boldsymbol{T_{5f}}$	NR

Table 5.6: List of top features in time-series experiments for different tweet types with corresponding rumor (R) and non-rumor (NR) tags.

Considering the data in Table 5.5 and Table 5.7 platform-specific and pragmatic features were the top two most significant categories, whereas the user popularity and network features were more effective in early time frame experiments.

5.5 Rumor Detection

In order to evaluate the effectiveness of the proposed time-series methodology and novel set of features, we compared our work with other approaches that used the same data set in identifying rumor.

Feature Category	Time Frame																	
	src	**6**		**30**				**90 minutes**					**3 Days**					
	F	*src*	*F*	*src*	*6*	*F*	*C*	*src*	6	*30*	*F*	*C*	*src*	*6*	*30*	*90*	F	C
Network			✓		✓		✓			✓						✓	✓	✓
User provenance	✓	✓		✓	✓	✓	✓	✓	✓	✓			✓					
Platform Specific	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
Pragmatic	✓	✓	✓	✓	✓	✓	✓	✓		✓		✓	✓		✓	✓	✓	✓

Table 5.7: The overall list of significant source (src) in-frame (F) and cumulative (C) features from different time-series.

5.5.1 Experimental Design

5.5.2 Baseline

We compared the best performance achieved in this work with [Zubiaga et al., 2016a] which proposed the data set and also performed the rumor-identification task by exploiting the content and source features and using the conditional random field (CRF) as the classifier. We also replicated another feature-based approach [Hassan et al., 2018] which achieved state-of-the-art performance in rumor tracking on the same data set by applying the set of source and content features and applying the random forest model as the classifier.

5.5.3 Experiments

We conducted multiple experiments to evaluate the prediction power of the proposed methodology in identifying rumor. We designed a setup similar to [Zubiaga et al., 2016a] and simulated a realistic scenario in which an event is completely unknown. We performed a 5-fold cross-validation experimental setting, in which each step had four of the events for training; the remainder event was used for testing. For evaluation purposes, all the output of all five runs was aggregated, and the microaveraged evaluation was used across runs. We trained and tested the effectiveness of the proposed set of features in different time series as well as in different tweet types. We went beyond that to study the in-frame and cumulative feature extraction in all the

experiments.

5.5.3.1 Conventional Classifiers

We compared different machine learning models including Random Forest (RF), Naive Base (NB), Multi_layer Perceptron (MLP), Support Vector Machine (SVM) [Chang and Lin, 2011], and AdaBoost classifier (ADA) [Freund and Schapire, 1997]. We used the scikit-learn [Pedregosa et al., 2011] Python package for all the experiments.

5.5.3.2 Neural Network Classifier

Given the success of neural networks and specifically the transfer learning for a variety of NLP tasks, we examined the impact of applying transfer learning in a neural network pipeline for rumor identification. Figure 5.5 shows the proposed neural architecture, which consists of five key components. First is the BERT pretrained [Devlin et al., 2018] embedding layer; we leveraged pretrained BERT and aimed to fine-tune the language model on all the tweets of the conversations. It is noteworthy that we reset and repeated fine-tuning the model by excluding the test set from the data set in the cross-validation experiments. In the second layer of the proposed architecture, we integrated and tested the effectiveness of two classifiers, including a recurrent neural network and long short-term memory (LSTM) [Hochreiter and Schmidhuber, 1997] networks, in particular, and convolutions neural networks (CNN). The third layer of this network is the dropout layer. This layer ignores some of the connections of the neurons, which results in less overfitting. We tested the effectiveness of adding the time-series feature in the different layers of this network. In the first experiment, we concatenated the BERT representation of the tweets with the time-series feature, and in another experiment, we tested by adding the features before the classification layer. The last key component of this architecture is the classification layer, in which the output of the dropout layer in addition to the time-series hot vector are reshaped into

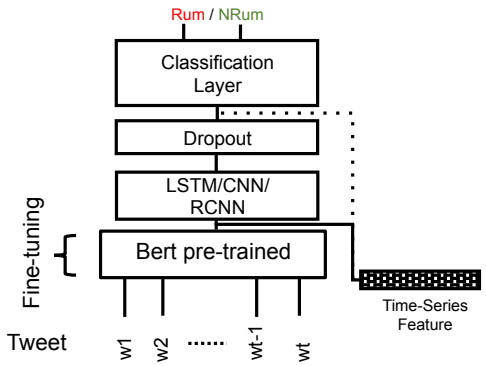


Figure 5.5: Neural Network architecture for rumor identification task

a logits tensor of batch size $\boldsymbol{X} * \mathbf{2}$, for which each tweet is given a likelihood of being classified as a rumor (Rum) or nonrumor (NRum). The probabilities are summed up to 1 by using the Softmax function. (The experiments of the NN approach for rumor identification is still in progress)

Figure 5.5 represents the architecture we used for identifying rumor conversations on Twitter.

5.5.4 Evaluation

Table 5.8 presents the performance of different categories of features extracted in all time frames from different tweet types. The highest performance achieved was 0.721, by combining all the features. The second and third categories of features were linguistic content by 0.658 and 0.628, respectively. We compared different machine learning algorithms, as presented in Table 5.9. The best performance was achieved by using SVM with the radial basis function (RBF). The SVM model with combined features could outperform both baselines by almost 5%..

5.5.4.1 Rumor Type Evaluation

In Figure Figure 5.6, We illustrate the performance of using the features which are extracted from different types of tweet in different time frames. Applying the extracted

Feature Category	F-Micro	Precision	Recall
Combined	**0.721**	**0.721**	**0.721**
Pragmatic	0.628	0.628	0.628
Social Media	0.581	0.581	0.581
User Provenance	0.623	0.623	0.623
Network	0.583	0.583	0.583
Linguistic Content	0.658	0.658	0.658

Table 5.8: The Performance of the proposed list of features in Rumor identification task

Classifier	F-Micro	Precision	Recall
SVM	**0.721**	**0.721**	**0.721**
MLP	0.703	0.703	0.703
ADA	0.651	0.651	0.651
NB	0.645	0.645	0.645
Zubiaga et al. (CRF)	0.607	0.667	0.556
Hassan et al. (RF)	0.669	0.669	0.669

Table 5.9: The performance of different classifiers in rumor identification task

features from the re-tweets causing to the highest performance in all time frames. It is worth noting that the difference between the performance of using re-tweets is only slightly higher than the other tweet types.

5.5.4.2 Time-Frame vs. Cumulative Features

Figure 5.6 shows our comparison of the performance of using in-frame versus the cumulative features in different time frames. Our experiments showed that the in-frame feature led to better performance. However, the difference was only slightly better.

Feature Category	F1-NR	Precision-NR	Recall-NR	F1-score-R	Precision-R	Recall-R
Content	0.787	0.724	0.862	0.480	0.609	0.396
User provenance	0.665	0.747	0.599	0.530	0.459	0.626
Pragmatic	0.581	**0.805**	0.455	**0.569**	0.443	0.797
Social Media	0.576	0.798	0.450	0.564	0.439	0.790
Network	0.388	0.786	0.258	0.538	0.390	**0.871**
Combined	**0.809**	0.727	**0.912**	0.483	**0.695**	0.370

Table 5.10: The performance of each category of features on rumor and non-rumor cases

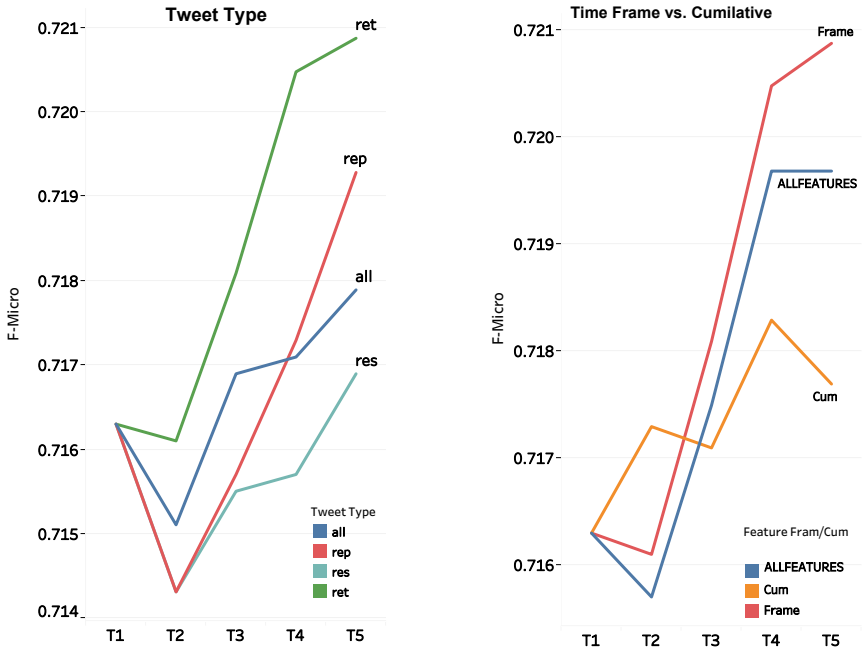


Figure 5.6: Tweet types, showing the performance of the model using the features extracted from different types, e.g., ret (Re-tweets), rep (Reply), res (Response), and all features combined. Frame versus Cumulative shows the performance of the model using the features extracted from related time frames, cumulatively, and all features, which is the combined set.

5.5.4.3 Rumor Vs Non-Rumor

Table 5.10 shows the performance of different categories of features per rumor and nonrumor classes. The combined set of features led to the highest fscore and recall in nonrumor and to high precision in identifying rumor. Pragmatic and social media features led to high performance in identifying rumor. The network features led to the highest recall and lowest precision in identifying rumor. For error analysis, we compared the confusion matrices (CM) of different categories of features, as shown in Figure 5.7. Applying the combined features led to a high number of false negatives (FN) in identifying rumor, whereas most of the cases were classified as nonrumor and applying the combined features led to high TP in the nonrumor cases. The high number of false negatives in the content and combined feature experiments indicated

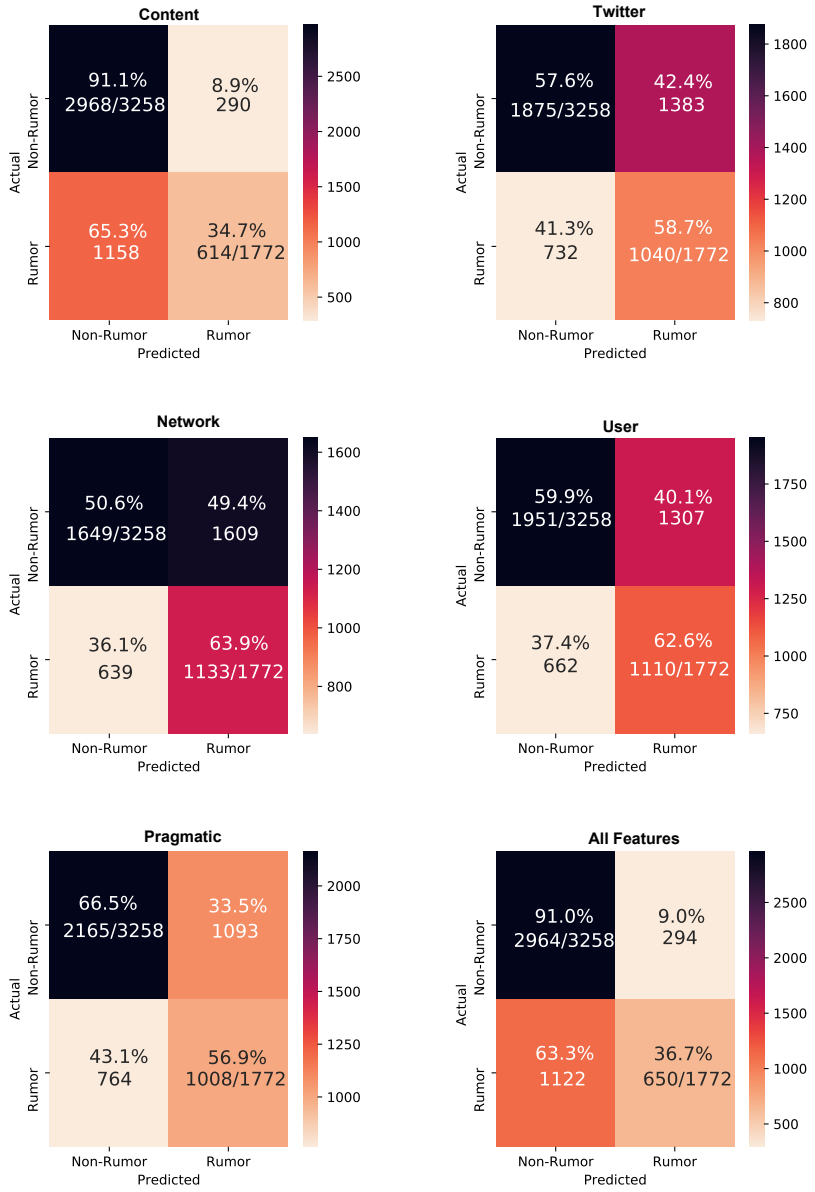


Figure 5.7: The confusion matrix of different classifiers which are trained by certain category of features.

the low accuracy of the linguistic content feature in identifying rumor. As opposed to content features, pragmatic features had the lowest number of false negatives for identifying rumor and a relatively low number of false negatives for nonrumor cases.

5.5.4.4 Features in Rumor Spread

Figure 6.11 shows the performance of using different categories of features that were accumulated by rumor spread over time from different conversations. Our objective was to understand the importance of different feature categories in different stages of rumor diffusion. According to the variations in results, the network feature extracted from $\boldsymbol{T_2}$ to $\boldsymbol{T_4}$ from the conversations did not cause any boost in the performance; however, from $\boldsymbol{T_4}$ to $\boldsymbol{T_5}$ we had a roughly 2% increase in the fscore. Overall, we could infer the importance of collecting network propagation features over time, but on the basis of the early analysis and the variation of results here, the most important feature in the network category comes from the early minutes of information diffusion. Unlike the Network features, pragmatic features that were collected from all type of tweets in the conversation led to a significant boost in performance, because from $\boldsymbol{T_1}$ to $\boldsymbol{T_5}$ we had close to a 5% improvement in performance in re-tweets and replies. Twitter or social media features in $\boldsymbol{T_2}$ got a 3% better result in comparison with the result from $\boldsymbol{T_1}$. Looking at Table 5.6 data at $\boldsymbol{T_2}$, we believe that the URLs and hashtags in that time frame could cause a boost in performance. The user provenance feature implies user popularity in the conversation. According to the variance of performance shown in Figure 6.11 for the user category the information of the participants in the conversation improves the overall performance of the model.

5.5.4.5 Events

We also investigated the performance of the proposed methodology for each evens. In Figure 5.8 (Right), the performance of the rumor identification task is illustrated for each event and different tweet types. Despite the overall improvement of the proposed methodology, the performance per event is not improving for all the events by using the conversational features over time. The results of testing the model on Charlie Hebdo test set shows even worse performance over time. We looked at the

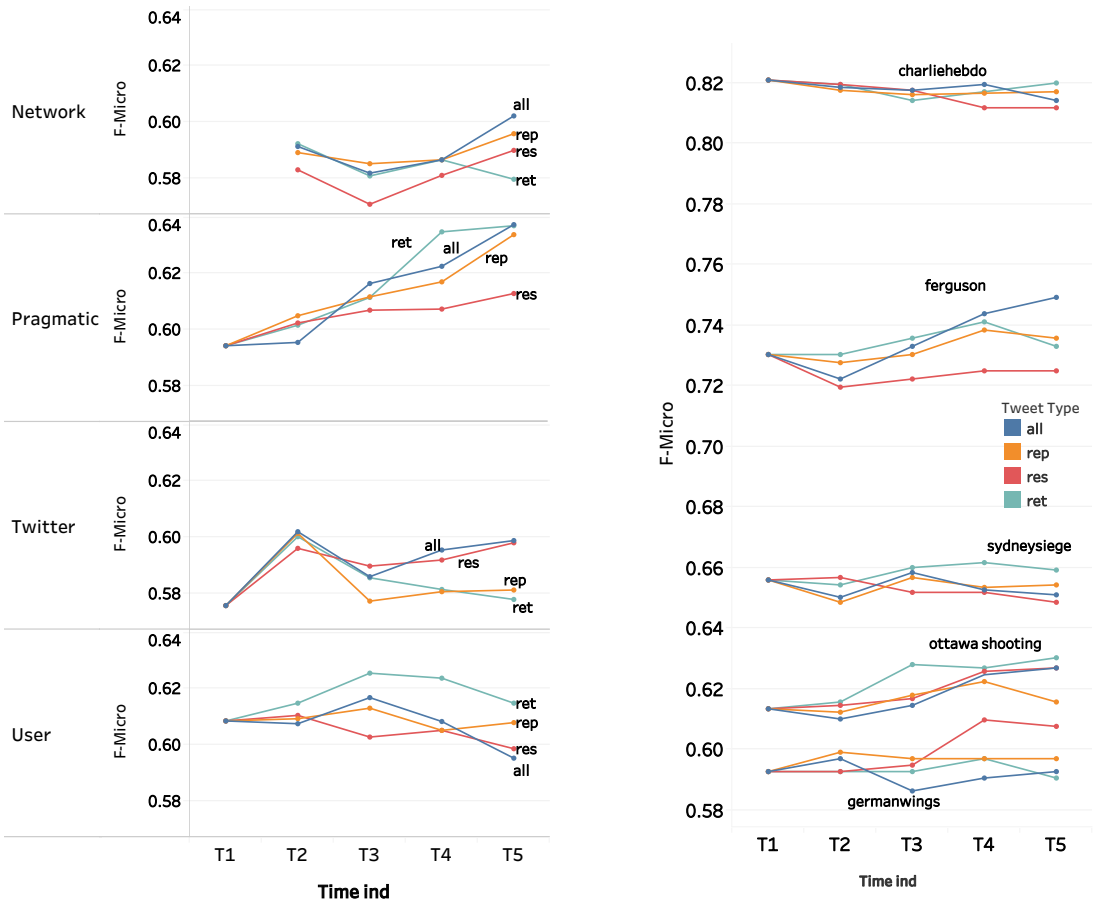


Figure 5.8: The left figure shows the performance (F-Micro) per feature category for different types of tweet in different time frames. Right figure shows the performance per event in various time frames for different types of tweet

dataset and noticed that most of the conversations in Charlie Hebdo case are a short conversation, which could be the reason since there is not enough information coded from information cascade. We assume that the length of conversation could affect the quality of our approach as for Ferguson, Germanwings, and Otawwa shooting we are seeing the high number of long-lasting conversations in the dataset.

5.6 Summary

In this chapter, we introduced a novel methodology to identify rumor by leveraging the content and metalinguistic feature of tweets. We investigated the significance of

the proposed set of features in identifying rumor. We studied the behavior of people in the rumor and nonrumor conversation using the pragmatic features. Our findings showed that the sources of the tweets in rumor conversation are more objective and easier to read than in nonrumor conversation. Tweets that initiate rumor show more committed belief and reported belief, whereas nonrumor tweets are more toward being opinionated and ask more questions. Overall, our experiments showed that the proposed methodology in this work helped us to achieve a better performance in the rumor identification task in comparison to the current state of the art. We are planning to expand the proposed methodology to a greater number of events in order to gain better insight into the nature of rumors.

Chapter 6: Analyzing the Rumor Veracity on Social Media

6.1 Introduction

The number of users who rely on social media to seek daily news and information rises daily, but not all information online is trustworthy. The unmoderated feature of social media makes the emergence and diffusion of misinformation even more intense. Consequently, the propagation of misinformation online could harm an individual or even a society. Most of the current approaches on verifying credibility perform well for the unfold topics which are already verified by a trustworthy resource [Qazvinian et al., 2011b, Hamidian and Diab, 2015]. However, the performance suffers when it comes to real life application for dealing with the emerging rumors which are priory unknown. Identifying the veracity of the rumor by relying on previous observations is a challenging task as the new emerging rumor could be entirely new regarding the event, propagation pattern, and also the provenance. Despite these challenges, many researchers have been studying how to validate the information online. Most of the current approaches rely on the generalizable attributes that could be aggregated from the content and network of the online information [Vosoughi et al., 2018, Kochkina et al., 2018, Grinberg et al., 2019]. Our first mission in this chapter is to study the veracity of rumors on twitter platform in conversation level. We use the new set of features which were described in the previous chapter and then investigate the importance of each in different time frames for rumor veracity tags. We leverage the proposed list of features in empirical analysis and examine the importance of each in a combination of various machine learning mechanism for identifying the veracity of rumor. The proposed methodology and set of features in this worked to boost the overall performance. Moreover, the newly proposed features, including the pragmatic and stance features, were significantly helpful to identify the unverified category of

rumor.

6.2 Problem Statement

We aimed to identify the veracity of the rumor conversation on social media. We relied on the information collected from both the source and the conversation and identified the likelihood that a conversation was a true or a false rumor. The standard data set with rumor veracity tags were used for all the experiments. We hypothesized that true and false conversations had unique characteristics that made them distinguishable. In this chapter, we study the conversational features in different time frames and propose a new set of features that are unique to false and true rumors. We also test our hypothesis in an empirical experiment in a rumor verification task.

6.2.1 Data

We are using the extended version of the PHEME dataset of rumors and non-rumors related to nine events. Each event contains the veracity information for each of the underlying rumor threads. This dataset [Kochkina et al., 2018] contains 3 degrees of annotation. First, each conversation thread annotated as either rumor or non-rumor which was described in the previous chapter; second, rumors as either true, false or unverified, and last, a subset of conversation stance category at the tweet level. The veracity annotation of the Rumors in this dataset conducted by the professional journalists. Annotating the rumor veracity tag is relatively a challenging task in comparison to other misinformation annotation. For example, for the PHEME dataset, The overlap between the non-expert annotator and the journalist for this dataset was only 60 65% on the rumor stories. Table 6.1 shows the overall stats of this dataset in different classes. As we can see in this dataset the events vary in size and have different class label balances. Overall, the PHEME dataset contains fewer rumors than non-rumors, while the majority class for rumors is true.

Events	Threads	Tweets	Rumours	Non-rumours	TRUE	FALSE	Unverified
Charlie Hebdo	2,079	38,268	458	1,621	193	116	149
Sydney siege	1,221	23,996	522	699	382	86	54
Ferguson	1,143	24,175	284	859	10	8	266
Ottawa shooting	890	12,284	470	420	329	72	69
Germanwings-crash	469	4,489	238	231	94	111	33
Putin missing	238	835	126	112	0	9	117
Prince Toronto	233	902	229	4	0	222	7
Gurlitt	138	179	61	77	59	0	2
Ebola Essien	14	226	14	0	0	14	0
Total	6,425	105,354	2,402	4,023	1,067	638	697

Table 6.1: The overall statistics of the PHEME data on rumor/nonrumor and rumor veracity tags for each event. The first five events are the largest and relatively balanced data sets of this corpus.

6.2.2 Reaction to True versus False Rumor in Early Hours

In this section, we aim to answer how the reaction to false and true rumor varies in the early stage of rumor propagation. Using the created time of the replied tweets, we extracted the number of tweets replied in different time intervals in all the conversations. The overall response time of the false and true rumor conversations show that 25% of the responded tweets were posted in the first 10 minutes and that more than 75% of the tweets were posted in the first 1.5 hours of the conversation. Figure 6.1 illustrates the kernel density estimate (kernel = Gaussiann', kernel size estimator method='scott') of different stories in the first two hours of rumor propagation. We picked two hours because of the majority of the tweets were posted in this time period. According to the figure, false and true cases have a similar pattern, but the number of replied tweets in true conversations are more than a false rumor in early hours of diffusion (MW-U statistic=51510031.5, pvalue=3.7661e-23). (In order to test the significant difference between rumor and nonrumors in different time intervals and distribution we used Mann-Whitney U and Kolmogorov Smirnov tests.)

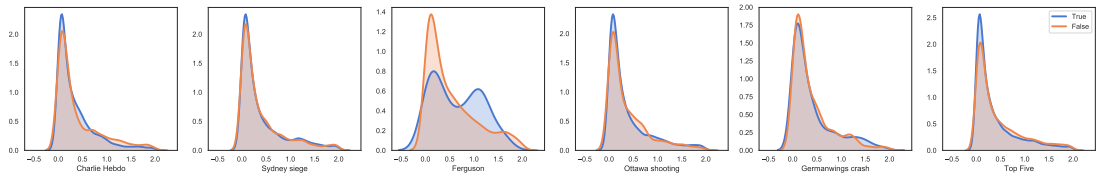


Figure 6.1: This plot shows the kernel density estimation of the false (red) versus true (blue) in the 2 hours of rumor propagation for the top five events. The x-axis shows the time elapse of the replied tweets (hours). The y-axis is the estimated density value of KDE. The top five stories are those with the highest number of tweets in the PHEME data set.

6.2.3 Reply, Response and Re-Tweet in Early Hours

In addition to the number of tweets in the early stage of rumor diffusion, the tweet type could be an informative signal of the source event popularity, novelty, and also legitimacy. We divided a tweet type into three main categories as described in the previous chapter: Replies, Re-tweets, and Response.

Figure 6.2 shows the rate of reaction to false and true rumors based on the tweet type in the first two hours of rumor propagation. According to the figure, a rumor that turns out to be true gets my number of Re-tweets Reply, (Mw-U statistic=47838.5, pvalue=0.0001), Reply (statistic=11458736.0, pvalue=5.530234288510657e-15), and also Response (statistic=51510031.5, pvalue=3.7661594728719586e-23),

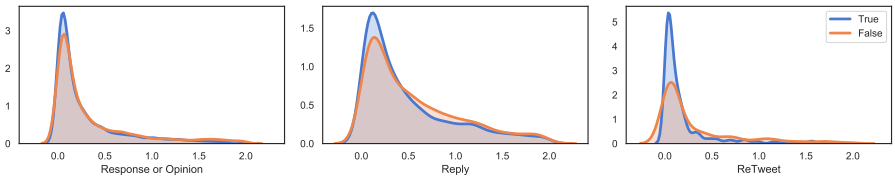


Figure 6.2: This plot shows the kernel density estimation of the false (red) versus true (blue) in the first 75% (2 hour interval) of rumor propagation for different tweet types. The x-axis indicates the response time difference to the source of the conversation (hours). The y-axis is the estimated density value of KDE.

6.3 Approach

We took a similar approach to the one proposed in the previous chapter. In order to evaluate the strength of the proposed category of features in different time frames, we trained multiple models with different combinations of the proposed set of features. We pursued three main objectives in our experiments: identifying the best performing set of features in classifying the veracity tag of the rumor; testing the importance of accumulating in-frame and cumulative features; and investigating the importance of studying the different types of tweets (e.g., Reply, Re-tweet, and Response) in identifying the veracity tag of rumors.

6.3.1 Pre-processing

In the pre-processing phase, the type of tweets in Twitter conversations is distinguished $\boldsymbol{R = \{Reply, ReTweet, Response\}}$. The second step of pre-processing is to separate tweets in different time-frames using the temporal information of the reacted tweet to the source of conversation.

6.3.1.1 Tweet Type

Tweet type could be an informative signal of the source event popularity, newsworthiness, and also legitimacy [Metaxas et al., 2014, Wu and Shen, 2015].
Using the Twitter API specification [1] about the tweet types I define each type as follow; Reply Mention, if tweet mentions more than one user from the tweet conversation, including the source of the tweet(exp. Reply to @user1 @user2 I agree....). Re-Tweets, in which author reposts (RT @user ....) the source tweets with or without some modification, and Response when tweet does not mention any other USER in the content and only leaves some comments.

[1] https://help.twitter.com/en/using-twitter/retweet-faqs

6.3.1.2 Time-Series

In the next step, we divide the reacted tweets in each rumor conversation into different time series by using the time gap of the reacted tweets with respect to the source of conversations (Time-gap indicates the time difference between the reacted tweets and the source of the conversation which is calculated by using the "created_time" attribute of the tweets). We run five experiments using the different range of time frames in $\boldsymbol{T = \{0, 6, 20, 80\ minutes, 3days\}}$. For each Event $\boldsymbol{E_i}$ in the PHEME dataset which consists of $\boldsymbol{C_i}$ rumor conversations than for $\boldsymbol{Tweet_{ij}}$, $\boldsymbol{i}$ is associated with time series index in $\boldsymbol{T}$ and $\boldsymbol{j}$ associated with the tweet type index in $\boldsymbol{R}$.

6.3.2 Feature Extraction

Using the Figure 5.2 scheme for extracting the features for each conversation of the events. The associated feature vector of the rumor conversations $\boldsymbol{C_i}$ was built in the following steps.

Step (1) Extracting all the features from the source of the conversations

Step (2) Extracting all the features of different types of tweet associated with the source of the conversation in each time frame $\boldsymbol{T_{fi}}$. For example, $\boldsymbol{T_{f2}}$ indicates the time frame between 6 to 30 minutes and $\boldsymbol{F_{rpf2}}$ is related to the features extracted from the replied tweets in the same time frame of the particular conversation.

Step (3) Extracting all the information from the different types of tweets cumulatively $\boldsymbol{T_{ci}}$ from the time the source tweet was posted until that time frame. For example, $\boldsymbol{T_{c3}}$ indicates the time between 0 and 30 minutes and $\boldsymbol{F_{rpc2}}$ is related to the features extracted from the replied tweets in the first 30 minutes of the conversation.

6.3.3 Experimental Setup

To analyze the importance of different categories of features the entire dataset was used for the training. For example, for the $\boldsymbol{T_{i,j}}$ experiment all the in-frame and cumulative features were extracted from $\boldsymbol{T_0}$, $\boldsymbol{T_0 i}$, $\boldsymbol{T_i j}$, and $\boldsymbol{T_0 j}$ time frames e.g., for the T=(6,20] experiment I use all the features from T=0, T=(0,6], T=(6,20], and T=[0,20]. For all the analysis,we first controlled for confounding variable and then removed the correlated variables with more than 80% similarities. For feature selection, we relied on Lasso [Tibshirani, 1996] and Feature ranking using the recursive feature elimination [Guyon et al., 2002] algorithm. For further adjustment of confounding variables and addressing multicollinearity, we applied Principle Component analysis (PCA) with the same number of components as features. The output of PCA was fed to the Logistic regression model for training.

6.3.4 Feature Interpretability

6.3.4.1 Network

The significant features for the network category are provided in Table 6.2. According to the table, the depth of the rumor conversation in the first 6 minutes for all types of tweets was highly correlated with being false. In other words, false rumors went deeper in the conversation tree, which is not surprising, considering the discussions in other literature [Vosoughi et al., 2018]. The same feature, between 6 and 20 minutes, is correlated with rumors that turned out to be true. One assumption is that the pace of re-tweets slowed after a couple of minutes, and true rumors get more re-tweets than false rumors in that time frame. Reply depth and response depth (the depth of the rumor conversation for the replied tweets and response tweets) are both correlated with being true.

Category	Feature	Description	Time Frame													
			6		20			80				3 days				
			F	*6*	*F*	*C*	*6*	*20*	*F*	*C*	*6*	*20*	*80*	*C*
Network	Depth	Average depth of all tweet	-1.46		1.01									
	Reply Depth	Average depth of Replies in the network tree	0.88									1.02(F)		
	Response depth	Average depth of Response in the network tree	2.10											

Table 6.2: The importance of network features in the rumor veracity task in different time frames. All the time-series features in the table were filtered only to demonstrate the significant features on the basis of 5% significance. Red and gray cells are correlated with false and true, respectively.

6.3.4.2 User Provenance

Table 6.3 presents the significance of the user provenance features in different time frames for the rumor veracity task. According to the table, in identifying the veracity tag of the rumor by using only the source information. the number of the followers of the source of rumor and the status count are correlated with false information. It is worth noting that the negative sign indicates correlation with false rumor, and the positive sign is correlated with true rumor. According to the results, verified users are less likely to post false information. Having a user profile with a description and an external (not Twitter) URL is correlated with the true rumor. In the information extracted from the conversation, the number of followers of users who re-tweet false information is higher than number of followers of users who re-tweet true rumor. We could infer that people who are followed by a lot of people on twitter are participating in false rumor conversation more often than the users who have fewer followers.

Category	Feature	Description	Time Frame																			
			src	6			20				80					3 days						
			F	*src*	*F*	*src*	*6*	*F*	*C*	*src*	*6*	*20*	*F*	*C*	*src*	*6*	*20*	*80*	*F*	*C*
User Provenance		Log(Average number of followers) of source	-0.14							0.48										
	Followers	Log(Average-number of followers) of retweeters															1.06(C)			
		Log(Average-number of followers) of responders																-0.67(C)		
	Verified	User has a verified account	0.70	0.55		0.68				0.48										
	Friends	Log(Average-number of users the replier account is following)							-0.39											
		Log(Average-number of users the retweeter account is following)					-0.79				-0.72							1.02(F)	-1.65	
	Statuses count	Log(Average Statuses count)	-0.33	-0.37		-0.36														
	Profile	User description (Existence of profile description)		1.85		1.32									1.46					

Table 6.3: The importance of user provenance features in the rumor veracity task in different time frames.

Category	Feature	Description	Time Frame																	
			src	**6**			**20**				**80**					**2 Weeks**				
			F	*src*	*F*	*src*	*6*	*F*	*C*	*src*	6	*20*	*F*	*C*	*src*	*6*	*20*	*80*	F	C
	Sentiment	The sentiment polarity																		
	Sentiment in Response	Sentiments in Respons											-0.81					-0.68(F)		
	Sentiment of Replied	Sentiments in Replied post			1.30				0.86							-1.40				
	Sentiment of Retweets	Sentiments in Re-Tweets																		
	Subjectivity	Personal opinion and factuality	0.43		0.65		0.83				0.74						0.68(C)			
	Subjectivity Reply	Subjectivity score in Replied posts																	1.53	
	Subjectivity Retweet	Subjectivity score in Re-Tweets																		
Pragmatic	Subjectivity Response	Subjectivity score in Response					-0.81		0.99			1.07(C)								0.61
	Belief (CB)	Committed Belief score																	-1.38	
	Belief (NA)	Non-Attributable Belief Score						0.46				0.45(F)							-1.40	
	Belief (ROB)	Reported Belief score		-0.55	0.54	-0.52	1.05			-0.47	1.02								-1.22	
	Belief (NCB)	None Committed Belief Score																	-2.50	
	Readability	Score of Readability	-0.05																	
	Coarse words	Number of coarse words	0.24	0.21																
	Coarse words in Replies	Average Number of coarse word							-0.44				-0.47(C)							

Table 6.4: The coefficient of the significant features from the pragmatic category in rumor veracity identification in different time frames of rumor propagation

6.3.4.3 Pragmatic Feature

Table 6.4 shows the coefficients of the significant features in the different runs of the experiments in various time frames. According to the table, a high score in reported belief in the source of conversation is correlated with false information. On the other side, high ROB in the first 6 minutes of the conversation is correlated with being true. One of our assumptions is that people use the reported belief in the source of false rumor to make their tweets look more factual and that, on the other side, people use ROB to speak on behalf of another source or entity that supports the statement made in the source tweet. Coarse words is another feature in this category that is more likely to appear in the false rumor conversations,. whereas interestingly, they are more likely to appear in the source of true rumors. Another exciting finding of this table is the subjectivity score, which is correlated with the true rumors. This observation indicated that false rumors are more opinionated than the true rumors, and this is valid both on the source and the reacted tweets in the rumor conversations.

6.3.4.4 Platform Specific

Table 6.5 presents the significant features of the social media category. According to the table, the existence of a legit URL in the content of the source tweet remains

Category	Feature	Description	Time Frame																	
			src	6		20				80					3 days					
				src	*F*	*src*	*6*	*F*	*C*	*src*	6	*30*	*F*	*C*	*src*	*6*	*30*	*120*	F	C~
Platform Specific	Hashtag	Hashtag existence	-0.23																	
	Hashtag in ReTweet	Hashtag existence in Retweet										-1.23(C)					-1.28			
	Hashtag in Response	Hashtag existence in Response					-1.9													
	Hashtag in Reply	Hashtag existence in Replies			1.04															
	Media	Media~existence Video,Picture,or GIF		0.22																
	Media in Replies	Media existence in ReTweets			-3.94						-0.006			0.006						
	Media in Response	Media existence in Response			1.2		2.7	3.3	-3.56			3.51(F)				3.06	3.31			
	Top 500 URL	If URL is among top 500 URLs										-0.006								
	URL Ligitimacy	Bias, fake, satire, ... URL	0.12	0.29		0.33				0.28					0.28					

Table 6.5: The importance of different platform specific features in rumor veracity tag in different time frames of rumor propagation

significant in all the experiments. Another interesting insight about this table is that the existence of media in the response tweet in all the time frames is highly correlated with being true. Thus it could be inferred that the existence of any sort of pictures, videos, or GIF acts as a verified reference for the true rumors. Among other features, the existence of a hashtag in the response tweet is correlated with false rumors. This is also valid for the source of the rumor.

6.3.4.5 Features Overtime

Table 6.6 shows the significance and correlation of different categories of features with true and false rumors in different time frames. If any of the proposed features in Table 5.2 was among the significant features, the corresponding cell is checked.

Feature Category	Time Frame																																			
	src		6				20								80										3 days											
	F		*src*		*F*		*src*		*6*		*F*		*C*		*src*		6		*20*		*F*		*C*		*src*		*6*		*20*		*80*		F		C	
	T	F	T	F	T	F	T	F	T	F	T	F	T	F	T	F	T	F	T	F	T	F	T	F	T	F	T	F	T	F	T	F	T	F	T	F
Network					✓	✓					✓																		✓							
User Popularity	✓	✓	✓	✓			✓	✓		✓				✓	✓			✓							✓				✓		✓	✓		✓		
Platform Specific	✓	✓	✓							✓		✓	✓					✓		✓			✓	✓			✓	✓		✓	✓			✓		
Pragmatic	✓	✓	✓	✓	✓			✓	✓	✓	✓		✓	✓		✓	✓		✓			✓						✓	✓			✓	✓	✓	✓	

Table 6.6: List of the significant categories with the veracity tag in different time frames

6.3.5 Insight from Selected Features

Table 6.7 shows the most significant features of each time frame among all the experiments. The listed features in this table were selected based on the basis of

5% significance level. We considered the Z value and standard error of each features in different experiments and selected only those with the highest Z value and low standard error, which remain equal and bellow significance rate in all the experiment. $\boldsymbol{T_i}$ corresponds to the time frame of the feature, for example, URL Legit $\boldsymbol{T_1}$ indicates that URL Legit feature of the source of conversation or Media in Response $\boldsymbol{T_2}$ indicates the existence of media in response tweets in the $\boldsymbol{T_2}$=(0,6] time frame. According to the results, the length of the tweet or syllabus count, in addition to the status count and ROB score of the source of conversation, is correlated with false rumor. Another insight is that people who tweet more often are more likely to tweet false rumor. As described earlier, one assumption about the correlation between ROB and false rumors is that using the reported language or talking on someone else's behalf makes the statement more uncertain, whereas it is expected that true statements be more defined and precise. According to our experiments, verified users on Twitter are less likely to post false rumor; however, this does not mean that verified users always post true statements. As discussed previously, the existence of media in the source of the rumor conversation is also an important signal of not being a false rumor. However, it is the opposite with respect to response tweets in the first couple of hours of rumor propagation. Another exciting finding of the false rumor conversations is that users who respond to false rumors have a higher number of followers than those who respond to true rumors. Among the features selected in the first three days of rumor propagation, the depth of the conversation for re-tweets was the most significant one and correlated with the true rumor. In summary, from the source of conversation, being verified and having legit URLs or media, in addition to having a high degree of belief, is correlated with being true. From the conversation perspective, people having media in response, asking fewer questions in the early minutes, and being subjective in their language signify the reactions to the rumor that turns out to be true.

Selected Features	Veracity Tag
Syllabus Count $\boldsymbol{T_1}$	F
URL legit $\boldsymbol{T_1}$	T
Verified $\boldsymbol{T_1}$	T
Statuses count $\boldsymbol{T_1}$	F
ROB Score $\boldsymbol{T_1}$	F
Media $\boldsymbol{T_1}$	T
Media in Response $\boldsymbol{T_2}$	T
NA in Retweets $\boldsymbol{T_2}$	F
Subjecitvity $\boldsymbol{T_2}$	T
Media in Response $\boldsymbol{T_{3c}}$	F
Media in Response $\boldsymbol{T_{3f}}$	T
Number of Follower Response $\boldsymbol{T_{3c}}$	F
Depth of Re-tweets $\boldsymbol{T_{4f}}$	T
CB in Replies $\boldsymbol{T_{4f}}$	F
Depth of Re-tweet $\boldsymbol{T_{5c}}$	T

Table 6.7: List of the significant features in all-time series. The True (T) or False (F) indicates the correlation of the corresponding feature with the rumor veracity tag

6.4 Rumor Veracity Classification

We conduct all the experiments for the rumor veracity classification by leveraging the SemEval task experimental setup. SemEval is an ongoing shared task for evaluations of computational sentiment analysis systems. Task 7 (RumourEval19) [Gorrell et al., 2019] is one of the twelve tasks, consisting of two subtasks. Task A is about stance orientation of people as supporting, denying, querying or commenting (SDQC) in a rumor discourse which we elaborate in coming chapter and Task B is about the verification of a given rumor which we discuss here. In this chapter we compared our methodology with the best performing approach in addition other top baselines of this task.

6.4.1 Data

The dataset provided for this task contains Twitter and Reddit conversation threads associated with rumors about nine different topics on Twitter and thirty different topics

Veracity examples:

***user1*:** Hostage-taker in supermarket siege killed, reports say. #ParisAttacks –URL– **[true]**

***user2*:** OMG. #Prince rumoured to be performing in Toronto today. Exciting! **[false]**

Figure 6.3: Examples from the rumor conversations and source tweets with a veracity value.

	Thread	**Tweet**	**False**	**True**	**Unverified**
Reddit	65	1805	34	18	13
Twitter	368	6419	97	161	110
Total	433	8224	131	179	123

Table 6.8: Number of the threads and tweets for the task B of the SemEval2019 which was about the rumor veracity identification on Reddit and Twitter. This data is the combination of training, development, and test set after removing the cases which were tagged as 'Removed' by the task organizers.

on Reddit. The Ottawa shootings, Charlie Hebdo, the Ferguson unrest, Germanwings crash, and Putin missing are some of the rumors in this dataset, which we also exploited in the previous section. The overall size of the data, including the development and evaluation set, is 65 rumors on Reddit and 37 rumors with 381 conversations on Twitter. Table 6.9 illustrates all the information of underlying replies and source rumor in both social media platforms. Regarding the quality of the annotation, as the task description already highlighted: the overall inter-annotator agreement rate (63.7%) explains the task to be challenging (for source tweets 81.1% then for replying tweets 62.2%). According to these numbers, there are many conflicting or inconsistent labels which will confuse the learning based models, and make the model and prediction result unstable. versample showing two examples of rumor veracity values from the dataset.

6.4.2 Experimental Setup

All the experiments were conducted in two different phases. In Phase 1, several experiments were conducted to evaluate the effectiveness of the proposed set of features in different time frames. We used the development set to tune all the models, and in Phase 2, the best-performing settings were used for evaluation of the model on the test set. It is worth noting that classification is on the conversation level for all the experiments.

	Reddit		**Twitter**	
	#Src	*#Rep*	*#Src*	*#Rep*
Training	30	667	297	4222
Development	10	426	28	1021
Evaluation	25	736	56	1010
Total	65	1829	381	6253

Table 6.9: Number of source (Src) conversations and replies (Rep) on Reddit and Twitter in the training, development, and evaluation sets.

6.4.2.1 Preprocessing

We applied various degrees of preprocessing on the content. First, we removed the very short, the deleted, and the removed cases (those labeled [deleted] or [removed] by the task organizers) from the data set. For compound words and hashtags, we used a simple heuristic. If the hashtag or word contained an uppercase letter in the middle of the word, we split it before the uppercase letter. For instance, PutinMissing was separated into two words: Putin missing.

6.4.2.2 Feature Extraction

The same methodology used for feature extraction as it was introduced in Section 5.4.2.1. Furthermore, we use the new feature called Stance and investigate the importance of that in the rumor veracity task. We go beyond that and design a new

pipeline to detect the stance tag of the post concerning the source of conversation on both Reddit and Twitter. Further explanation of the proposed pipeline for Stance is elaborated at the end of this chapter.

6.4.2.3 Baselines

We compare the best performance achieved in this work with the top performances in the SemEval 2017 [Derczynski et al., 2017] and 2019. The first performance in 2019 belongs to [Quanzhi Li, 2019] Which proposed ensambling model by leveraging the content, user and network-base features in addition to preprocessing the data annotation for the cases which were similar. We also compared our work with the best performing approach of 2017 on the Twitter data.

6.4.2.4 Hybrid Model

We also compared the newly proposed work in this chapter to a hybrid model that ranked as the third team in the Shared task of 2019 and was proposed by [Sardar Hamidian, 2019a]. The hybrid model included a neural network–based (NN) model to encode the contextual representation of the post and its source in the conversation, and then a rule-based model that was designed mainly to improve the performance on the minority classes. The proposed model expands the veracity tags for the source of the conversation to the underlying posts and creates a new set of veracity tags including Source_True, Source_False, and Source_Unverified (a six-way classification). The classification was performed in two steps. In the first step, only the source of the thread participated in the classification, and for the sources with a low confidence value a voting mechanism was applied among all the posts in associated conversations. That is, if the majority of tweets in the conversation were classified as Parent_true, then the source of the conversation would be labeled as true if it had been classified as false in the first step.

Model	F-Macro	Precision	Recall	TRUE	FALSE	Unverified
Our Proposed Method	**0.650**	**0.679**	**0.649**	0.687	0.645	**0.619**
Hybrid (SemEval 3rd)	0.26	0.382	0.362	0.53	0	0.26
Quanzhi Li (SemEval 1st)	0.577	0.596	0.603	**0.721**	**0.696**	0.313

Table 6.10: The best performance achieved by the proposed methodology in SemEval 2019 rumor veracity detection task. Per class performance is based on macro averaged F-measure.

6.4.2.5 SemEval2019

Table 6.10 shows the performance of applying the proposed methodology in this work with the first- and third-place participants of the SemEval 2019 rumor veracity classification task. The proposed model used a combination of pragmatic and user-based features in addition to random forest as the classifier. It is worth noting that the best performance was achieved by extracting the features from the whole time frames from the replied and responded tweets across the data set. Although the performance per class of [Quanzhi Li, 2019] is still higher than our method in the true and false classes, nonetheless we are getting quite remarkable performance in the unverified class and achieving a 0.61 Fscore. Our proposed approach performs better than the third-place hybrid approach in overall and per class performance.

6.4.3 Machine Learning

We investigate the performance of different machine learning approaches. In the first phase, we rely on conventional machine learning approaches including SVM, NB, RF, and Ada which were also discussed in previous chapters. Then we use the neural network approach including the single task and Multi-task learning for the same experiment in addition to Stance classification task.

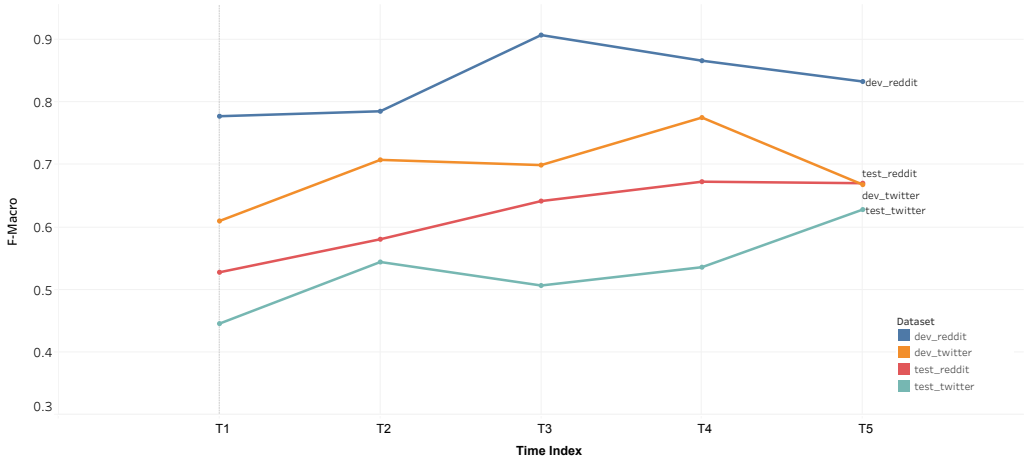


Figure 6.4: Performance of the proposed model on development and test set for each platform by the features extracted over the time from the conversation. The y-axis showing the $\boldsymbol{i}$ time index in $\boldsymbol{T_i}$. For example, $\boldsymbol{T_1}$ refers to the first second or only the features extracted from the source of the conversation, and $\boldsymbol{T_2}$ refers to the [0-6] time frame.

6.4.4 Results and Evaluation

6.4.4.1 Impact of Time Series Methodlogy

Figure 6.4 shows the performance of the different models that are trained using the extracted features in different time frames. It is observed that using the time-series features helps to achieve a better performance regarding the f-measure on both Reddit and Twitter platforms. There is a large gap between the development and test set performance. The existence of some of the training set events in the development set would explain the high performance, whereas all the events in the test set are unseen events that do not exist in the training data.

6.4.4.2 Feature Category Analysis

We conducted all the experiments by applying only a certain category of features. In addition to the features discussed in this chapter's feature section, we added stance as

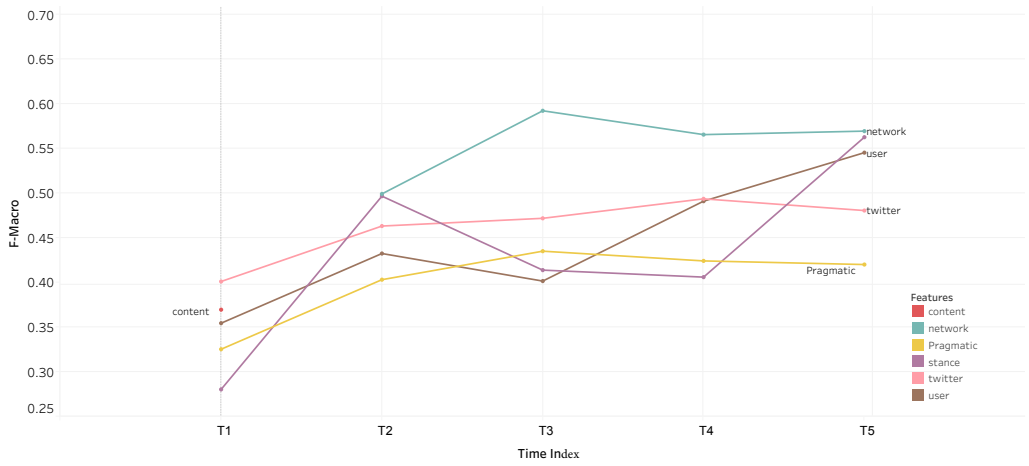


Figure 6.5: Performance of each category of features on test set extracted over the time from the rumor conversation. The y-axist shows the $\boldsymbol{i}$ time index and the x-axis shows the F-macro.

the new feature in our experiments. Figure 6.5 illustrating the performance of the proposed list of features extracted over the time from the rumor conversations on the test set in all the experiments. It is worth noting that content (red) based features refer to linguist content features from the source of conversation which in $\boldsymbol{T_0}$ as we can see the best per feature category performance was achieved by the network and user-based features. It is interesting to see that in all the categories, the time-series feature led to better performance.

6.4.4.3 Per Class Per Feature

We also investigated the impact of the proposed set of features in different True, False, and Unverified classes. classfeature shows the performance of each category of features per class in different time frames. We averaged the performance among the first 50% of the experiments, running with a different configuration, machine learning, and different types of tweets in order to have a fair evaluation of the proposed set of features. Our results per category illustrate the importance of network, stance, Twitter, and user categories for the true class. However, this is magnified in the user category more

True

Time Index	content	network	Pragmatic	stance	twitter	user
T1	0.2690		0.2834	0.0833	0.3288	0.4236
T2	0.2034	0.4124	0.2540	0.1905	0.4174	0.3077
T3	0.2034	0.2727	0.2566	0.3333	0.3431	0.3303
T4	0.2034	0.3335	0.2222	0.3151	0.3406	0.3077
T5	0.1667	0.2857	0.2614	0.4000	0.3280	0.4211

False

Time Index	content	network	Pragmatic	stance	twitter	user
T1	0.4226		0.3651	0.2857	0.4281	0.3020
T2	0.4878	0.4248	0.4059	0.4689	0.4128	0.4472
T3	0.4679	0.4667	0.4286	0.5395	0.3846	0.4900
T4	0.4783	0.4641	0.4128	0.4558	0.4410	0.5161
T5	0.4772	0.4838	0.4264	0.4593	0.4768	0.4914

Unverified

Time Index	content	network	Pragmatic	stance	twitter	user
T1	0.0000		0.1435	0.0000	0.1833	0.0000
T2	0.0000	0.0625	0.1548	0.0000	0.1667	0.0000
T3	0.0000	0.1213	0.0000	0.0000	0.0000	0.0000
T4	0.0000	0.1111	0.0000	0.0000	0.0000	0.0000
T5	0.0000	0.1818	0.0000	0.1667	0.0000	0.0357

Figure 6.6: Performance of each category of features per class in different time frames.

than in the feature category. This was also observed in the analysis section when the user-based information was correlated with the true rumors. The stance column in the true table shows that the information that came from the conversation over time helped to get better accuracy in identifying the true category. This is an interesting finding, because with no external knowledge and only by relying on the stance information the rumor could be verified, and more information from the conversation over time helps to get better performance. It is observed that more information from the conversation over time from all categories led to a better verification performance on the false class. Pragmatic, user-based, and stance features were the three main categories for which information from the conversation led to almost 10% improvement in the experiments. As discussed before, one of the advantages of the proposed methodology in this work is to identify unverified rumors. It was observed that pragmatic, Twitter, and network-based features played a key role in the leading performance of our methodology in identifying unverified rumors.

6.4.4.4 Cross-Platform

Figure 6.7 shows the performance of different experiments across Reddit and Twitter platforms. According to the result, the performance of training only on Twitter data leads to better performance on both Twitter and Reddit testing. It is interesting to see that using Reddit as the training set yields high performance in predicting false rumors, whereas Twitter performs better in predicting true rumors. The best result for the unverified class comes from training on the combined data set. It is interesting to see that the Twitter model performs surprisingly well on Reddit. We investigated the features that performed best for this experiment and realized that pragmatic features are leading to the best performance when it comes to cross-platform experiments. Overall, identifying the true classes in Reddit and false classes on twitter are extremely hard. Overall it is advantageous to use combined data for training rather than a single model for each platform.

6.4.4.5 Conventional Machine Learning

In this section the performance of all the classifiers have been compared in different time frames for different category of features. Figure 6.8 showing only the performance of the four conventional machine learning that we applied for our experiments on development and test of our experiments. The best performance achieved using the Naive based (NB) classifier by using the extracted features in $\boldsymbol{T_5}$.

6.4.4.6 Single task NN pipeline

We applied the neural network approach for this task using Figure 6.9 Single architecture. In the single step task, we use the content of the conversation in addition to the features extracted from the conversation in all the time frames. We used Glove pre-trained word embedding (Trained on Twitter data). The embedding representation of Each word goes to the Bi-LSTM layer and the last hidden layer of the Bi-LSTM

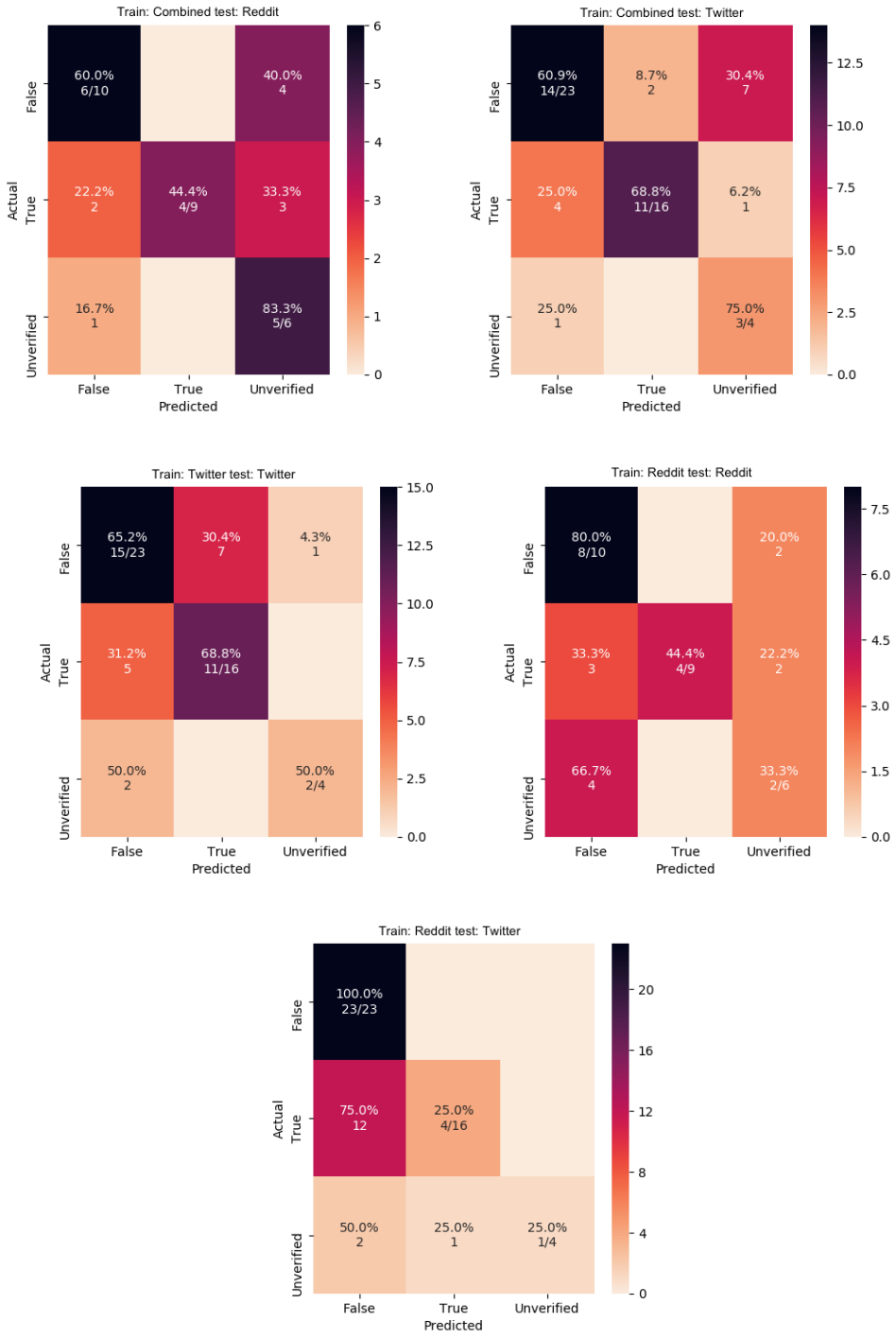


Figure 6.7: Confusion matrices on training and testing across platforms.

layer is used as the representation of the tweet. On the other side, we also examine the impact of using the Convolutions Neural network layer for filtering the manually

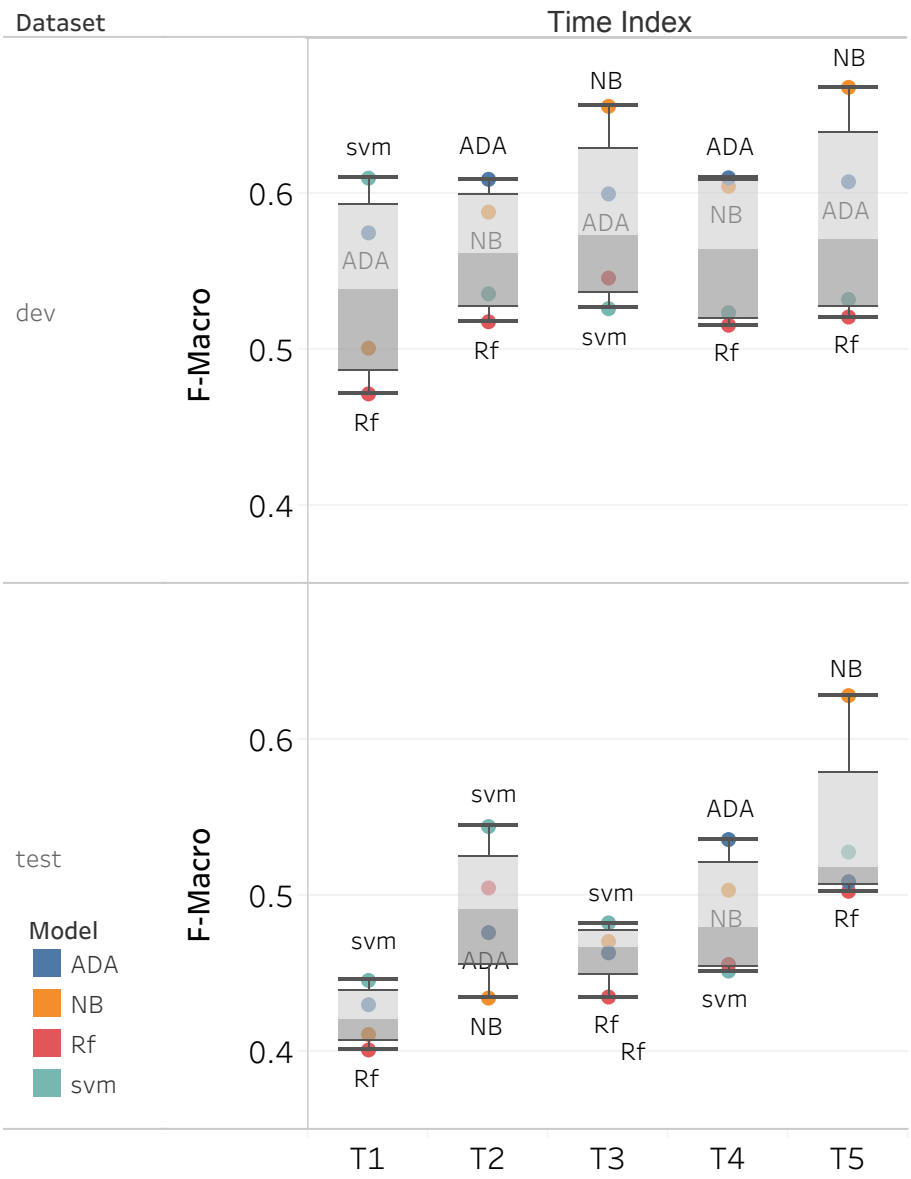


Figure 6.8: Performance of different conventional machine learning models applied in the all the experiments conducted for rumour veracity tasks on dev and test set.

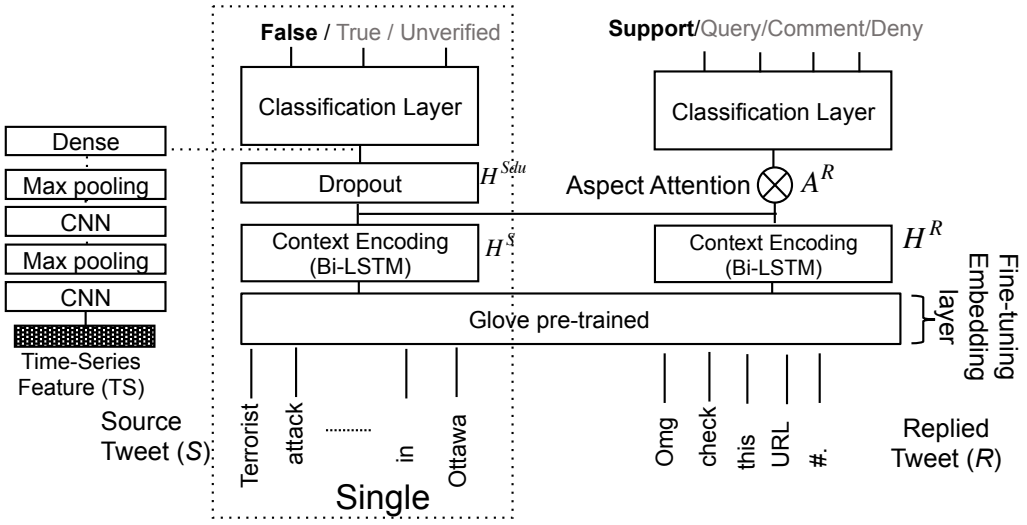


Figure 6.9: The Neural Network architecture including the single and multi-task learning pipeline with the cross-target self attention mechanism

extracted features. The hot vector representation of the manually extracted features is passed through the one dimensional CNN layer and a max pooling layer and is concatenated with the output of the dropout layer. In the last step, the concatenated layer goes to a softmax layer for classification.

6.4.5 Multi-Task Learning (MTL)

We also investigate the effectiveness of adding a homogeneous auxiliary task in a joint pipeline for identifying the rumor veracity. Our primary analysis verifies that Stance of the replied tweets in the rumor conversations is one of the key signals for determining the veracity of the source of the rumor. We expand the proposed MTL pipeline by adding the cross-target attention mechanism.

6.5 Task 2: Stance

Stance definition in social media is the opinion orientation of the people in the conversation toward the certain topic, entity or event. Identifying Stance is determining whether different posts associated with a rumour support it, deny it, query it, or

	Thread	Tweet	*Comment*	*Query*	*Deny*	*Support*
Reddit	65	1805	1630	82	54	39
Twitter	368	6419	4311	515	482	1111
Total	433	8224	5941	597	536	1150

Table 6.11: Number of threads and tweets for stance classification on Reddit and Twitter. This data is the combination of training, development, and test set.

just comment on it. The expressed stance of the user in the conversation could be indicative of the veracity of the source rumour. In this section we aim to build the stance classification component to be used in the rumor veracity task. We apply the supervised machine learning approach in addition to external features from the content of the post in socil media conversations. We described different parts of the proposed pipeline for stance classification component in coming sections.

6.5.1 Data: SemEval 2019

Dislike the previous copra this dataset contains the data of two domains including the Twitter and Reddit threads associated with rumors; nine different topics on Twitter and thirty different topics on Reddit. The Ottawa shootings, Charlie Hebdo, the Ferguson unrest, Germanwings crash, and Putting missing are some of the rumors which were also existing in previous SemEval datasets. The overall size of the data after removing the cases which were labled as removed by the organizers of the SemEval is illustrated in the Table 6.11. One of the samples from this dataset is illuestrated in Figure 6.10

6.5.1.1 Data insight

Figure 6.11 shows the distribution of the tags for veracity (Task B) and stance (Task A) across different platforms. According to the table, the stance orientation of the rumor conversations varies between Twitter and Reddit. In general, Reddit users leave more comments than Twitter users and this is regardless of the rumor veracity.

SDQC support classification. Exp 1:

u1: We understand there are two gunmen and up to a dozen hostages inside the cafe under siege at Sydney.. ISIS flags remain on display #7News **[support]**

- **u2:** @u1 not ISIS flags **[deny]**
- **u3:** @u1 sorry - how do you know it's an ISIS flag? Can you actually confirm that? **[query]**
  - **u4:** @u3 no she can't cos it's actually not **[deny]**
- **u5:** @u1 More on situation at Martin Place in Sydney, AU –LINK– **[comment]**
- **u6:** @u1 Have you actually confirmed its an ISIS flag or are you talking shit **[query]**

SDQC support classification. Exp 2:

u1: These are not timid colours; soldiers back guarding Tomb of Unknown Soldier after today's shooting #StandforCanada –PICTURE– **[support]**

- **u2:** @u1 Apparently a hoax. Best to take Tweet down. **[deny]**
- **u3:** @u1 This photo was taken this morning, before the shooting. **[deny]**
- **u4:** @u1 I don't believe there are soldiers guarding this area right now. **[deny]**
  - **u5:** @u4 wondered as well. I've reached out to someone who would know just to confirm that. Hopefully get response soon. **[comment]**
    - **u4:** @u5 ok, thanks. **[comment]**

Figure 6.10: Examples of tree-structured of rumor thread discussing the veracity of a rumour, where the label associated with each tweet is the target of the stance classification

In false rumors Twitter conversations are more oriented toward denial than Reddit's conversations; however, Twitter users support and deny false rumors to relatively the same extent. Twitter users are more supportive and ask more questions in regards to true rumors than the Reddit users, but they both deny true rumors to almost the same amount.

Interestingly, in both platforms, people question unverified rumors more than true and false rumors. For the source of conversation, Reddit and Twitter are significantly different. Regardless of the veracity, the source in Reddit conversations is more skewed to the query than the other stance tags, while Twitter is more toward the support. Despite some common characteristics Reddit and Twitter users behave differently when it comes to rumors. Reddit users do not deny the TRUE or UNVERIFIED rumors and question more when the rumor is false, yet Twitter users support more without any inquiries. It is worth noting that the conclusions mentioned in this section could only be valid for the data provided and in other conditions the same correlations might not be present.

			Socialmedia	
Source/Replies	Task B	Task A	Reddit	Twitter
Replies	False_Source	comment	93.58%	68.29%
		deny	3.95%	12.15%
		query	1.73%	6.61%
		support	0.74%	12.96%
	True_Source	comment	88.89%	68.67%
		deny	5.56%	6.29%
		query	1.85%	8.57%
		support	3.70%	16.47%
	Unverified_Source	comment	88.31%	68.65%
		deny	6.49%	7.84%
		query	3.90%	9.20%
		support	1.30%	14.31%
Source	False	comment		3.23%
		deny	11.76%	6.45%
		query	64.71%	
		support	23.53%	90.32%
	True	comment		0.73%
		deny		2.19%
		query	85.71%	
		support	14.29%	97.08%
	Unverified	comment		8.16%
		deny		2.04%
		query	83.33%	
		support	16.67%	89.80%

Figure 6.11: The distribution percentage of stance and verification tags on Twitter and Reddit dataset.“TaskB_Source" (exp. False_Source) indicate the verification tag of the source of conversation.

6.5.2 Approach

For the stance classification task, we mainly rely on the content to determine the stance of the sources and replied response in the conversation. Our primary analysis in the insight section showed that there is a significant correlation between the rumor veracity and stance orientation of the people in the conversation. We observed that for the unverified rumors, the stance orientation is more toward queries rather than concluding support or denial; on the other hand, for true rumors, people are more likely to support or comment on the conversation than question or deny. It could be concluded that stance is key information to determine the veracity of the source in the rumor conversation. We define the Stance classification as a four-way classification

experiment. We propose a multi-task learning approach, which was described earlier in mtlver. The proposed model including a neural network-based (NN) model to encode the contextual representation of the post and its parent in the conversation. We apply the attention mechanism, which is mainly designed to improve the performance by discovering domain-specific aspects for cross-target stance inference. In the coming sections we will further explain the different components of stance classification pipeline in Figure 6.9.

6.5.2.1 Embedding layer

Two main inputs to these layers are stance-bearing Response from the conversation $\boldsymbol{R}$ and source $\boldsymbol{S}$ (e.g, source tweet in Figure 6.10 and one of the replied tweets in the same conversation). We use word embeddings [Pennington et al., 2014] to represent each word in the input as a dense vector. The output of the embedding layer are two sequences of vectors $\boldsymbol{S} = \{\boldsymbol{s_1}, ..., \boldsymbol{s_{|S|}}\}$ and $\boldsymbol{R} = \{\boldsymbol{r_1}, ..., \boldsymbol{r_{|R|}}\}$, where $\boldsymbol{s}$, $\boldsymbol{r}$ are word vectors.

6.5.2.2 Context Encoding Layers

In these layers, I encode the contextual information using the source and conversation content. I use a Bi-directional Long Short-Term Memory Network (Bi-LSTM) to capture the adjacent left and right contexts of each word in the input. Using the Equation (6.1), I first use a Bi-LSTM to encode the source of conversation: where $\boldsymbol{h} \in \mathbb{R}^{\boldsymbol{h}}$ and $\boldsymbol{r} \in \mathbb{R}^{\boldsymbol{h}}$ are the hidden and cell state of the LSTM. The arrow symbols indicate the forward and backward passes and $\boldsymbol{t_i}$ is the word vector at time step $\boldsymbol{i}$.

$$\begin{aligned}
[\overrightarrow{\boldsymbol{h_i^S}}\overrightarrow{\boldsymbol{c_i^S}}] &= \overrightarrow{\boldsymbol{LSTM_S}}(\boldsymbol{s_i}, \overrightarrow{\boldsymbol{h_{i-1}^S}}, \overrightarrow{\boldsymbol{c_{i-1}^S}}) \\
[\overleftarrow{\boldsymbol{h_i^S}}\overleftarrow{\boldsymbol{c_i^S}}] &= \overleftarrow{\boldsymbol{LSTM_S}}(\boldsymbol{s_i}, \overleftarrow{\boldsymbol{h_{i-1}^S}}, \overleftarrow{\boldsymbol{c_{i-1}^S}})
\end{aligned} \tag{6.1}$$

In order to get the condition encoding of a replied sentence $\boldsymbol{R}$, I get the output of

the last hidden state of the $\boldsymbol{Bi-LSTM_S}$ layer to be used as the initialization for the $\boldsymbol{BILSTM_R}$. The output of $\boldsymbol{Bi-LSTM_R}$ is the contextuall encoding of the sentence $\boldsymbol{H^R = \{h_1^R, ..., h_{|R|}^R\}}$ and $\boldsymbol{h}$ is the concatination of forward and backward LSTM layers.

$$\begin{aligned}[\overrightarrow{h_1^R}\ \overrightarrow{c_1^R}] &= \overrightarrow{LSTM_R}(R_1, \overrightarrow{h_{|S|}^S}, \overrightarrow{c_{|S|}^S}) \\ [\overleftarrow{h_{|R|}^R}\ \overleftarrow{c_{|R|}^R}] &= \overleftarrow{LSTM_R}(R_{|R|}, \overleftarrow{h_1^S}, \overleftarrow{c_1^S})\end{aligned} \tag{6.2}$$

6.5.2.3 Aspect Attention Layer

I use the self-attention method to signal the core parts of a stance-bearing replied sentence. Self-attention is a mechanism for selecting certain parts of a sequence by relating to its elements at different positions [Vaswani et al., 2017]. In this problem, the self-attention process is based on the assumption that the core parts of the source and replied sentence are those that are compatible with the semantics of the source and entire sentence. I score the compatibility between the encoded sequence $\boldsymbol{H^R}$ and each of its hidden states $\boldsymbol{h^R}$:

$$\boldsymbol{c_i = w_2^\top \sigma(W_1 h_i^R + b_1) + b_2} \tag{6.3}$$

Where $\boldsymbol{W_1}$, $\boldsymbol{w_2}$, $\boldsymbol{b_1}$ and $\boldsymbol{b_2}$ are all trainable parameters a,d $\boldsymbol{\sigma}$ is the activation function. I computer the attention wight $\boldsymbol{a_i}$ for each hidden state using the softmax operation.

$$\boldsymbol{a_i = \frac{exp(c_i)}{\sum_{j=1}^{|R|} exp(c_j)}} \tag{6.4}$$

In the final step of the attention layer, I encode the representation based on the attention wights. $\boldsymbol{A^R}$ is the domain aspect encoding for the of the attention layer.

$$\boldsymbol{A^R = \sum_{j=1}^{|R|} a_i h_i^R} \tag{6.5}$$

6.5.2.4 Prediction Layer

For both tasks I use the softmax layer for the prediciton. For the Stance task, the $\boldsymbol{A^R}$ is used and for the veracity tag the concatenation of $\boldsymbol{H^{sdu}}$ and the output of CNN pipeline $\boldsymbol{TS}$ is passed to Multi layer perceptron ($\boldsymbol{MLP}$).

$$\begin{aligned} \hat{\boldsymbol{Y}_{stance}} &= \boldsymbol{softmax}(\boldsymbol{MLP}(\boldsymbol{A^R})) \\ \hat{\boldsymbol{Y}_{Veracity}} &= \boldsymbol{softmax}(\boldsymbol{MLP}(\boldsymbol{H^{Sdu}} \bigoplus \boldsymbol{TS})) \end{aligned} \tag{6.6}$$

6.5.3 Model Training

I use the multi class-cross-entropy loss for both tasks and sum up the value of the error function for the MTL experiments. For individual experiment I only used single cross entropy function.

$$\boldsymbol{J(\theta)} = -\sum_{i}^{N}\sum_{j}^{C} \boldsymbol{y_j^{(i)}} \boldsymbol{log}\ \hat{\boldsymbol{y}}_j^{(i)} + \boldsymbol{\lambda}||\Theta|| \tag{6.7}$$

In this equation, $\boldsymbol{N}$ is the size of the training set, and $\boldsymbol{y}$ is the gold labels or the ground truth for each class in the associated task. $\hat{\boldsymbol{y}}$ indicated the predicted probability for each class.$\boldsymbol{\lambda}$ is the $\boldsymbol{L_2}$ coefficient and $\boldsymbol{\theta}$ denotes the trainable parameters in this model.

6.5.3.1 Training

For all the NN experiments we run the training for five time and the average result is reported in this table. Adam [Kingma and Ba, 2014], Root Mean Square Propagation (RmsProp), and Stochastic Gradient Descent (SGD) optimization algorithms. The parameters get updated in every training epoch. The character and Glove pre-trained embedding size [100,200,300] are examined with batch size 50 with 100 epochs. The training is stopped after no improvements in five consecutive epochs to ensure the convergence of the models. The highest performance on the development set was

	F-Macro	Precision	Recall
Conventional(NB)	**0.651**	**0.679**	**0.649**
Hybrid [Sardar Hamidian, 2019a]	0.260	0.382	0.362
QuanzhiLi [Quanzhi Li, 2019]	0.577	0.596	0.603
NN	0.248	0.288	0.247
NN+Feature	0.442	0.531	0.445
MTL	0.280	0.298	0.282
MTL+Feature	0.477	0.501	0.472

Table 6.12: The result of the conventional and NN machine learning approaches in rumor veracity task

achieved under the following parameters: hidden size of Bi-LSTM (200); optimization (RMSprop); initial learning rate (0.005); L2 ($\boldsymbol{Lambda = 0.1}$); character and word embedding size (100,100); dropout size (0.3). For the Convectional layers of the manually generated features, we use two CNN (32-3 Kernal, 16-3 kernel) layers with Relu activation following with max-pooling layers with size two and two dens layers. The output of the CNN network for the handcrafted features is flattened and concatenated with Contextual embedding of the source tweet.

6.5.4 Result and Evaluation

Surprisingly the NN approach could not get a better performance in comparison with the conventional machine learning approach. However, in the primary phase of running the model, the MTL pipeline could get higher performance on the development set. We believe that the existence of the similar events in dev and training set causes the deep learning model getting high performance on the development set.

6.5.4.1 Stance

Table 6.13 illustrating the performance of applying the proposed methodology on the stance classification task. It is worth noting that Stance classification was not the main objective of this project. However, the proposed methodology in this work could achieve 0.49 F-macro and ranked as the fourth team among 21 participants.

	F-Macro	**Presision**	**recall**
FB	0.477	0.450	0.530
Single NN	0.12	0.12	0.12
MTL	0.353	0.364	0.386
MTL+Attention	0.498	0.475	0.556
Hybrid	0.435	0.417	0.511
BLCU[Ruoyao Yang, 2019]	**0.619**	**0.634**	**0.602**

Table 6.13: The performance of applying the MTL and attention mechanism on the stance classification task.

The best performance achieved by [Ruoyao Yang, 2019] by expanding the provided dataset for the minority task. Table 6.13 showing the performance of feature Based (FB) by relying on the feature which was proposed in previous chapters. The Single NN experiment is based on relying the Single path at Figure 6.9 network without using the handcrafted features. The MTL +Attention mechanism leads to the best performance among all the experiments by reaching 0.498 F-macro.

6.6 Summary

In this chapter, we investigated the impact of the time-series features in addition to multi-tasks learning methodology on rumor veracity task. Our analysis showed that pragmatic feature and stance information of the conversation is significantly important in identifying the unverified type of rumor. On the other hand, user information and the existence of legitimate URLs are highly correlated with true rumors. We further studied the stance of rumor conversations on Reddit and Twitter platforms. In one of our observations, we realized that Twitter users are more supportive and ask more questions in regards to true rumors than the Reddit users, but they both deny true rumors to almost the same amount. We leveraged the insight obtained from the experiments and developed an automated veracity classification pipeline, which led to state-of-the-art performance in SEMEVAL 2019 veracity task.

Chapter 7: Conclusions & Future Directions

7.1 Conclusions

Social media plays a pivotal role in the spread of information, however, the absence of validation and verification of such environments allows for the spread of rumors online. In this thesis, we investigated the rumor phenomena from different angles. We designed multiple experiments and introduced a novel set of linguistic and meta-linguist attributes which could be leveraged in identification and verification of rumor on social media. We first addressed the rumor tracking task in which rumor classification is done on the tweet level. The main objective in rumor tracking is to identify the related rumor tweets in a three-way classification task, which are either rumor, nonrumor, or undetermined. We compared single-step and two-step supervised paradigms and relied on the content and social media specific attributes. Our results signified the importance of the content-based features in comparison to other feature categories. Moreove, the results yielded were superior to all the supervised methods tested on the same test set. We observed from the other related work that applying the dense representation instead of conventional BOW or TF_IDF aids to capture the semantic textual similarity and boost the overall performance of rumor retrieval task and on tweets in particular [Agirre et al., 2015, Guo and Diab, 2012]. Leveraging the combination of tweet latent vector or dense presentation of the tweet and BOW in rumor retrieval task obtained the highest Mean Average Precision (MAP) in comparison to all the supervised methods tested on the same test set.

We went beyond the rumor tracking on the tweet level and investigated the rumor detection problem on the conversation level. We mainly examined the signals that can help us to identify the unseen rumors which are unknown a priori. This task is more challenging since the information of the new rumor including the event and

propagation pattern are totally unknown in advance. We relied on conversational features to investigate rumors and their spread on social media. A novel time-series method is designed for this task to monitor and extract four categories of attributes from the source and every single post in the conversation. These four categories consist of network propagation, user profile, social media specific, and a novel set of content-based features. Our experiments provide exciting insights into various perspectives. We observe the common use of reported belief in the content of the rumors as opposed to nonrumor cases which the source of rumors are more subjective and interrogative. In another observation, we noticed that people use the hate language more often in rumor conversations than nonrumor ones. From the social media features, the existence of media in the source of conversation which includes pictures, videos or gifs are an imprudent signal of being nonrumor, as opposed to URL, reported belief, and status count (number of the tweets that user posted before) which are correlated by being a rumor. One other interesting finding of this work is that the verified and unverified users post rumor to relatively the same extent and in a few cases the source of rumors were the verified news outlets profile.

Rumor veracity could turn out to be true, false, or remain unverified for a long time. In another task, we investigated the veracity of rumors in different time series. We observed that in the rumors which turn out to be true the rate of re-tweet is higher than the rumor conversations which turn out to be false. In Chapter 6 We also discussed and examined the importance of the Stance signal in rumor veracity. We developed a new component to identify the stance of the replied post concerning the source of conversations and investigated the people reactions to the true and false rumors in different social media platforms including Reddit and Twitter. Our observations showed that the Reddit users leave more comments than Twitter users, and this is regardless of the rumor veracity. Moreover, in false rumors, Twitter conversations are more oriented toward denial than Reddit's conversations; however, Twitter users

support and deny false rumors to relatively the same extent. Also, Twitter users are more supportive and ask more questions in regards to true rumors than the Reddit users, but they both deny true rumors to almost the same amount. We leveraged the insight gained from the statistical analysis and our observations from the data to the empirical analysis in order to test the predictive power of the proposed set of features in identifying the veracity of rumor. We showed that the proposed methodology in this thesis yields the state-of-the-art performance.

7.2 Problems and Limitations

- RDC and TLV are limited to the stories that exist in the training set and not generalizable to unseen rumors.
- Rumor Identification pipeline works better when it comes to the rumor which remains unverified for a long time after the post.
- The quality of the analysis in the rumor identification experiment could be improved by testing on more datasets with more number of rumors.
- The improvement of the time-series features in rumor identification is significant but with minor improvements on the small conversation.
- The performance of the two-step committed belief tagger suffers from poor quality of the event sequence tagger from the first step of the pipeline.

7.3 Research Contributions

This thesis contributes the following to the field of social media and computational linguistic in online discourse:

- The thesis contributes RDC as a novel machine learning pipeline for rumor tracking task and investigates the importance of content features on rumor

tracking and opinion classification [Hamidian and Diab, 2015].

- The thesis investigates the application of TLV instead of the conventional approaches like TF-IDF and Bag of Word in a two-way Rumor Retrieval (RR) task. This component uses the latent representation of the tweet and leads to a state of the art performance in rumor retrieval task.

  Dataset

  As an extension of the RR pipeline, we collected, pre-processed, and annotated more than 10,000 rumor from 2014 to 2016. The tweet ID and rumor tag are available for research use [Hamidian and Diab, 2016].

- The thesis introduces the first NN Multi-Genre sequence tagging LCB model. This tagger has since generated great research interest in applying it to other tasks, such as rumor identification, stance classification, and rumor veracity [Rambow et al., 2016].

- The thesis contributes a novel domain-independent approach for identification of unseen emerging rumor. The proposed methodology differs significantly from the conventional methods which are only applicable to identify the unfolded longstanding rumors which are known a priori. The methodology is based on leveraging the linguistic and meta-linguistic signals from the content, source, and network propagation information in rumor conversation. [Sardar Hamidian, 2019b].

- The thesis investigates the rumor veracity task by studying the interpretability of contextual and non-contextual signals in rumor conversations. The thesis also compares the results obtained by several conventional and neural network machine learning systems. The proposed models in this work could outperform the current state-of-the-art performance in rumor veracity classification in SemEval 2019 [Sardar Hamidian, 2019a].

- The thesis provides a novel neural network approach by applying the multi-task learning framework to attain the stance orientation of the people in rumor conversation. The stance signal could significantly boost the overall performance of identifying the Unverified type of rumor in the related rumor veracity tasks. [Sardar Hamidian, 2019a].

7.3.1 Publications

- Hamidian, Sardar, and Mona Diab. "GWU NLP at SemEval-2019 Task 7: Hybrid Pipeline for RumourVeracity and Stance Classification on Social Media." Proceedings of the Thirteenth International Workshop on Semantic Evaluation, NAACL, 2019

- Hamidian, Sardar, and Mona Diab. "Rumor detection and classification for Twitter data." Proceedings of the Fifth International Conference on Social Media Technologies, Communication, and Informatics (SOTICS). 2015.

- Hamidian, Sardar, and Mona Diab. "Rumor identification and belief investigation on Twitter." Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis. 2016.

- Hamidian, Sardar, and Mona Dian, "Fine-Grained and Coarse-Grained Rumor Detection Methodology", Mid-Atlantic Student Colloquium on Speech, Language and Learning, MASC-SLL, January, 2015.

- Rambow, Owen, Tao Yu, Axinia Radeva, Sardar Hamidian, Alexander Fabbri, Debanjan Ghosh, Christopher Hidey et al. "The Columbia-GWU system at the 2016 TAC KBP BeSt evaluation." In Proceedings of the 2016 NIST TAC KBP Workshop. 2016.

- Hamidian, Sardar, and Mona Diab. "Rumor Detection and Analysis on Social

Media", (Will be submitted to TACL 2019)

- Hamidian, Sardar, and Mona Diab. "Multi-platform Rumor veracity and stance Classification pipeline",NLP for Freedom: Censorship, Disinformation, and Propaganda, (Will be submitted to NLP4IF 2019)

7.4 Future Directions

- We aim to expand the methodology proposed in this work to other languages and investigate the importance of the content based features and explicitly pragmatic features on rumor veracity task.

- We aim to improve the proposed time series-method and leverage more advanced methodology to encode conversational and network propagation information. Using unsupervised approach like BigGraph [Lerer et al., 2019] is one of the options.

- Levering the external resource like Wikipedia for the rumor veracity pipeline by using the attention mechanism.

- We have collected more than 2 billion tweets in 2018 using the Twitter API. One of the main goals is to create the most extensive multilingual rumor corpus, which consists of the same rumors in different languages. We aim to study the different and common modalities and pragmatics of natural language which are used in different languages in rumor conversations.

- Dividing the rumors into different categories based on the event type (Pas, Future, Ongoing) and studying the content based attributes which are unique to the certain type of stories or events.

Thank you...

Bibliography

A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. Passonneau. Sentiment analysis of twitter data. In *Proceedings of the workshop on languages in social media*, pages 30–38. Association for Computational Linguistics, 2011.

E. Agirre, C. Banea, C. Cardie, D. Cer, M. Diab, A. Gonzalez-Agirre, W. Guo, I. Lopez-Gazpio, M. Maritxalar, R. Mihalcea, et al. Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pages 252–263, 2015.

M. M. Ajmal, S. Almatrooshi, and M. Hussain. Since 2019 since 2018 since 2015 custom range... sort by relevance sort by date role of public policies in promoting csr: empirical evidence from business and civil society of uae. 2018.

H. Allcott and M. Gentzkow. Social media and fake news in the 2016 election. *Journal of Economic Perspectives*, 31(2):211–36, 2017.

G. W. Allport and L. Postman. The psychology of rumor. 1947.

A. Bandura and S. Wessels. Self-efficacy. 1994.

C. Boididou, S. Papadopoulos, Y. Kompatsiaris, S. Schifferes, and N. Newman. Challenges of computational verification in social multimedia. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 743–748. ACM, 2014.

J. Cao, J. Guo, X. Li, Z. Jin, H. Guo, and J. Li. Automatic rumor detection on microblogs: A survey. *arXiv preprint arXiv:1807.03505*, 2018.

R. Caruna. Multitask learning: A knowledge-based source of inductive bias. In *Machine Learning: Proceedings of the Tenth International Conference*, pages 41–48, 1993.

C. Castillo, M. Mendoza, and B. Poblete. Information credibility on twitter. In *Proceedings of the 20th international conference on World wide web*, pages 675–684. ACM, 2011.

C. Castillo, M. Mendoza, and B. Poblete. Predicting information credibility in time-sensitive social media. *Internet Research*, 23(5):560–588, 2013.

C.-C. Chang and C.-J. Lin. Libsvm: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):27, 2011.

L. Derczynski, K. Bontcheva, M. Liakata, R. Procter, G. W. S. Hoi, and A. Zubiaga. Semeval-2017 task 8: Rumoureval: Determining rumour veracity and support for rumours. *arXiv preprint arXiv:1704.05972*, 2017.

R. Deveaud and F. Boudin. Effective tweet contextualization with hashtags performance prediction and multi-document summarization. In *INitiative for the Evaluation of XML Retrieval (INEX)*, pages n–a, 2013.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

M. T. Diab, L. Levin, T. Mitamura, O. Rambow, V. Prabhakaran, and W. Guo. Committed belief annotation and tagging. In *Proceedings of the Third Linguistic Annotation Workshop*, pages 68–73. Association for Computational Linguistics, 2009.

N. DiFonzo and P. Bordia. *Rumor psychology: Social and organizational approaches*, volume 1. American Psychological Association Washington, DC, 2007.

P. Domm. False rumor of explosion at white house causes stocks to briefly plunge; ap confirms its twitter feed was hacked. *CNBC. COM*, 23, 2013.

Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1): 119–139, 1997.

G. Gorrell, K. Bontcheva, L. Derczynski, E. Kochkina, M. Liakata, and A. Zubiaga. SemEval-2019 Task 7: RumourEval: Determining rumour veracity and support for rumours. In *Proceedings of SemEval*. ACL, 2019.

J. Gottfried and E. Shearer. *News Use Across Social Medial Platforms 2016*. Pew Research Center, 2016.

N. Grinberg, K. Joseph, L. Friedland, B. Swire-Thompson, and D. Lazer. Fake news on twitter during the 2016 us presidential election. *Science*, 363(6425):374–378, 2019.

W. Guo and M. Diab. A simple unsupervised latent semantics based approach for sentence similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 586–590. Association for Computational Linguistics, 2012.

W. Guo, W. Liu, and M. Diab. Fast tweet retrieval with compact binary codes. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 486–496, 2014.

I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine learning*, 46(1-3):389–422, 2002.

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

S. Hamidian and M. Diab. Rumor detection and classification for twitter data. In *Proceedings of the Fifth International Conference on Social Media Technologies, Communication, and Informatics (SOTICS)*, pages 71–77, 2015.

S. Hamidian and M. Diab. Rumor identification and belief investigation on twitter. In *Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 3–8, 2016.

J. Harsin et al. The rumour bomb: Theorising the convergence of new and old trends in mediated us politics. *Southern Review: Communication, Politics & Culture*, 39 (1):84, 2006a.

J. Harsin et al. The rumour bomb: Theorising the convergence of new and old trends in mediated us politics. *Southern Review: Communication, Politics & Culture*, 39 (1):84, 2006b.

N. Y. Hassan, W. H. Gomaa, G. A. Khoriba, and M. H. Haggag. Supervised learning approach for twitter credibility detection. In *2018 13th International Conference on Computer Engineering and Systems (ICCES)*, pages 196–201. IEEE, 2018.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9 (8):1735–1780, 1997.

P. Hosseini, M. Diab, and D. A. Broniatowski. Does causal coherence predict online spread of social media? In *International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation (SBP-BRiMS).* Springer., 2019.

M. Hüning. Textstat simple text analysis tool. *Dutch Linguistics, Free University of Berlin, Berlin*, 2005.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

R. H. Knapp. A psychology of rumor. *Public opinion quarterly*, 8(1):22–37, 1944.

E. Kochkina, M. Liakata, and A. Zubiaga. All-in-one: Multi-task learning for rumour verification. *arXiv preprint arXiv:1806.03713*, 2018.

C. M. M. Kotteti, X. Dong, and L. Qian. Multiple time-series data analysis for rumor detection on social media. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 4413–4419. IEEE, 2018.

K. K. Kumar and G. Geethakumari. Detecting misinformation in online social networks using cognitive psychology. *Human-centric Computing and Information Sciences*, 4 (1):14, 2014.

S. Kwon, M. Cha, and K. Jung. Rumor detection over varying time windows. *PloS one*, 12(1):e0168344, 2017.

A. Lerer, L. Wu, J. Shen, T. Lacroix, L. Wehrstedt, A. Bose, and A. Peysakhovich. Pytorch-biggraph: A large-scale graph embedding system. *arXiv preprint arXiv:1903.12287*, 2019.

X. LIAO, Z. HUANG, D. YANG, X. CHENG, and G. CHEN. Rumor detection in social media based on a hierarchical attention network. *SCIENTIA SINICA Informationis*, 48(11):1558–1574, 2018.

Y. Liu and S. Xu. Detecting rumors through modeling information propagation networks in a social media environment. *IEEE Transactions on computational social systems*, 3(2):46–62, 2016.

S. Loria, P. Keen, M. Honnibal, R. Yankovsky, D. Karesh, E. Dempsey, et al. Textblob: simplified text processing. *Secondary TextBlob: Simplified Text Processing*, 2014.

J. Ma, W. Gao, P. Mitra, S. Kwon, B. J. Jansen, K.-F. Wong, and M. Cha. Detecting rumors from microblogs with recurrent neural networks. In *IJCAI*, pages 3818–3824, 2016.

J. Ma, W. Gao, and K.-F. Wong. Detect rumor and stance jointly by neural multi-task learning. In *Companion of the The Web Conference 2018 on The Web Conference 2018*, pages 585–593. International World Wide Web Conferences Steering Committee, 2018a.

J. Ma, W. Gao, and K.-F. Wong. 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)*, volume 1, pages 1980–1989, 2018b.

H. B. Mann and D. R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, pages 50–60, 1947.

P. T. Metaxas, E. Mustafaraj, K. Wong, L. Zeng, M. O'Keefe, and S. Finn. Do retweets indicate interest, trust, agreement? *arXiv preprint arXiv:1411.3555*, 2014.

G. A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

A. Moschitti. A study on convolution kernels for shallow semantic parsing. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 335. Association for Computational Linguistics, 2004.

T. Nasukawa and J. Yi. Sentiment analysis: Capturing favorability using natural language processing. In *Proceedings of the 2nd international conference on Knowledge capture*, pages 70–77. ACM, 2003.

P. Ozturk, H. Li, and Y. Sakamoto. Combating Rumor Spread on Social Media: The Effectiveness of Refutation and Warning. doi: 10.1109/HICSS.2015.288. URL https://pdfs.semanticscholar.org/1121/ac6e4848aef1bb05059631f09b228b9747f3.pdf.

E. Pavlick and J. Tetreault. An empirical analysis of formality in online communication. *Transactions of the Association of Computational Linguistics*, 4(1):61–74, 2016.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

V. Prabhakaran, O. Rambow, and M. Diab. Automatic committed belief tagging. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 1014–1022. Association for Computational Linguistics, 2010.

V. Prabhakaran, T. By, J. Hirschberg, O. Rambow, S. Shaikh, T. Strzalkowski, J. Tracey, M. Arrigo, R. Basu, M. Clark, et al. A new dataset and evaluation for belief/factuality. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, pages 82–91, 2015.

V. Qazvinian, E. Rosengren, D. R. Radev, and Q. Mei. Rumor has it: Identifying misinformation in microblogs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1589–1599. Association for Computational Linguistics, 2011a.

V. Qazvinian, E. Rosengren, D. R. Radev, and Q. Mei. Rumor has it: Identifying misinformation in microblogs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1589–1599. Association for Computational Linguistics, 2011b.

L. S. Quanzhi Li, Qiong Zhang. eventai at semeval-2019 task 7: Rumor detection on social media by exploiting content, user credibility and propagation information. In *Proceedings of the Thirteenth International Workshop on Semantic Evaluation*, pages 851–855, 2019.

O. Rambow, T. Yu, A. Radeva, S. Hamidian, A. Fabbri, D. Ghosh, C. Hidey, T. Peng, M. Diab, K. McKeown, et al. The columbia-gwu system at the 2016 tac kbp best evaluation. In *Proceedings of the 2016 NIST TAC KBP Workshop*, 2016.

S. Y. Rieh and N. J. Belkin. Understanding judgment of information quality and cognitive authority in the www. In *Proceedings of the 61st annual meeting of the american society for information science*, volume 35, pages 279–289, 1998.

A. Ritter, S. Clark, O. Etzioni, et al. Named entity recognition in tweets: an experimental study. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1524–1534. Association for Computational Linguistics, 2011.

R. L. Rosnow and E. K. Foster. Rumor and gossip research. *Psychological Science Agenda*, 19(4):1–2, 2005.

C. L. D. Y. . Ruoyao Yang, Wanying Xie. Blcu nlp at semeval-2019 task 7: An inference chain-based gpt model for rumour evaluation. *Proceedings of the Thirteenth International Workshop on Semantic Evaluation*, 2019.

M. D. Sardar Hamidian. Gwu nlp at semeval-2019 task 7: Hybrid pipeline for rumour veracity and stance classification on social media. *Proceedings of the Thirteenth International Workshop on Semantic Evaluation*, 2019a.

M. T. D. Sardar Hamidian. Detecting rumors and their spread on social media. In *Proceedings of the 23nd Conference on Computational Natural Language Learning (submitted)*, 2019b.

R. Saurí and J. Pustejovsky. Factbank: a corpus annotated with event factuality. *Language resources and evaluation*, 43(3):227, 2009.

E. Seo, P. Mohapatra, and T. Abdelzaher. Identifying rumors and their sources in social networks. In *Ground/Air Multisensor Interoperability, Integration, and Networking for Persistent ISR III*, volume 8389, page 83891I. International Society for Optics and Photonics, 2012a.

E. Seo, P. Mohapatra, and T. Abdelzaher. Identifying rumors and their sources in social networks. In *Ground/Air Multisensor Interoperability, Integration, and Networking for Persistent ISR III*, volume 8389, page 83891I. International Society for Optics and Photonics, 2012b.

K. Shu, S. Wang, and H. Liu. Exploiting tri-relationship for fake news detection. *arXiv preprint arXiv:1712.07709*, 2017.

C. Silverman. Verification handbook. 2013.

C. Silverman. This analysis shows how viral fake election news stories outperformed real news on facebook. *BuzzFeed News*, 16, 2016.

R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

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

T. Takahashi and N. Igata. Rumor detection on twitter. In *Soft Computing and Intelligent Systems (SCIS) and 13th International Symposium on Advanced Intelligent Systems (ISIS), 2012 Joint 6th International Conference on*, pages 452–457. IEEE, 2012.

K. Thomas, C. Grier, D. Song, and V. Paxson. Suspended accounts in retrospect: an analysis of twitter spam. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, pages 243–258. ACM, 2011.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

S. Vosoughi. *Automatic detection and verification of rumors on Twitter*. PhD thesis, Massachusetts Institute of Technology, 2015a.

S. Vosoughi. *Automatic detection and verification of rumors on Twitter*. PhD thesis, Massachusetts Institute of Technology, 2015b.

S. Vosoughi, D. Roy, and S. Aral. The spread of true and false news online. *Science*, 359(6380):1146–1151, 2018.

W. Y. Wang. " liar, liar pants on fire": A new benchmark dataset for fake news detection. *arXiv preprint arXiv:1705.00648*, 2017.

C. N. Wathen and J. Burkell. Believe it or not: Factors influencing credibility on the web. *Journal of the Association for Information Science and Technology*, 53(2): 134–144, 2002.

B. E. Weeks and R. K. Garrett. Electoral consequences of political rumors: Motivated reasoning, candidate rumors, and vote choice during the 2008 us presidential election. *International Journal of Public Opinion Research*, 26(4):401–422, 2014.

G. Werner, V. Prabhakaran, M. Diab, and O. Rambow. Committed belief tagging on the factbank and lu corpora: A comparative study. In *Proceedings of the Second Workshop on Extra-Propositional Aspects of Meaning in Computational Semantics (ExProM 2015)*, pages 32–40, 2015a.

G. Werner, V. Prabhakaran, M. T. Diab, and O. Rambow. Committed belief tagging on the factbank and lu corpora: A comparative study. 2015b.

J. Wiebe, T. Wilson, and C. Cardie. Annotating expressions of opinions and emotions in language. *Language resources and evaluation*, 39(2-3):165–210, 2005.

B. Wu and H. Shen. Analyzing and predicting news popularity on twitter. *International Journal of Information Management*, 35(6):702–711, 2015.

Y. Yang and T. Hospedales. Deep multi-task representation learning: A tensor factorisation approach. *arXiv preprint arXiv:1605.06391*, 2016.

F. Yu, Q. Liu, S. Wu, L. Wang, and T. Tan. Attention-based convolutional approach for misinformation identification from massive and noisy microblog posts. *Computers & Security*, 83:106–121, 2019.

L. Zeng, K. Starbird, and E. S. Spiro. Rumors at the speed of light? modeling the rate of rumor transmission during crisis. In *2016 49th Hawaii International Conference on System Sciences (HICSS)*, pages 1969–1978. IEEE, 2016.

A. Zubiaga, M. Liakata, and R. Procter. Learning reporting dynamics during breaking news for rumour detection in social media. *arXiv preprint arXiv:1610.07363*, 2016a.

A. Zubiaga, M. Liakata, R. Procter, G. W. S. Hoi, and P. Tolmie. Analysing how people orient to and spread rumours in social media by looking at conversational threads. *PloS one*, 11(3):e0150989, 2016b.

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