# Assessing Trust and Veracity of Data in Social Media

**Sarah Alkhodair**

A Thesis

In

The Concordia Institute

For

Information Systems Engineering

Presented in Partial Fulfillment of the Requirements

For the Degree of

Doctor Of Philosophy (Information and Systems Engineering) at

Concordia University

Montréal, Québec, Canada

February 2019

# CONCORDIA UNIVERSITY

## SCHOOL Of GRADUATE STUDIES

This is to certify that the thesis prepared

By: Sarah Alkhodair

Entitled: Assessing Trust and Veracity of Data in Social Media

and submitted in partial fulfillment of the requirements for the degree of

Doctor Of Philosophy (Information and Systems Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

| | |
|---|---|
| Dr. Rastko Selmic | Chair |
| Dr. Mounir Boukadoum | External Examiner |
| Dr. Otmane Ait Mohamed | External to Program |
| Dr. Yong Zeng | Examiner |
| Dr. Nizar Bouguila | Examiner |
| Dr. Benjamin C. M. Fung | Thesis Co-supervisor |
| Dr. Rachida Dssouli | Thesis Co-supervisor |

Approved by

Dr. Chadi Assi, Graduate Program Director

April 8, 2019

Dr. Amir Asif, Dean
Gina Cody School of Engineering & Computer Science

# Abstract

**Assessing Trust and Veracity of Data in Social Media**

**Sarah Alkhodair, Ph.D.**

**Concordia University, 2019**

Social media highly impacts our knowledge and perception of the world. With the tremendous amount of data that is circulating in social media and initiated by a vast number of users from all over the world, extracting useful information from such data and assessing its veracity has become much more challenging. Data veracity refers to the trustworthiness and certainty of data. The challenges of handling textual data in social media have raised the need for efficient tools to extract, understand, and assess the veracity of information circulating in social media at a given time. In this thesis, we present three research problems to address major challenges of handling textual data in social media.

First, overwhelming the user with huge volumes of short, noisy, and unstructured textual data complicates the task of understanding what topics are discussed by users in micro-blogging websites. Topic models were proposed to automatically learn a set of keywords that better describe each topic covered by a large corpus of text documents to enable fast and effective browsing and exploration of its contents. However, in order for the results of topic modeling algorithms to be useful, these results have to be interpretable. Applying topic models to social media data to get meaningful results is not a trivial task. In this thesis, we study the problem of improving interpretation of topic modeling of micro-posts in social media. We propose a new method that incorporates topic modeling, a lexical database, and the set of hashtags available in the corpus of micro-posts to produce a higher quality representation of each extracted topic. Extensive experiments on two real-life datasets collected from Twitter show that our method outperforms the state-of-the-art model in terms of perplexity, topics' coherence, and their quality.

Second, the nature and flexibility of social media facilitate the process of posting unverified information, especially during the rapid diffusion of breaking news. Efficiently detecting and acting upon unverified breaking news rumors throughout social media is of high importance to minimizing their harmful effect. However, detecting them is not a trivial task. They belong to unseen topics or events that are not covered in the training dataset. In this thesis, we study the problem of assessing the veracity of information contained in micro-posts regarding emerging stories and topics of breaking news. We propose a new approach that jointly learns word embeddings and trains a neural network model with two different objectives to automatically identify unverified micro-posts spreading in social media during breaking news. Extensive experiments on real-life datasets show that our proposed model outperforms the state-of-the-art classifier as well as other baseline classifiers in terms of precision, recall, and F1.

Finally, the uncertainty and chaos associated with hot and sensitive breaking news and emergencies facilitate the explosive spread of high-engaging breaking news rumors that might be extremely damaging. In such a case, authorities have to prioritize the rumors verification process and act upon high-engaging breaking news rumors quickly to reduce their damaging consequences. However, this is an extremely challenging task. In this thesis, we study the problem of identifying rumors micro-posts that are most likely to become viral and achieve high engagement rates among recipients in social media during breaking news. We propose a multi-task neural network to jointly learn the two tasks of breaking news rumors detection and breaking news rumors popularity prediction. Extensive experiments on real-life datasets show that the performance of our joint learning model outperforms other baseline classifiers in terms of precision, recall, and F1 and is capable of identifying high-engaging breaking news rumors with high accuracy.

# Acknowledgments

First and foremost, I am very grateful to *Allah* Almighty for completing my doctoral research. Alhamdulillah for his countless blessings and infinite grace.

I would like to express my sincerest gratitude and appreciation to my supervisor, *Dr. Benjamin C. M. Fung*, whose precious guidance, efforts, valuable advice, and continuous follow-up have had a significant impact on the production of this thesis. It has been an honor working with you. You are a great source of knowledge and inspiration and a role model for humility and patience. I could never ask for a better supervisor. I am deeply grateful for all the things I have learned from you throughout these past years.

A great thank you from the heart for all my friends and family in Canada, USA, and Saudi Arabia. I am especially grateful for your continuous support, care, and encouragement. You have always believed in me and pushed me toward achieving my goals. I am blessed to have you in my life. I would also like to extend my thanks to the Saudi Cultural Bureau in Canada for the financial and academic support throughout this journey.

Finally, this thesis is dedicated to *Zainab and Abdulghani*, my dear parents. Thank you for always being there for me and for surrounding me with your blessings and sweet prayers. Thank you for your generous love and continuous support. Thank you for always having trust in me. The dedication extends to include my siblings: *Haifaa, Reem, Nawaf, Abeer, Fawaz*, and especially my companion in this journey, my beloved brother *Rayan*. I could not achieve this without you. Despite all the circumstances and difficulties I have experienced, I always find you standing by my side. Thank you for being in my life.

# Contents

**List of Figures** x

**List of Tables** xii

**Chapter 1 Introduction** 1

1.1 Objectives . . . 3

1.2 Contributions . . . 4

1.2.1 Improving Interpretations of Topic Modeling in Microblogs . . . 4

1.2.2 Detecting Breaking News Rumors of Emerging Topics in Social Media . . . 4

1.2.3 Identifying High-engaging Breaking News Rumors in Social Media . . . 5

1.3 Thesis Organization . . . 5

**Chapter 2 Literature Review** 7

2.1 Topic Models . . . 7

2.1.1 Topic Models for (Average-length) Text Documents . . . 7

2.1.2 Topic Models for Short Text Documents . . . 8

2.1.3 Nonparametric Topic Models . . . 9

2.1.4 Hashtags in Topic Models . . . 9

2.1.5 WordNet in Topic Models . . . 10

2.2 Rumor Detection and Analysis . . . 11

2.2.1 Rumor Detection . . . 11

2.2.2 Rumor Tracking . . . 12

2.2.3 Rumor Stance Classification . . . 12

2.2.4 Rumor Veracity Classification . . . 12

2.3 Fake News Detection . . . 13
2.3.1 Detection of Check-worthy News Articles . . . 13
2.3.2 News Articles Veracity Prediction . . . 14
2.4 Popularity Prediction . . . 14
2.4.1 Popularity Prediction in Social Media Based on Early Observations . . . 14
2.4.2 Popularity Prediction of News Articles Based on Topics (Contents) Similarities . . . 15

**Chapter 3 Preliminaries** **16**
3.1 Topic Models . . . 16
3.1.1 Latent Dirichlet Allocation (LDA) . . . 17
3.1.2 Author-Topic Model (AT) . . . 19
3.2 Artificial Neural Networks (NN) . . . 20
3.2.1 A Perceptron in Artificial Neural Networks . . . 20
3.2.2 A Simple Neural Network Architecture . . . 21
3.2.3 Recurrent Neural Networks (RNN) . . . 22
3.2.4 Convolutional Neural Network (CNN) . . . 23
3.3 Attention in Neural Networks . . . 25
3.3.1 Sequence-to-Sequence Language Translation with Neural Networks . . . 26
3.3.2 Basic Attention Mechanism . . . 26
3.4 Representation Learning . . . 28

**Chapter 4 Improving Interpretations of Topic Modeling in Microblogs** **30**
4.1 Introduction . . . 30
4.1.1 The Challenges . . . 31
4.1.2 Contributions . . . 33
4.2 Problem Description . . . 34
4.3 Background Information . . . 36
4.3.1 Twitter-LDA . . . 36
4.3.2 Inference Using Gibbs Sampling . . . 36

4.4 Methodology . . . 38
4.4.1 Gibbs Sampling . . . 38
4.4.2 Count Boosting . . . 39
4.4.3 Posterior Distribution Calculation . . . 41
4.4.4 Improved Topic Clustering . . . 41
4.5 Experiments . . . 42
4.5.1 Datasets . . . 42
4.5.2 Data Preprocessing . . . 42
4.5.3 Perplexity . . . 43
4.5.4 Topics Coherence . . . 43
4.5.5 Users' Evaluation . . . 46
4.5.6 Topical Trends Over Time . . . 49
4.5.7 Users' Interests Over Time . . . 50
4.5.8 Customized Taxonomy . . . 51
4.6 Conclusion . . . 54

**Chapter 5 Detecting Breaking News Rumors of Emerging Topics in Social Media 55**

5.1 Introduction . . . 55
5.2 Deep Learning Model for Breaking News Rumors Detection . . . 59
5.2.1 Problem Statement . . . 59
5.2.2 Proposed Model . . . 60
5.3 Experiment . . . 63
5.3.1 Datasets . . . 63
5.3.2 Baselines and Feature Sets . . . 64
5.3.3 Experimental Settings . . . 64
5.3.4 Evaluation Measures . . . 65
5.3.5 Results . . . 66
5.3.6 Case Studies . . . 73
5.4 Limitation . . . 76

**Chapter 6 Identifying High-engaging Breaking News Rumors in Social Media** **78**

6.1 Introduction . . . . . 78

6.2 Joint Learning Model for Identifying High-engaging Breaking News Rumors . . . 82

6.2.1 Problem Statement . . . . . 83

6.2.2 Proposed Model . . . . . 83

6.3 Experiments . . . . . 88

6.3.1 Dataset . . . . . 89

6.3.2 Feature Sets . . . . . 89

6.3.3 Experimental Settings . . . . . 91

6.3.4 Experimental Results . . . . . 91

**Chapter 7 Conclusions** **100**

**Chapter 8 Future Directions** **103**

**Bibliography** **105**

# List of Figures


Figure 3.1 Plate notation of the Latent Dirichlet Allocation (LDA) topic model . . . . 18

Figure 3.2 Plate notation of the Author-Topic (AT) model . . . . . 19

Figure 3.3 A perceptron with three inputs in artificial neural networks . . . . . 21

Figure 3.4 A simple neural network architecture . . . . . 22

Figure 3.5 A simple recurrent neural network (RNN) architecture . . . . . 23

Figure 3.6 A simple convolutional neural network (CNN) architecture . . . . . 24

Figure 3.7 An example of applying a 2x2 max-filter with a stride of 2 on a 2-dimensional input matrix of size 4x4 . . . . . 25

Figure 3.8 A simple example of a seq2seq model translating the English sentence "I like listening to music" to Arabic . . . . . 26

Figure 3.9 A seq2seq neural network model with attention . . . . . 28

Figure 3.10 A visualization example of word embeddings in a 3-dimensional embedding space. . . . . 29

Figure 4.1 Plate notation of the Twitter-LDA topic model . . . . . 36

Figure 4.2 An overview of the proposed model for improving interpretations of topic modeling in microblogs . . . . . 38

Figure 4.3 Perplexity on OffAcc dataset . . . . . 44

Figure 4.4 Topical trends over time . . . . . 50

Figure 4.5 StyleForum's tweets over time . . . . . 51

Figure 4.6 LuckyMagazine's tweets over time . . . . . 52

Figure 4.7 Prada's tweets over time . . . . . 52

Figure 4.8 YSL's tweets over time . . . . . 53

Figure 4.9 Examples of WordNet customization . . . . . 54

Figure 5.1 The proposed breaking news rumors detection model. A micro-post $mp$ is first tokenized into a sequence of words $mp = \langle w_1, ..., w_T \rangle$. Next, the word2vec model converts the sequence of words into a sequence of vectors $x = \langle x_1, \ldots, x_T \rangle$ and passes it through weighted connections to the LSTM-RNN model. Finally, the LSTM-RNN model predicts the class as the output vector at the last time step $T$ . . 60

Figure 6.1 The proposed joint learning model for identifying high-engaging breaking news rumors in social media . . . . . 83

Figure 6.2 Receiver Operating Characteristic (ROC) curves for the two variations of the proposed joint learning model showing the ROC curves and the Area Under Curve (AUC) scores for each of the five runs and the Mean $\pm$ variance of the AUC scores across all five runs. Figure 6.2.a shows the obtained results for the Multi-task CNN-based model and Figure 6.2.b shows the obtained results for the Multi-task CNN-Attn-based model . . . . . 94

# List of Tables


Table 4.1 Two sets of keywords representing two different topics . . . . . 32

Table 4.2 Two sets of keywords representing the topic “Brands” . . . . . 33

Table 4.3 Datasets statistics . . . . . 42

Table 4.4 Average percentage of the correct answers for both models . . . . . 48

Table 4.5 Examples of topics in OffAcc and FashionKW datasets . . . . . 48

Table 4.6 Improvements of the sets of keywords resulting from Twitter-LDA-WNH over Twitter-LDA . . . . . 49

Table 4.7 Different interpretation of two sets of keywords resulting from Twitter-LDA and Twitter-LDA-WNH . . . . . 49

Table 5.1 Percentages of rumors and non-rumors tweets in the PHEME datasets . . . . 64

Table 5.2 Content-based and social-based features . . . . . 65

Table 5.3 Micro-averaged precision (p), recall (R), and F1 scores of detecting rumors and non-rumors across all five runs for baseline classifiers and our proposed model 67

Table 5.4 Micro-averaged mean $\pm$ variance of precision (p), recall (R), and F1 scores of detecting rumors and non-rumors across all five runs for our proposed model using other syntactic features . . . . . 68

Table 5.5 Micro-averaged mean $\pm$ variance of precision (p), recall (R), and F1 scores of detecting rumors and non-rumors across all five runs for our proposed model under different settings of training word2vec model . . . . . 70

Table 5.6 Precision scores of different classifiers before and after using social-based features associated with each dataset . . . . . 70

Table 5.7 Importance scores of each of the features in each dataset measured as the gain ratio between this feature and the true class label . . . . . 71

Table 5.8 Standard Deviation values of social-based features for the PHEME datasets . 72
Table 5.9 The classification performance of our model on a real-life breaking news case study in terms of precision (p), recall (R), and F1 . . . . . 74
Table 5.10 Examples of tweets collected from real-life breaking news and how it was classified by our model. . . . . 74
Table 5.11 The classification performance of our model on a real-life multiple breaking news case study in terms of precision (p), recall (R), and F1 . . . . . 75
Table 6.1 Percentages of rumors and non-rumors tweets in the PHEME datasets . . . . 89
Table 6.2 List of the stylometric features . . . . . 90
Table 6.3 The list of primal emotions and the associated emotional triggers emoticons used in social media . . . . . 90
Table 6.4 Mean $\pm$ variance of the precision (P), recall (R), and F1 scores of identifying high-engaging breaking news rumors using different features sets and variations of our proposed joint learning model . . . . . 92
Table 6.5 Precision (P), recall (R), and F1 scores of the two tasks of breaking news rumors detection and breaking news rumors popularity prediction across all five runs for the single-task baseline classifiers and our proposed joint learning models using different features sets . . . . . 98
Table 6.6 Best performing feature sets with each model for the single task of breaking news rumors detection in terms of precision (P), recall (R), and F1 . . . . . 99
Table 6.7 Best performing feature sets with each model for the single task of breaking news rumors popularity prediction in terms of precision (P), recall (R), and F1 . . . 99

# Chapter 1

# Introduction

The explosive growth of the Internet and the more affordable smartphone and mobile data plans have resulted in the wide adoption of social media websites from users all over the world. Oxford Dictionary defines social media as "websites and applications that enable users to create and share content or to participate in social networking"[1]. Social media is also defined as "forms of electronic communication (such as websites for social networking and micro-blogging) through which users create online communities to share information, ideas, personal messages, and other content (such as videos)"[2]. According to these definitions, social media covers all existing online community-based websites and applications that facilitate content sharing, interactions, and collaborations among participants. There is a huge number of existing social media websites and applications nowadays. This number keeps growing and attracting more and more users every day. According to their latest global statistics report, *We Are Social*[3] has reported a rapid growth of social media users in 2018 to become more than 3.4 billion users around the world in September 2018 (10% more than September 2017), with an increase of "almost 1 million new users every day during the past 12 months"[4]. This suggests that nearly half the world's 7.6 billion inhabitants are now on social media, and this number is dramatically increasing over time.

[1] Source:https://en.oxforddictionaries.com/definition/social_media, Retrieved on Oct 21, 2018

[2] Source: https://www.merriam-webster.com/dictionary/social%20media, Retrieved on Oct 21, 2018

[3] Source: https://wearesocial.com/uk/, Retrieved on Nov 4, 2018

[4] Source: https://wearesocial.com/uk/blog/2018/10/the-state-of-the-internet-in-q4-2018, Retrieved on Nov 4, 2018

Social media websites highly impact people's knowledge and perception of the world. The available huge volume of textual data in these social media websites contains valuable real-time information from every corner of the globe. This volume is getting larger every day. Textual data in the form of web pages, blogs, tweets, and forums cover a vast range of topics and contain an enormous amount of information. However, this comes with a price. The fact that text in social media websites is unstructured, imprecise, uncertain, difficult to trust, initiated and spread by a huge number of users, has brought several challenges to the forefront.

First, overwhelming the user with such a volume of text data complicates the task of understanding and extracting useful information. This raises the need for having effective tools to automatically extract useful information from unstructured document collections. Topic models were proposed to solve this problem by automatically detecting the underlying semantic structure of large text document collections and providing short descriptions of such documents. Uncovering this structure facilitates browsing and exploring the collection and allows the user to effectively access documents with similar topics. However, dealing with short text documents like micro-posts in social media is challenging. This is due to the nature of text in micro-posts such as the lack of co-occurrence patterns and high sparseness. Furthermore, micro-posts are extremely noisy, and each post contains very few words, further complicating the process of extracting meaningful topics.

Second, with the explosive growth of textual data, an important question is: To which extent can we trust this data, and how can we assess its veracity? Data veracity refers to "the degree to which data is accurate, precise and trusted"[5]. Users of social media websites tend to rapidly spread breaking news and trending stories or pieces of information with no guarantee of truth or quality. Breaking news refers to an unexpected event that has just begun or is currently developing. According to the basic law of rumors [6], the more the importance and uncertainty of a topic, the more it is associated with rumors. This explains why breaking news is usually associated with many rumors, especially at the early stages of diffusion. A rumor has been defined as "a story or a statement whose truth value is unverified" [6]. Acting upon unverified information spreading throughout the social network in a timely fashion is of high importance to minimize its harmful effect. However, in order for the involved parties to verify or refute the spreading rumors fast, these

[5]Source: https://simplicable.com/new/data-veracity, Retrieved on Nov 6, 2018

rumors have to be detected first. However, this is not a trivial task.

Third, social media websites are increasingly adopted by users from all over the world as a major source of news gathering, especially during the development of breaking news and emergency situations. Important breaking news causes a state of uncertainty and anxiety to dominate society. This puts people in a low cognitive mode and encourages them to closely follow up with any information update regarding the current development of the breaking news, share such information regardless of its veracity, and act upon this information immediately [73]. In this case, the appearance and spread of new breaking news rumors throughout social media happens very fast. Therefore, after a few minutes, the damaging consequences of a high-engaging breaking news rumor are more likely to have already happened. This increases the burden of handling breaking news and emergencies by the authorities. However, not all rumors have the potential to spread in social media. Breaking news rumors that are written in a manner that ensures they achieve the highest prevalence among the recipients will potentially cause the most damage. These rumors are extremely difficult to detect, intended to touch and satisfy the emotional needs of recipients, and have the potential to become extremely viral in social media in just a few minutes. This highlights the importance to not only identify breaking news rumors, but also to predict which rumors are most likely to become viral in social media and might require immediate attention from authorities. Identifying such rumors can be extremely helpful in prioritizing the rumor verification process during breaking news to reduce potential damaging consequences.

## 1.1 Objectives

The main objectives of this thesis can be summarized as follows:

**Objective #1.** to improve interpretations of topic modeling in social media with the goal of providing a better overview of topics circulating social media websites at a given time.

**Objective #2.** to solve the problem of breaking news rumors detection with the goal of identifying unverified information spreading in social media websites during the development of breaking news and trending stories.

**Objective #3.** to detect high-engaging breaking news rumors with the goal of identifying rumors micro-posts that are most likely to achieve high engagement rates in social media during the development of breaking news, and thus require immediate attention from authorities.

## 1.2 Contributions

The key contributions of this thesis are summarized below.

### 1.2.1 Improving Interpretations of Topic Modeling in Microblogs

Topic models were proposed to detect the underlying semantic structure of large collections of text documents to facilitate the process of browsing and accessing documents with similar ideas and topics. Applying topic models to short text documents to extract meaningful topics is challenging. The problem becomes even more complicated when dealing with short and noisy microposts in Twitter. In such case, applying topic models results in topics represented by similar sets of keywords, which in turn makes the process of topic interpretation more confusing. To the best of our knowledge, this thesis is the first to propose a new method that combines an English lexical database, WordNet, along with the set of hashtags and topic models with the goal of improving the sets of keywords used to represent each topic extracted from short texts in Twitter. We emphasized the importance of different keywords to different topics based on the semantic relationships and the co-occurrences of keywords in hashtags. We also proposed a method to find the best number of topics to represent the text document collection. Our proposed approach can be used to dynamically build a customized taxonomy for a specific domain. Experiments on two real-life Twitter datasets suggest that our method performs better than the original *Twitter-LDA* [104] in terms of perplexity, topic coherence, and the quality of keywords for topic labeling.

### 1.2.2 Detecting Breaking News Rumors of Emerging Topics in Social Media

Users of social media websites rapidly spread breaking news information without considering their truthfulness, thus facilitating the spread of rumors. Efficiently detecting and acting upon breaking news rumors throughout social networks is of high importance to minimize their harmful

effect. However, detecting them is not a trivial task. They belong to unseen topics or events that are not covered in the training dataset. Most previous studies on rumor detection assume that rumors are always false and focus on long-standing rumors. In contrast, this thesis studied the problem of detecting breaking news rumors that spread in social media regardless of their truth value. In this thesis, we propose a new semi-supervised learning solution that jointly learns word embeddings and trains a recurrent neural network with two different objectives to automatically identify breaking news rumors. The proposed strategy is simple but effective to mitigate the topic shift issues in emerging breaking news. Our experiment simulates a cross-topic emerging rumor detection scenario with a real-life rumor dataset. The experimental results suggest that our proposed model outperforms state-of-the-art methods in terms of precision, recall, and F1.

### 1.2.3 Identifying High-engaging Breaking News Rumors in Social Media

High-engaging breaking news rumors are those written in a manner that ensures achievement of the highest prevalence among the recipients. Such rumors are difficult to detect, spread very fast, and can cause serious damages to society. Fortunately, the characteristics of high-engaging rumors are very much in line with the characteristics of widely popular posts in social media, in general. To the best of our knowledge, this thesis is the first to tackle the problem of identifying high-engaging breaking news rumors in social media. We propose a multi-task neural network model to *jointly* learn the two tasks of breaking news rumors detection and breaking news rumors popularity prediction in social media. The proposed model learns the salient semantic similarities among important features for identifying high-engaging breaking news rumors and separates them from the rest of the input text. Extensive experiments on five real-life datasets of breaking news suggest that our proposed model is capable of detecting breaking news rumors and predicting their future popularity and outperforms all baselines in terms of precision, recall, and F1.

## 1.3 Thesis Organization

The rest of the thesis is organized as follows.

- In Chapter 2, we provide a comprehensive review of the related state-of-the-art work in the

literature. First, we provide a review of different categories of topic modeling algorithms, then we cover related work in rumor detection and analysis followed by work in fake news detection, and finally we cover related work in popularity prediction of posts in social media.

- In Chapter 3, we provide important background knowledge. First, we provide brief descriptions of two important topic models, then we cover some basic knowledge on artificial neural networks and attention mechanisms, and finally we provide a high-level explanation of representation learning.

- In Chapter 4, we study the problem of improving the sets of keywords used to represent each topic extracted from short texts in Twitter. First, we provide a brief introduction followed by the formal definition of our problem, then we describe the proposed model to address the research problem, and finally we present our experiments and a detailed discussion of the obtained results. The results of this chapter have been published in [4].

- In Chapter 5, we study the problem of identifying unverified information spreading on social media during breaking news. First, we provide a brief introduction followed by the formal definition of our problem, then we describe the proposed model to address the research problem, and finally we present our experiments and a detailed discussion of the obtained results. This chapter has has been published in [5].

- In Chapter 6 we study the problem of identifying breaking news rumors that are expected to have high-engaging rates among recipients in social media. First, we provide a brief introduction followed by the formal definition of our problem, then we describe the proposed model to address the research problem, and finally we present our experiments and a detailed discussion of the obtained results. This chapter is currently under review in the 28th International Joint Conference on Artificial Intelligence (IJCAI 2019).

- In Chapter 7, we provide a general conclusion of the three research problems discussed in this thesis.

- Finally, in Chapter 8, we provide some future directions.

# Chapter 2

# Literature Review

This chapter provides an in-depth literature review of important related work to the focus of this thesis.

## 2.1 Topic Models

Modeling text documents has attracted a lot of attention in the past years due to the vast amount of text documents and the huge amount of useful information that can be extracted from them. Topic models have been proposed to automatically detect the underlying topical structure of large corpora of text document. The objective of applying a topic model to a corpus of text documents is to learn a set of keywords that better describe each topic covered by that corpus. Knowing the topical structure of a corpus of text documents enables fast and effective browsing and exploration of its contents. The following subsections cover important work in different categories of topic modeling.

### 2.1.1 Topic Models for (Average-length) Text Documents

One of the most well-known topic modeling algorithms is the *Latent Dirichlet Allocation* (*LDA*) proposed by Blei et al. [17]. This model is a three-level hierarchical Bayesian model that models each text document (item) as a mixture distribution of underlying topics. In addition to modeling the underlying topics of the documents, some researchers proposed to include additional information sources. A representative work in this category is the work done by Rosen-Zvi et al. [82] that jointly

models the authors' interests as well as the topics contained in the text documents collection. The proposed *Author Topic Model* (*AT*) model includes the authorship information to extend the existing generative probabilistic models. Other extensions to the existing topic models were proposed so that time is taken into consideration when modeling the text documents. The work proposed by Wang and McCallum [97] is one such example. They proposed the *Topic Over Time* (*TOT*) model that models the co-occurrence patterns of words jointly with the time. In this work, the topics as well as their meaning are treated as constants that do not change over time. The evolution captured by this model is that of the occurrence and co-occurrence patterns of the topics rather than the changes in the underlying word distribution of each topic. The work proposed by Basher and Fung [9] is another example of author topic over time model. Blei and Lafferty [14] proposed a *Dynamic Topic Model* (*DTM*) that is capable of capturing the evolution of the underlying topics in large document corpora where documents are sequentially organized. The DTM models the trajectory of each topic over the specified time span. Song et al. [89] proposed a *Hierarchical Topic Evolution Model* (*HTEM*) that is capable of providing a hierarchical organization of topics and detecting their evolution over time. Knights et al. [50] proposed a *Compound Topic Model* that builds a single model based on two distinct sets of data: one represents the past data and the other represents the current data to detect emerging topics. Liu and Turtle [58] proposed a *Realtime Interest Model* (*RIM*) that models the interests of a user in an online manner, and then uses an online ranking mechanism to rank the search results based on a user's current interests. Other topic models address the problem of ignoring the high correlation between the presences of the underlying topics, which is natural in large text corpora such as the *Correlated Topic Model* (*CTM*) proposed by Blei and Lafferty [15]. CTM extends LDA to model the correlations between the occurrences of underlying topics within the text corpora. CTM is more expressive than the LDA in that it provides a richer way of exploring and visualizing the text corpora, but with this higher flexibility the CTM model suffers from higher computational cost than LDA.

### 2.1.2 Topic Models for Short Text Documents

Many recent works focus on proposing topic models that deal with short text documents instead of traditional long text documents. This is due to the high sparseness of the short text documents

and the lack of co-occurrences patterns. In the effort to model short text documents, some authors proposed extensions to the existing models that can better handle short text documents. The work of Zhao et al. [104] is one example. They proposed a modified Author Topic Model (AT), called *Twitter-LDA*, that is capable of modeling topics in the short messages of Twitter. The proposed model is based on the observation that a single tweet usually covers only one topic. In addition to modeling each topic as a probability distribution over the vocabulary of terms and each author as a probability distribution over the set of topics, Twitter-LDA addresses the noisy nature of short messages of Twitter by introducing a background topic to capture the background terms in Twitter. This model has several advantages in modeling short text messages. In addition to finding more or comparable meaningful topics to those found by the LDA and the AT models, the Twitter-LDA's assumption that every tweet covers a single topic is more convenient when computing the tweet-level statistics. Sasaki et al. [86] proposed an online topic model that extends Twitter-LDA by enabling the ratio between the topical words and the background words to be different for every user. Furthermore, based on the *Topic Tracking Model* (*TTM*), they extend the model to allow online inference.

### 2.1.3 Nonparametric Topic Models

Another category of topic models is the nonparametric models, based on the *Hierarchal Dirichlet Processes (HDP)* [94], where the user does not need to specify the number of topics in advance. Researchers also proposed to include additional information such as authorship [25], time [29], and word embeddings [10]. Our proposed method in Chapter 4 is a parametric topic model that requires the number of topics to be provided in advance. It provides a mechanism to automatically adjust the number of topics to be presented to the user and, at the same time, provides the user with the flexibility of choosing the number of topics that best serves his/her needs.

### 2.1.4 Hashtags in Topic Models

Several works also proposed the inclusion of *hashtags* in topic models. She and Chen [87], for example, proposed a *Topic Model Hashtag Recommendation on Twitter* (*TOMOHA*). The model assigns a hashtag distribution for every topic and uses the trained model to recommend the most

probable hashtags to be included in a new tweet. Similarly, Godin et al. [32] proposed to use topic models for hashtag recommendation. They applied LDA to detect the underlying topics first. Then, for every new tweet, they infer the topic distribution and recommend hashtags based on the top words in that topic distribution. Another work that includes hashtags in topic models is the one proposed by Ma et al. [64]. The *Tag-Latent Dirichlet Allocation* extends LDA to include the observed hashtags. In this model, every observed hashtag is modeled as a distribution over topics in the effort to better understand what different hashtags mean and how they are correlated semantically. These works include hashtags to make recommendations and to understand the semantic correlations between the hashtags. On the other hand, our method in Chapter 4 proposes to use the set of hashtags to improve the set of keywords used to represent each topic.

### 2.1.5 WordNet in Topic Models

Several works proposed the employment of *WordNet* in Topic Model for different purposes. The work in [60] proposed a *WordNet-enhanced Topic Model* wherein they used WordNet for concepts construction as a preprocessing step. Those concepts were then treated by the topic model as observed data. Musat et al. [74] covered a similar idea but instead of using WordNet to construct the concepts in a preprocessing step, they first applied LDA and then employed WordNet as a post-processing step to build a conceptual ontology. The topical subtree was then used to determine the related concepts and the outliers for a given topic. Newman et al. [75] proposed to use WordNet to evaluate the results of LDA. They proposed a score function that is based on the similarity between every pair of terms for a given topic. Boyd-Graber et al. [18] proposed a topic model for word sense disambiguation. Their work is an extension of LDA that includes the word sense as a hidden variable. In Chapter 4, we proposed to use WordNet as an intermediate step in the inference process of the posterior distribution to improve the extracted topics based on the relationships between terms in WordNet. We also proposed to build a customized version of WordNet and use it to evaluate the results of clustering the topics.

## 2.2 Rumor Detection and Analysis

The nature of the textual data and how fast it spreads in social media raised the need of building tools capable of automatically identifying rumors and assessing their veracity. Work in this field falls into one of four categories: rumor detection, rumor tracking, rumor stance classification, and rumor veracity classification [108]. Although approaches proposed in the last three categories are beneficial for handling long-standing rumors, their applicability to handle breaking news rumors of emerging topics, which is the focus of Chapter 5 and Chapter 6 in this thesis, might be limited. They are based on the assumption that the rumor is already known and a stream of micro-posts about it is available. They skip the first and most important step in the process of detecting and analyzing rumors, which is identifying these rumors in the first place.

To illustrate the difference, this section briefly describes each category and covers some representative work.

### 2.2.1 Rumor Detection

Rumor detection is the first and most important task. The goal is to identify unverified information spreading across social media. Yet, there has been very little work in this category. The first rumor detection method was proposed by Zhao et al. [105]. The proposed method starts by identifying "signal tweets". These tweets are then grouped into different clusters, each representing a rumor. Next, each cluster is summarized and the summary is used to retrieve more related tweets. Finally, the clusters are ranked by their likelihood of being rumors. Their method is based entirely on using a list of user-defined regular expressions to identify the "signal tweets". Thus, for their method to better handle new, unseen stories, this list needs to be revised periodically. Zubiaga et al. [107] proposed a rumor detection model based on a sequential classifier. The proposed model classifies a tweet as a rumor or non-rumor based on previously encountered data. This method achieves higher performance than the previous work [105]. However, it suffers from the cold start problem [107]. In Chapter 5, we propose a semi-supervised model that employs representation learning and deep learning models to learn and exploit the lexical and temporal features of rumor micro-posts [5]. The proposed model outperforms the state-of-the-art model [107] in detecting breaking news rumors in

terms of accuracy. In Chapter 6, we propose a multi-task model to detect breaking news rumors and predict which of them are most likely to become viral in social media and, therefore, need immediate attention.

### 2.2.2 Rumor Tracking

Rumor tracking also gains limited attention in the literature. The research problem here is to determine if a given micro-post is related to one of the rumors known in advance. The first work in this category was proposed in [79]. The authors proposed a supervised machine learning approach to judge the relevance of new tweets to the known set of rumors. In [39], the authors proposed a tweet latent vector representation of tweets and used the *Semantic Textual Similarity (STS)* [37] to assess the relevance of new tweets to the known rumors.

### 2.2.3 Rumor Stance Classification

Rumor stance classification is a well-studied problem: given a set of micro-posts related to a rumor, classify the orientation expressed in the text of a micro-post as supporting, denying, or questioning the rumor. Most existing works in this category are supervised learning where a predictive model is trained based on different features. The first and most cited work is [79], which proposed several content-based, network-based, and micro-blog-based features. There is a family of works [39, 57, 61, 107] that focus on introducing new features and studying their performance with different classifiers.

### 2.2.4 Rumor Veracity Classification

Rumor Veracity classification is another well-studied problem in the literature. Most existing works in this category [52, 57, 62, 100] also employ supervised learning where predictive models are trained based on different features to determine the veracity of rumors spreading in social media. Unsupervised methods were recently proposed to tackle this problem, including *recurrent neural networks (RNN)* [63] and *recurrent neural networks with attention mechanism* [21, 47]. This problem is sometimes referred to as rumor detection, where authors of such works adopt an invalid

definition of rumors as being "false" pieces of information. Thus, the goal of these proposed methods is predicting the truth value of an unverified story rather than detecting these unverified stories. On the other hand, our proposed models in Chapter 5 and Chapter 6 aim at identifying micro-posts containing unverified breaking news information and flagging these micro-posts as rumors.

## 2.3 Fake News Detection

In this section, we provide an overview of a closely related problem to rumor detection and analysis known as fake news detection. Fake news refers to news articles that are intentionally written to contain false information. Work in this field can be broadly categorized into two families: identifying check-worthy news articles and predicting the veracity of these articles. This section highlights some of the recent contributions in each of these categories.

### 2.3.1 Detection of Check-worthy News Articles

Identifying check-worthy news articles refers to the task of detecting news articles that contain important information yet to be verified. Hassan et al. [40] tackled this problem by proposing a supervised learning method. They first constructed a dataset of spoken sentences labeled as non-factual sentence, unimportant factual sentence, or important factual sentence. Next, *Naive Bayes (NB)*, *Support Vector Machine (SVM)*, and *Random Forest (RF)* multi-class classifiers were used to identify sentences belonging to each of the three categories. The problem of identifying check-worthy news articles is similar to the problem of rumors detection studied in Chapter 5 and Chapter 6. However, this line of work deals with news articles written in a structured way with full English sentences to discuss one or more well-defined topics. Thus, its applicability to the short, unstructured, and noisy text of micro-posts in social media might be limited. More importantly, the proposed method depends on a pre-built database of factual and non-factual English sentences. Therefore, it may not be able to successfully cope with the emerging facts and topics of breaking news rumors.

### 2.3.2 News Articles Veracity Prediction

Predicting the veracity of news articles is highly related to the problem of rumors veracity classification. It is also a well-studied problem. The task is to determine whether or not a check-worthy news article is fake. Different machine learning algorithms have been used to tackle this problem such as SVM, *bi-directional long short-term memory networks (Bi-LSTM)*, *convolutional neural networks (CNN)* [96], RNNs [84], *homogeneous credibility networks* [46], and *heterogeneous credibility networks* [45]. Work in this category assumes that the check-worthy articles are already known and aims at predicting its veracity. In contrast, our work in Chapter 5 and Chapter 6 aims at identifying micro-posts that contain unverified information. The goal is to flag rumors micro-posts during the rapid diffusion of breaking news to minimize their harmful consequences.

## 2.4 Popularity Prediction

Predicting the popularity and diffusion of information in social media has increasingly attracted a great deal of attention in recent years. The main task here is to build a prediction model that is capable of estimating the future popularity of a post.

### 2.4.1 Popularity Prediction in Social Media Based on Early Observations

Most existing work investigates this problem by predicting the popularity of a micro-post in social media by observing its popularity for some time after it is posted. For example, Li et al. [54] proposed to use early views along with the attractivness of a video to predict its future popularity. Similarly, Zaman et al. [101] proposed a Bayesian approach that uses network information as well as the time path of previous retweets as features to predict the future popularity of a tweet. Also, Yan et al. [99] proposed a *Spatial and Temporal Heterogeneous Bass model (STH-Bass)* that uses information gathered after the first day of posting a tweet to predict its future popularity. Zhao et al. [103] proposed a *Self-Exciting Point Process Model (SEISMIC)* to predict the future retweet intensity of a tweet as the product of its "infectivity" and the excitation effect of all of its previous retweets. Similarly, Kobayashi and Lambiotte [51] also proposed a *Time-Dependent Hawkes Process (TiDeH)* model to predict the tweet future intensity based on the infectivity function of a

tweet. Mishra et al. [70] proposed a model that combines features-based approaches with the point process models to predict the future popularity of a tweet based on its previous retweets. Xie et al. [98] proposed a model that uses early observations to predict the future popularity of Tumblr posts. Recently, Chen et al. [20] proposed a new model to predict the future dynamic and popularity of a tweet by leveraging the observed dynamics of its retweeting of the first two hours after it is posted. All these popularity prediction methods predict the future popularity of a micro-post based on the early observations of its dynamics in social media. This requires monitoring its popularity for some time after it is posted to gather sufficient observations for a reliable prediction. Such methods are not applicable when dealing with breaking news and emergency situations. People tend to spread breaking news rumors and act upon them immediately, which can cause extreme damage in just a few minutes. In contrast, our proposed model in Chapter 6 does not need a collection of early observations to predict the future popularity of a rumor micro-post.

### 2.4.2 Popularity Prediction of News Articles Based on Topics (Contents) Similarities

There are other lines of research that investigate the problem of predicting the popularity of a news article prior to publishing. For example, Bandari et al. [8] used Twitter data to predict the future popularity of a news article before it is published. Similarly, Abbar et al. [1] proposed to predict the popularity of a new news article based on the recent popularity of its topic and similar articles. In addition to dealing with news articles rather than the short text of social media micro-posts, this line of work requires a collection of related posts, similar articles, or similar topics that does not exist in the case of breaking news. In contrast, our proposed model in Chapter 6, does not need a collection of related posts or topics to predict the future popularity of a rumor micro-post.

# Chapter 3

# Preliminaries

In this chapter, we provide some important background knowledge on topic models, artificial neural networks, attention mechanism, and representation learning.

## 3.1 Topic Models

The volume of available text documents is getting larger every day; therefore, a simple search may result in millions of text documents and articles. Overwhelming the user with such a volume of text data complicates the task of understanding and extracting useful information. This problem requires a solution that allows users to effectively search and explore collections of text documents in a structured way to facilitate accessing documents with similar ideas, i.e., topics. Topic models have been proposed to automatically detect the underlying semantic structure of text document collections. Knowing the topical structure of the collection enables effective browsing and information exploration. The main idea behind topic models is to describe text documents using short descriptions in such a way that allows handling of large collections of text documents and, at the same time, preserving the required statistical relationships.

Several topic models have been proposed in the last several years to handle short text documents in addition to traditional text documents. In the following subsections, we provide brief descriptions of two important topic models: the *Latent Dirichlet Allocation* and the *Author-Topic* model.

### 3.1.1 Latent Dirichlet Allocation (LDA)

The Latent Dirichlet Allocation (LDA) topic model proposed by Blei et al. [17] is one of the most well-known topic modeling algorithms and serves as the foundation of many other topic models, some of which will be described in the following subsections. LDA models each document as a probability distribution over topics and every topic as a probability distribution over a fixed vocabulary of terms. The model assumes a fixed number of topics for the entire collection of text documents. This set of topics is covered by each document with different proportions.

LDA adopts the hidden variable model of documents. A hidden variable model is a structured distribution where there is an interaction between the hidden random variables and the observed data. In this model, a hidden structure is posited in the observed data and then the posterior probabilistic inference is used to discover this structure. In LDA, words in text documents are the observed data while the latent topic structure of the document collection is represented by the hidden variables. Given the text documents in the collection, the posterior distributions of the hidden variables determine the hidden topical structure of that collection. To further clarify this interaction between the observed data, i.e., text documents, and the hidden variables, i.e., the latent topical structure of the document collection, one can refer to the LDA's probabilistic generative process. The generative process was described by Blei and Lafferty [16] as "the imaginary random process that is assumed to have produced the observed data". Formally, let $D$ be the corpus of text documents, $K$ be the set of topics in $D$, and $V$ be the vocabulary of terms. Let $\Phi$ be the distributions of topics over vocabulary and $\Psi$ be the distributions of documents over topics. $\beta$ and $\eta$ denote the hyperparameters of LDA and $g$ denotes the topic assignment. Then, the generative process for LDA is shown in Algorithm 1.

A graphical representation of LDA, known as the plate notation, is shown in Figure 3.1. In this representation each node represents a random variable and each edge represents the dependence between variables. Shaded nodes and unshaded nodes are used to differentiate between observed random variables and hidden random variables, respectively. Boxes represent replications. The same plate notation will be used throughout this chapter to represent different topic models.

**Algorithm 1** The generative process of LDA

**Input** A corpus of text documents $D$

**Output** Distributions of topics over vocabulary $\Phi$ and distributions of documents over topics $\Psi$.

**for** each topic $k_i$ indexed by $i = 1$ to $|K|$ **do**
    Draw a distribution over the vocabulary of terms $\phi^{k_i} \sim Dir\,(\beta)$.
**end for**
**for** each document $d_j$ indexed by $j = 1$ to $|D|$ **do**
    Draw a vector of topic proportions $\psi^{d_j} \sim Dir\,(\eta)$.
    **for** each word $w_n$ indexed by $n = 1$ to $|d_j|$ **do**
        Draw a topic assignment $g_{d_j,w_n} \sim Mult\left(\psi^{d_j}\right), g_{d_j,w_n} \in K$.
        Draw a term $v_{d_j,w_n} \sim Mult\left(\phi^{g_{d_j,w_n}}\right), v_{d_j,w_n} \in V$.
    **end for**
**end for**

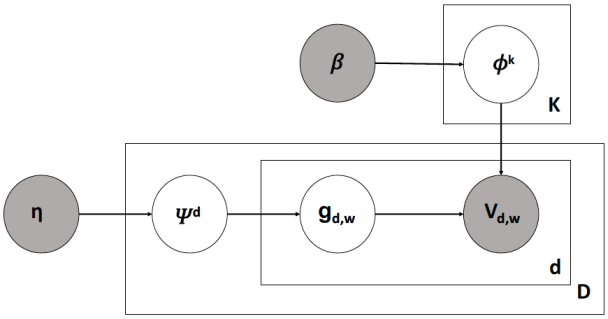


Figure 3.1: Plate notation of the Latent Dirichlet Allocation (LDA) topic model

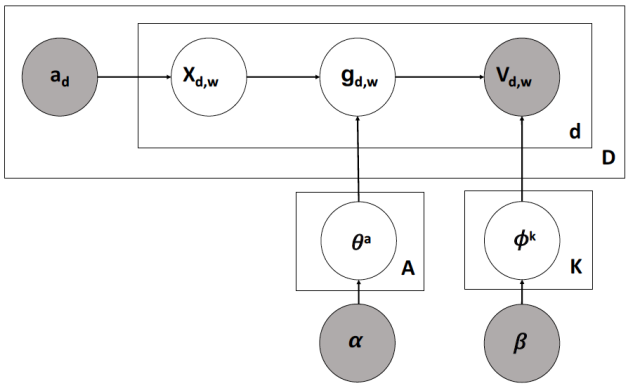


Figure 3.2: Plate notation of the Author-Topic (AT) model

The main idea of LDA is the following: given a collection of text documents, the algorithm computes the posterior distributions of the hidden variables to determine the topical structure of the collection. Since it is intractable to obtain the exact value of the posterior distribution, several approximation algorithms were proposed to handle this problem such as the *variational inference* [17] and *Gibbs sampling* [82]. Further details of LDA can be found in [17].

### 3.1.2 Author-Topic Model (AT)

The Author-Topic model was proposed by Rosen-Zvi et al. [82] to model an author's interests as well as the topics contained in the text documents collection. This model assumes that the set of authors of each text document is observed and models each author as a distribution over topics that reflect an author's interests and each topic as a distribution over the terms in the vocabulary. The Author-Topic model includes the authorship information to extend the existing generative probabilistic models. Formally, let $D$ be the corpus of text documents, $K$ be the set of topics in $D$, $V$ be the vocabulary of terms, and $A$ be the set of authors of $D$. Let $\phi^{k_i}$ be the word distribution for a topic $k_i$ and $\theta^{a_b}$ be the topic distribution of an author $a_b$. Let $a_{d_j}$ be the vector of authors of document $d_j$. Let $g$ and $x$ denote the topic and author assignments, respectively. Let $\beta$ and $\alpha$ denote the hyperparameters of Author Topic model. Then, the generative process for the Author Topic model is shown in Algorithm 2, and its plate notation is shown in Figure 3.2.

Similar to LDA, to learn the hidden structure of the text document collection, the value of the

**Algorithm 2** The generative process of Author Topic model

**Input** A corpus of text documents $D$ and the set of authors $A$
**Output** Distributions of authors over topics $\Theta$, distributions of topics over vocabulary $\Phi$, topics $G$ and author $X$ assignment to words.

**for** each author $a_b$ indexed by $b = 1$ to $|A|$ **do**
    Draw a distribution over topics $\theta^{a_b} \sim Dir\,(\alpha)$.
**end for**
**for** each topic $k_i$ indexed by $i = 1$ to $|K|$ **do**
    Draw a distribution over vocabulary of terms $\phi^{k_i} \sim Dir\,(\beta)$.
**end for**
**for** each document $d_j$ indexed by $j = 1$ to $|D|$ (Given the vector of authors $a_{d_j}$) **do**
    **for** each word $w_n$ indexed by $n = 1$ to $|d_j|$ **do**
        Conditioned on $a_{d_j}$, Draw an author $x_n \sim Uniform\left(a_{d_j}\right)$.
        Conditioned on $x_n$, Draw a topic $g_n \sim Discrete\,(\theta^{x_n})$.
        Conditioned on $g_n$, Draw a term $v_n \sim Discrete\,(\phi^{g_n})$.
    **end for**
**end for**

posterior distribution should be estimated using approximation algorithms. Further details of the Author Topic model can be found in [82].

## 3.2 Artificial Neural Networks (NN)

Artificial Neural Networks, or Neural Networks (NN), are a family of computing networks that was inspired by the biological neural networks in the human brain. It was first proposed by McCulloch and Pitts [66] in 1943 with the objective of building a computer program capable of imitating the human brain's ability to learn and make decisions. The basic building block in artificial neural networks is known as the perceptron. A perceptron in artificial neural networks corresponds to the neuron in biological neural networks. Thus, an artificial neural network consists of multiple perceptrons connected together to build a network structure. To understand how complex neural networks work, one should first understand how a single perceptron works. In the following subsections, we first give a high-level explanation of how a perceptron works followed by an example of a simple artificial neural network architecture. Next, we briefly introduce two important neural network architectures: the *Recurrent Neural Network (RNN)* and the *Convolutional Neural Network (CNN)*.

### 3.2.1 A Perceptron in Artificial Neural Networks

Perceptrons were first proposed by Rosenblatt [83] in 1962. A perceptron in artificial neural networks works as follows. It takes several binary inputs along with their weights, performs some calculations, and produces a single binary output. Formally, let $inputs = \langle inp_1, \ldots, inp_{|inputs|} \rangle$

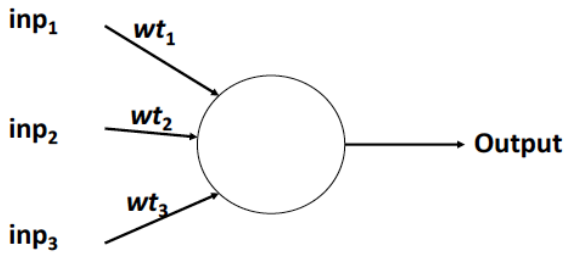


Figure 3.3: A perceptron with three inputs in artificial neural networks

be the binary inputs to a perceptron and $weights = \langle wt_1, \ldots, wt_{|inputs|} \rangle$ be the corresponding real-valued weights expressing the importance of each input to the output. Then, the output of a perceptron is calculated as follows:

$$Output = \begin{cases} 0 & if \quad inputs.weights + b \leq 0 \\ 1 & if \quad inputs.weights + b > 0 \end{cases} \tag{1}$$

where $b$ is the bias and $inputs.weights$ is the dot product of the $inputs$ and the $weights$ matrices. Figure 3.3 illustrates the basic architecture of a single neural network perceptron with three inputs.

### 3.2.2 A Simple Neural Network Architecture

Figure 3.4 illustrates a simple neural network architecture. In this neural network, the leftmost column of perceptrons is called the input layer. Consequently, all perceptrons within this layer are called the input neurons. Similarly, the rightmost layer of perceptrons is called the output layer, and perceptrons within this layer are called the output neurons. All layers between the input and the output layers are called the hidden layers, and all perceptrons within hidden layers are called hidden neurons. The multiple output arrows from each perceptron indicate that the single output of that perceptron is being used as input to several perceptrons in the subsequent layer. Such multiple layers neural networks are sometimes referred to in the literature as *Multi-Layer Perceptrons (MLPs)* [76].

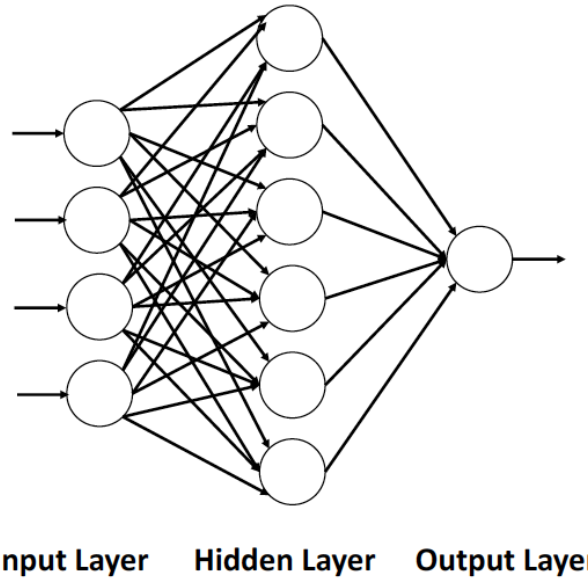


Figure 3.4: A simple neural network architecture

A well-known type of multi-layer neural networks is the *feed-forward neural networks* in which the output of a layer is passed to the next layer. In these networks, the information is always passed forward [76]. The following two subsections cover two important feed-forward neural network families.

### 3.2.3 Recurrent Neural Networks (RNN)

*Recurrent Neural Networks (RNNs)* represent a rich family of feed-forward neural networks used mainly to handle variable-length sequential or time series data. RNNs have been used for many tasks, including sequence generation and classification. A standard RNN works as follows [36]: given an input vector sequence $\mathfrak{S}$ of length $T$, denoted by $\mathfrak{S} = \langle \mathfrak{s}_1, \ldots, \mathfrak{s}_T \rangle$, for each time step $t = 1$ to $T$, the algorithm iterates over the following equations to update the hidden states of the network $\mathfrak{h} = \langle \mathfrak{h}_1, \ldots, \mathfrak{h}_T \rangle$ and generate the outputs $\mathfrak{o} = \langle \mathfrak{o}_1, \ldots, \mathfrak{o}_T \rangle$ [63]:

$$\mathfrak{h}_t = tanh\left(\mathcal{U}\mathfrak{s}_t + \mathcal{W}\mathfrak{h}_{t-1} + \mathfrak{b}\right) \tag{2}$$

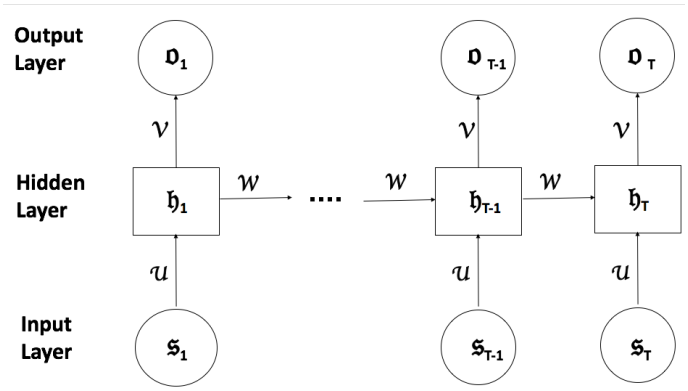


Figure 3.5: A simple recurrent neural network (RNN) architecture

$$\mathfrak{o}_t = \mathcal{V}\mathfrak{h}_t + \mathfrak{c} \tag{3}$$

where the terms $\mathcal{W}, \mathcal{U}$, and $\mathcal{V}$ denote weight matrices connecting hidden to hidden, input to hidden, and hidden to output layers, respectively, and the terms $\mathfrak{b}$ and $\mathfrak{c}$ denote bias vectors. The $tanh\,(.)$ denotes a hyperbolic tangent non-linear function. Figure 3.5 illustrates a simple RNN architecture.

A RNN is trained by unfolding it into a deep feedforward network. Meaning that, for every time stamp in the input sequence, a new hidden layer is created in the corresponding feedforward network architecture. However, due to the finite length unfolding used to train RNNs, especially the above vanilla RNN, they are incapable of learning long-distance temporal dependencies when traditional activation functions are used. In RNNs, the gradient of the current time stamp completely depends on the next time stamp during the back-propagation step and the gradient of the traditional activation functions, such as the hyper tangent functions, are in the range of $[0, 1]$. Since the back-propagation algorithm computes the gradients in RNNs by applying the chain rule, the gradients will either vanish or explode. One solution to address this problem is to use the extended RNNs architectures designed to store previously seen information in a better way such as Long Short-Term Memory (LSTM) [36, 41] and *Gated Recurrent Unit (GRU)*[22].

### 3.2.4 Convolutional Neural Network (CNN)

*Convolutional Neural Networks (CNNs)* [53] represent another family of feed-forward neural networks used mainly to handle images. In CNNs, the layers of the network have their neurons

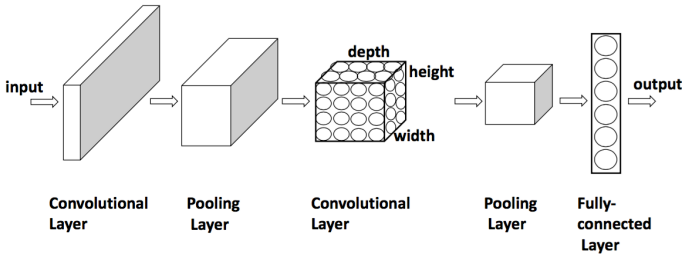


Figure 3.6: A simple convolutional neural network (CNN) architecture

arranged in 3-dimensions: width, height, and depth. Thus, each layer takes a 3-dimensional matrix as an input and generates another 3-dimensional matrix as its output except the output layer, which reduces its output to a single vector of probability scores. Figure 3.6 shows a simple CNN architecture. A typical CNN consists of three main types of layers: *convolutional layers, pooling layers*, and *fully-connected layers*.

Convolutional layers perform convolution operations by applying convolutional filters over the input 3-dimensional matrix to produce different feature maps. A convolutional filter (sometimes referred to as the kernel) is also a 3-dimensional matrix that has to be of the same depth as the input matrix. A single convolution operation works as follows. It slides a convolutional filter over the input matrix and performs an element-wise multiplication of the two matrices at every location. The amount by which the convolutional filter slides over the input matrix is known as the *stride*. The multiplication results are then summed up into a feature map. In CNNs, each convolutional layer will perform several convolution operations using different convolutional filters and, thus, produce many feature maps. These feature maps are then passed to the subsequent layer.

Pooling layers are usually inserted between successive convolutional layers in CNNs. They are mainly used for dimensionality reduction (downsampling) of the feature maps along the spatial dimensions: width and height. This reduces the learning computational cost of the model's parameters and, consequently, controls the overfitting issue. *Max-pooling* is one of the most used types of pooling in CNNs. A max-pooling layer slides a *max-filter* over the input matrix and takes only the maximum value at each location. These maximum values are used to create the output matrix of the

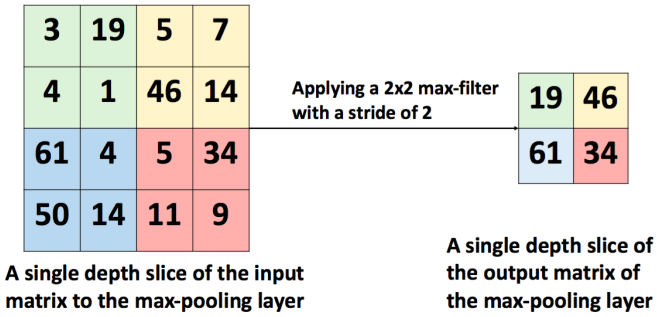


Figure 3.7: An example of applying a 2x2 max-filter with a stride of 2 on a 2-dimensional input matrix of size 4x4

pooling layer, where each element is the maximum value of a sub-region in the input matrix. This helps the model keep only the significant information and pass it to the subsequent layer. Figure 3.7 illustrates a simple example of applying a 2x2 max-filter with a stride of 2 on a 2-dimensional input matrix of size 4x4.

Fully-connected layers are only attached to the end of CNNs to produce the final outputs. In a fully-connected layer, each output of the previous layer is connected to all neurons in the fully-connected layer. The final output of a CNN model is a vector of probability scores, one for each output class.

## 3.3 Attention in Neural Networks

Attention mechanisms are a family of computational mechanisms that help neural networks attend differently to different inputs. Basically, an attention mechanism takes into consideration several inputs and calculates a degree of importance for each of these inputs to predict the output. This helps the learning model decide which inputs to focus on and to what extent. Attention mechanisms were originally proposed by Bahdanau et al. [7] to overcome the issues of *sequence-to-sequence* machine translators. To understand the basic concept of attention mechanisms in neural networks, one must first understand how sequence-to-sequence models work. In the following subsections, we provide an overview of sequence-to-sequence neural network machine translators followed by a description of the basic attention mechanism in neural networks.

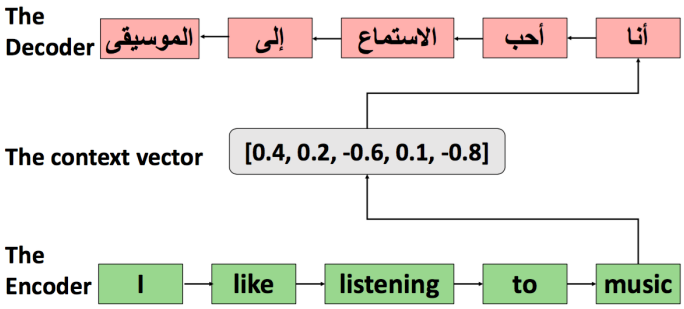


Figure 3.8: A simple example of a seq2seq model translating the English sentence "I like listening to music" to Arabic

### 3.3.1 Sequence-to-Sequence Language Translation with Neural Networks

Sequence-to-Sequence neural networks machine translators, known as *seq2seq* models, were first proposed by Sutskever et al. [91] with the objective of translating a sequence of words from one language to another. A typical seq2seq neural network model consists of two connected RNNs: *the encoder* and *the decoder*. It works as follows. First, the *encoder* takes a sequence of words of length $\mathcal{M}$ in a source language and encodes it into a single fixed-length context vector. Then, the *decoder* uses this context vector to generate the output sequence of words of length $\mathcal{N}$ in the target language. Figure 3.8 illustrates a simple example of a seq2seq model translating the English sentence "I like listening to music" to Arabic.

Seq2seq neural network machine translators suffer from two main issues. First, the difference in length between the input sequence and output sequence of words might cause alignment difficulties in vanilla RNNs. Furthermore, they suffer from gradient vanishing/exploding problems, especially when the sequence of words is long [42].

### 3.3.2 Basic Attention Mechanism

Bahdanau et al. [7] proposed to use an attention mechanism in the seq2seq neural network machine translator to solve the aforementioned issues. The proposed attention mechanism was incorporated into the seq2seq neural network model as an intermediate step between the encoder and the decoder, as shown in Figure 3.9.

Attention mechanisms work as follows. Instead of the single context vector generated by the encoder in the original seq2seq machine translator architecture, the proposed attention mechanism learns a context vector for each output target word [42]. Formally, let $Src = \langle w_1^{src}, \ldots, w_{\mathcal{M}}^{src} \rangle$ be the sequence of $\mathcal{M}$ input source words to the encoder, and let $H^{enc} = \langle h_1^{enc}, \ldots, h_{\mathcal{M}}^{enc} \rangle$ be the sequence of $\mathcal{M}$ hidden outputs generated by the encoder. Similarly, let $H^{dec} = \langle h_1^{dec}, \ldots, h_{\mathcal{N}}^{dec} \rangle$ be the sequence of $\mathcal{N}$ hidden states generated by the decoder, and $Trgt = \left\langle w_1^{trgt}, \ldots, w_{\mathcal{N}}^{trgt} \right\rangle$ be the sequence of $\mathcal{N}$ output target words generated by the decoder. Let $j$ be the current decoding step, then the context vector of $w_j^{trgt}$ is calculated as follows. First, the similarity between $w_j^{trgt}$ and each $w_i^{src}$ in $Src$ is calculated as follows:

$$e_{ji} = \mathfrak{a}\left(h_{j-1}^{dec}, h_i^{enc}\right) \tag{4}$$

where $\mathfrak{a}$ is the alignment function. Then, the attention score $att_{ji}$ of $w_j^{trgt}$ and each $w_i^{src}$ in $Src$ is calculated as follows:

$$att_{ji} = \frac{exp(e_{ji})}{\sum_i^{\mathcal{M}} e_{ji}} \tag{5}$$

Finally, the context vector $con_j$ of $w_j^{trgt}$ is calculated as follows:

$$con_j = \sum_i^{\mathcal{M}} att_{ji} h_i^{enc} \tag{6}$$

The context vector $con_j$ is then used by the decoder to generate the target output word $w_j^{trgt}$ as follows:

$$w_j^{trgt} = f(h_{j-1}^{dec}, w_{j-1}^{trgt}, con_j) \tag{7}$$

The great success of the Attention-based machine translation models has motivated many recent works to incorporate attention mechanisms into different machine learning and natural language processing tasks.

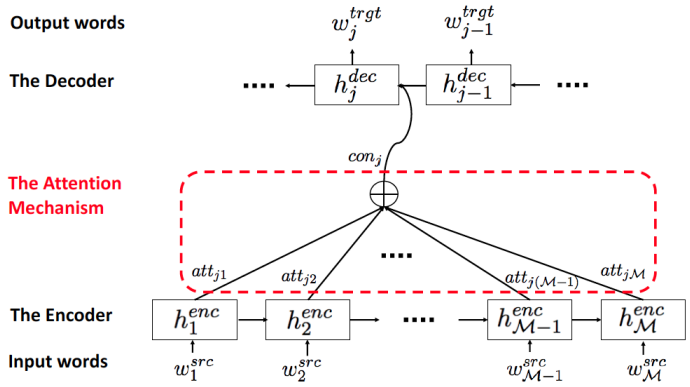


Figure 3.9: A seq2seq neural network model with attention

## 3.4 Representation Learning

Representation learning is the process of transforming raw input data to a representation that can be then used by machine learning algorithms for different tasks. The goal of representation learning algorithms is " disentangling the unknown underlying factors of variation that explain the observed data" [35]. Several advantages of representation learning have encouraged its wide adoption in the last few years. First, studies have shown that using learned representations as input data to machine learning algorithms often yields better results than the results obtained using hand-crafted representations [12]. Furthermore, for a simple task, representation learning algorithms can learn a good data representation in only a few minutes. Also, these learned representations of input data exploit hidden factors and allow for a more thorough analysis of the observed data.

Representation learning has been successfully used for different machine learning tasks including speech recognition and object recognition, as well as many natural language processing tasks. For example, Ding et al. [28] showed how representation learning can improve authorship analysis. Similarly, the work proposed in [27] showed how representation learning can help software binary analysis.

A widely adopted representation of textual data in many natural language processing tasks is the distributed vector representation of words, also known as *word embeddings*. In this representation, a word is represented using a real-valued low-dimensional vector [56, 59]. The simplest form of a word vector representation is the 1-hot vector. This representation uses a dictionary of words to

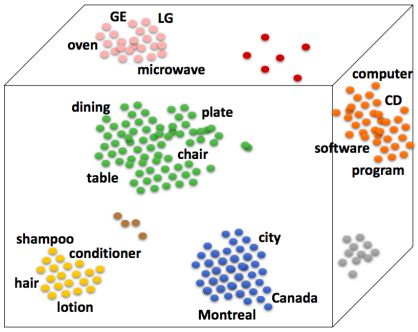


Figure 3.10: A visualization example of word embeddings in a 3-dimensional embedding space.

map a word to a vector representation as follows. A word is converted into a vector of the same size as the dictionary. This vector has the value *one* only at the position that corresponds to the position of the word in the dictionary and has *zeros* everywhere else. For example, given the dictionary of words: [‘Apple’, ‘Are’, ‘Computer’, ‘Hi’, ‘Sports’]. The 1-hot vector representation of the word ‘Computer’ is [0,0,1,0,0] and of the word ‘Sports’ is [0,0,0,0,1].

To learn more flexible vector representations of words, several techniques have been proposed where a word is represented as a D-dimensional vector of probability scores learned automatically from large unlabeled textual corpora [11, 23, 55, 68]. These techniques use shallow neural networks to learn the distributed vector representations of words so that semantically similar words appear close to each other in the *embedding space*. Figure 3.10 shows a visualization example of how word embeddings appear in a 3-dimensional embedding space.

# Chapter 4

# Improving Interpretations of Topic Modeling in Microblogs

The materials in this Chapter have been published in the Journal of the Association for Information Science and Technology (JASIST) [4] in 2018, with impact factor 2.875.

## 4.1 Introduction

Statistics from Twitter show that around 500 million tweets were tweeted per day in February 2016[1]. The huge volume of text in microblogs contains valuable real-time information from different regions of the world. Having an effective method to automatically extract knowledge from such a volume of textual data would provide tremendous advantages to trend and topic analysis in marketing. *Latent Dirichlet Allocation* (*LDA*) [17] is a widely adopted topic modeling method that can automatically generate a set of topics from a large collection of textual data. In this chapter, we study the shortcomings of LDA and the challenges of applying LDA to microblogs for topic analysis. Furthermore, we present a customized version of *Twitter-LDA* [104] that can better represent the generated topics for microblogs by incorporating a lexical database, domain-specific keywords, and hashtags into the generative model. The proposed method is specifically designed for the domains that satisfy the following four properties: (1) huge volume of textual data, (2) each

[1]Source: http://www.internetlivestats.com/twitter-statistics/, retrieved on January 24, 2017

individual piece of text is very short with overlapping vocabularies, (3) concepts and terminologies are domain-specific, and (4) terminologies change rapidly by the community over time. To illustrate the effectiveness of the proposed method, we present objective quantitative results, together with users' evaluations, on real-life fashion tweets. To further ensure the generated results are meaningful and useful to fashion practitioners, we closely collaborate with a domain expert in fashion communication. We choose fashion communication as the domain of case study because it satisfies the aforementioned properties. Our method can be generalized to other domains that share similar properties such as video games, photography, and social media applications.

### 4.1.1 The Challenges

The problem of handling large collections of text documents and the effectiveness in extracting useful information from the available data has drawn the attention of many researchers. The absence of semantic structures in such collections makes the process of browsing and accessing text documents with similar ideas, i.e., topics, very difficult. With such large collections, a simple search query may result in millions of text documents that overwhelm the user with textual data. Topic Models were proposed to solve this problem by automatically detecting the underlying semantic structure of large text document collections and providing short descriptions of text documents. Uncovering this structure facilitates browsing and exploring the collection and allows the user to effectively access documents with similar topics.

LDA [17] is one of the most well-known topic models in the literature and serves as the foundation of many other models. LDA assumes a fixed number of topics for the entire corpus. Each topic in LDA is defined as a distribution over a vocabulary of terms, and each document is modeled as a mixture distribution of underlying topics. The difficulty of applying LDA on short text documents to generate meaningful results raised the need of proposing new topic models to handle them. Twitter-LDA [104] is a topic model that was proposed to handle the micro-posts, known as tweets, available in Twitter. This topic model takes into consideration the observation that a single tweet has a single author and usually covers a single topic.

Dealing with short text documents like tweets is challenging. In addition to the lack of co-occurrence patterns and high sparseness of the short text documents, tweets are very noisy. They

Table 4.1: Two sets of keywords representing two different topics

| | |
|---|---|
| **A** | hair, fashion, photo, menstyle, kingjames, tbt, home, interiordesign, designer, women |
| **B** | fashion, style, mensfashion, wear, ootd, onlineshopping, stylish, fashionable, love, menstyle |

are often written using informal English with a lot of slang, domain-specific vocabularies, acronyms, and grammatical errors. They also contain URLs, emoticons, mentions, and hashtags. Even though Twitter has lifted the limitation that a tweet can contain only up to a maximum of 140 characters, cleaning the text of tweets leads to very few words in each tweet, which further complicates the process of extracting meaningful topics. The problem becomes even more challenging when the corpus actually covers one major topic, for example, fashion. In such case, we use topic models to detect subtopics. A major challenge here is that the same fashion-related terms are used across tweets covering different subtopics, leading to even fewer co-occurrence patterns of the distinctive terms that distinguish one subtopic from another. Thus, topics detected by topic models in this case are very similar, making it a difficult task to recognize which topic is represented by a given set of keywords.

**Example 1.** Table 4.1 shows two different sets of keywords representing two different topics detected by a topic model. Keywords shown in the table cannot be easily used to recognize what topic is represented by each set of keywords. ■

We employ *WordNet* to address this challenge and improve the set of keywords that represent each topic. WordNet is an English lexical database where words are grouped, based on their meanings, into unordered sets of synonyms. The most distinctive terms to a topic tend to be the most probable terms in its distribution over the vocabulary of terms. However, in the case of detecting subtopics of one general topic, some general terms might have higher probabilities than the most distinctive ones. Using the semantic relations in WordNet, we aim at emphasizing the importance of such distinctive terms. We also aim at taking advantage of the set of *hashtags* that exists in the corpus. Hashtags in Twitter are strong indicators of the topic covered by a tweet. Emphasizing the importance of terms similar to such strong indicators also improves the topic representation and makes it more focused rather than being about a general topic.

Table 4.2: Two sets of keywords representing the topic “Brands”

| | |
|---|---|
| **A** | prada, philiptreacy, hat, saint, laurent, collect, pradacelebs, campaign, spring, women |
| **B** | louisvuitton, assist, team, campaign, givenchy, service, love, tiffani, celebratingmonogram, collect |

Another known challenge in topic modeling is how to determine in advance the number of topics to be detected. Traditional topic models assume a fixed number of topics that should be specified by the user in advance. Providing a larger number of topics might result in different sets of keywords that represent the same topic. Having such results reduces the effectiveness of the topic model in terms of providing meaningful results to the user.

**Example 2.** Table 4.2 shows two sets of keywords detected by a topic model. By glancing through these sets of keywords, one can notice that all topics are mainly about “Brands”. Providing the user with two sets of keywords about the same topic makes the interpretability process more confusing. ■

To address this challenge, we propose to employ clustering algorithms to merge similar sets of keywords into a single topic and thus adjust the number of topics to be presented to the user. By doing this we provide the user with an estimation of the number of topics to be extracted from the text document collection, and at the same time the user still has the flexibility of choosing the number of topics that best serves his/her needs for obtaining different topic granularities.

Several works proposed to employ WordNet in topic models as a preprocessing step [60] or a post-processing step [74] to handle average-length text documents. Our model, on the other hand, focuses on very short text documents in Twitter and employs the semantic relations between terms in WordNet as an intermediate step in the inference scheme of the topic model.

### 4.1.2 Contributions

To the best of our knowledge, this is one of the first works that combines WordNet, hashtags, and topic models with the goal of improving the sets of keywords used to represent each topic extracted from short texts in Twitter. The main contributions of this work are:

- *Improved topic representation.* We used an English lexical database, WordNet, along with the

set of hashtags that exists in the corpus, to improve the set of keywords representing every topic by emphasizing the importance of distinctive terms in the distribution of every topic over the vocabulary of terms. Experimental results suggest that our method provides the user with better sets of keywords to represent each topic than does Twitter-LDA, and it improves the user's interpretation of the detected topics.

- *Customized taxonomy for a specific domain.* Our proposed approach can be used to dynamically build a customized taxonomy for a specific domain. To illustrate this capability, we chose *fashion* as the domain in this study. Specifically, we use the maximal frequent itemsets extracted from the corpus to dynamically build a customized version of WordNet that contains fashion-related terms. The customized Wordnet can be used in different text mining tasks.

- *Adjustment of the number of detected topics.* We propose a mechanism to automatically adjust the number of topics to be presented to the user by merging topics represented by similar sets of keywords into a single topic. Experimental results suggest that the coherence of the merged topics is better than the coherence of the original topics.

- *Exploring changes of fashion topics over time.* To illustrate the capability of the proposed method, we evaluate the method by exploring the fashion topics and showing how they were covered over time and how specific users covered these topics by their tweets. Finally, we evaluate our method on two datasets collected from Twitter. The results obtained from the experiments show that our method is better than the original Twitter-LDA in terms of perplexity and topic coherence and provides better results in terms of the quality of the detected topics.

The rest of the chapter is organized as follows. In section 4.2, we provide the problem description. In section 4.3, we provide some important background knowledge. In section 4.4, we describe our proposed method. Finally, in section 4.5, we cover the experiments and results.

## 4.2 Problem Description

Given a corpus of tweets, our goal is to improve the most representative keywords of the set of topics covering the corpus by utilizing WordNet and the set of hashtags found in the corpus. We also

aim at merging topics represented by similar sets of keywords and building a customized version of WordNet.

**Definition 1** (Vocabulary)**.** A vocabulary is a set of distinct terms that are used to construct the text documents denoted by $V = \{v_1, \ldots, v_{|V|}\}$. A term $v_i$ is an item from a vocabulary V. ■

**Definition 2** (Tweet)**.** A tweet is a textual message that consists of a set of words denoted by $d = \{w_1, \ldots, w_{|d|}\}$. ■

**Definition 3** (Corpus)**.** A corpus is a collection of tweets $D = \{d_1, \ldots, d_{|D|}\}$ that is written using $V$. ■

**Definition 4** (Hashtag)**.** A hashtag is a textual word or phrase that has the symbol # as a prefix. Let $H$ be the set of hashtags in the corpus $D$. A tweet $d_j$ can link to a set of hashtags $H_{d_j} \subseteq H$. ■

**Definition 5** (Author)**.** Let $A = \{a_1, \ldots, a_{|A|}\}$ be the set of authors of $D$. An author $a_b$ contributed to one or more documents in $D$. ■

**Definition 6** (Topics in tweets)**.** Let $K = \{k_1, \ldots, k_{|K|}\}$ be the set of topics covered by $D$. A topic $k_i$ is modeled as a probability distribution $\phi^{k_i}$ over $V$. Each tweet $d_j$ has a single topic $k_{d_j}$. ■

**Definition 7** (Topic representation)**.** A topic $k_i$ is represented using the top $s$ probable terms $S^{k_i}$ in its probability distribution $\phi^{k_i}$ over $V$. ■

*WordNet* is an English lexical database where words are grouped based on their meanings into unordered sets of synonyms called *synsets*. Each group of synsets denotes a distinct concept and is linked to other groups of synsets via conceptual relations such as *hypernyms*, *hyponyms*, and *entailment*.

Formally, given a corpus of tweets $D$, we want to model the set of topics $K$ covered by $D$ each as a probability distribution $\phi^{k_i}$ over $V$ and the interests of each author $a_b \in A$ each as a probability distribution $\theta^{a_b}$ over $K$. We also want to tag every tweet $d_j$ with one of the topics in $K$, enhance the top $s$ probable terms $S^{k_i}$ representing each topic $k_i$ in $K$, build a customized version of WordNet to contain fashion-related terms, and merge similar sets of keywords to adjust the number of topics detected from $D$.

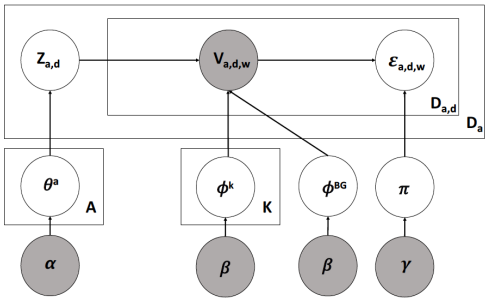


Figure 4.1: Plate notation of the Twitter-LDA topic model

## 4.3 Background Information

In this section, we provide a brief description of *Twitter-LDA* and posterior inference using *Gibbs sampling*.

### 4.3.1 Twitter-LDA

Twitter-LDA [104] is a modified Author-Topic model that takes into consideration the observation that in Twitter a single tweet has a single author and covers a single topic. Let $K$ be the set of topics in the collection of tweets, $\phi^{k_i}$ be the word distribution for a topic $k_i$, and $\phi^{BG}$ denote the words distribution for background words. Let $\theta^{a_b}$ be the topic distribution of the author $a_b$ and $\pi$ be the Bernoulli distribution, which determines the choice between topical or background words. Let $D_{a_b} = \{d_1, \ldots, d_{|D_{a_b}|}\}$ be the set of tweets written by author $a_b$. The generative process for Twitter-LDA is shown in Algorithm 3, and its plate notation is shown in Figure 4.1.

Further details about Twitter-LDA can be found in [104].

### 4.3.2 Inference Using Gibbs Sampling

The basic idea of topic modeling is to posit a hidden latent topical structure on the observed data and then use the posterior probabilistic inference to learn this structure. Since it is difficult to obtain the exact value of the posterior distribution, approximation algorithms such as *Gibbs sampling* [82] are used.

**Algorithm 3** The generative process of Twitter-LDA

**Input** A corpus of tweets $D$ and the set of authors $A$
**Output** Distributions of authors over topics $\Theta$, distributions of topics over vocabulary $\Phi$, and topics of tweets $Z$.

Draw a distribution over the vocabulary of terms $\phi^{BG} \sim Dir\,(\beta)\,, \pi \sim Dir\,(\gamma)$.
**for** each topic $k_i$ indexed by $i = 1$ to $|K|$ **do**
    Draw a distribution over the vocabulary of terms $\phi^{k_i} \sim Dir\,(\beta)$.
**end for**
**for** each author $a_b$ indexed by $b = 1$ to $|A|$ **do**
    Draw a distribution over topics $\theta^{a_b} \sim Dir\,(\alpha)$.
    **for** each tweet $d_j$ indexed by $j = 1$ to $|D_{a_b}|$ **do**
        Draw $z_{a_b,d_j} \sim Mult\,(\theta^{a_b})$.
        **for** each word $w_n$ indexed by $n = 1$ to $|d_j|$ **do**
            Draw $epsilon_{a_b,d_j,w_n} \sim Mult\,(\pi)$.
            **if** $\epsilon_{a_b,d_j,w_n} = 0$ **then**
                Draw $v_{a_b,d_j,w_n} \sim Mult\left(\phi^{BG}\right)$
            **else**
                Draw $v_{a_b,d_j,w_n} \sim Mult\left(\phi^{z_{a_b,d_j}}\right)$
            **end if**
        **end for**
    **end for**
**end for**

Gibbs sampling [31] is a Form of *Markov Chain Monte Carlo* that is widely used by topic models as an approximation algorithm to estimate the value of the posterior distribution on random variables. A two-step inference scheme [82] is employed. The process starts by running a Gibbs sampler to estimate the value of $P(z, x|D, \alpha, \beta)$, where $z$ and $x$ represent the author and topic assignment of words in the corpus $D$, respectively. And $\alpha$ and $\beta$ are the hyperparameters of the topic model. Next, the values of the posterior distribution on the random variables $\Theta$ and $\Phi$ are calculated using the following formulas:

$$\phi^{v_j k_i} = \frac{W(v_j, k_i) + \beta}{\sum_{v'_j} W(v'_j, k_i) + V\beta} \tag{8}$$

$$\theta^{k_i a_b} = \frac{\mathfrak{T}(k_i, a_b) + \alpha}{\sum_{k'_i} \mathfrak{T}(k'_i, a_b) + K\alpha} \tag{9}$$

where $\Phi$ and $\Theta$ represent the probability distribution of topics over the vocabulary and the probability distribution of the author over topics, respectively. $W$ is the count matrix that holds the counts for every term-topic pair, and $W(v_j, k_i)$ represents how many times the term $v_j$ was used in topic $k_i$. Similarly, $\mathfrak{T}$ is the count matrix that holds the counts for every topic-author pair, and $\mathfrak{T}(k_i, a_b)$ represents how many terms author $a_b$ used to write about topic $k_i$.

In our model, we added an intermediate step where we update the term-topic count matrix $W$

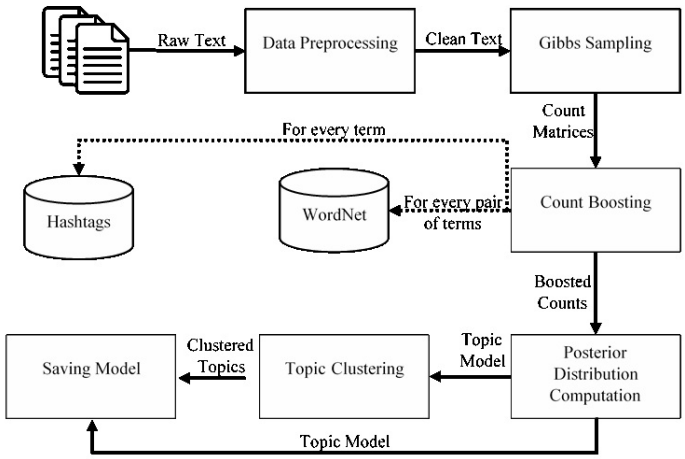


Figure 4.2: An overview of the proposed model for improving interpretations of topic modeling in microblogs

based on the similarity between terms using WordNet [69], an English lexical database, and the set of hashtags $H$ in the corpus. Then the updated term-topic count matrix is used by the final step to compute the posterior distribution on $\Phi$. This will be described in details later in this chapter.

## 4.4 Methodology

Figure 4.2 depicts an overview of the core modules of the proposed method. The first module applies some standard text preprocessing steps. The following three modules represent the inference process. First, the process starts by running a *Gibbs sampler*. Second, WordNet and the set of hashtags are used to adjust the importance of different terms to different topics. Third, the posterior distribution on the random variables is calculated. Finally, the topic clustering module groups similar topics together in order to provide a coherent representation of the topics to the user.

### 4.4.1 Gibbs Sampling

This module represents the first step in the inference scheme. A Gibbs sampler [31, 82] is used to estimate the topics and authors assignments, $z$ and $x$, and record their counts in two count matrices: $W$ and $\mathfrak{T}$. The first one contains the counts of every term-topic pair, while the other contains the

counts of every topic-author pair. The algorithm of Gibbs sampling has two steps. First, it randomly initializes the topic assignments $z$ and the author assignments $x$ for each word $w_i$. Second, during each Gibbs sampling iteration, it samples the author assignment $x_i$ and topic assignment $z_i$ for each individual word $w_i$ conditioned on fixed authors and topics assignments for all other words in the corpus. After completing a predefined number of iterations, the assignments $x$ and $z$ and the counts $W$ and $\mathfrak{T}$ are recorded to be used in the calculation of the posterior distribution on the probability distribution of topics over the vocabulary $\Phi$ and the probability distribution of the author over topics $\Theta$. We will focus on the topics distributions over vocabulary $\Phi$ in the rest of the chapter.

### 4.4.2 Count Boosting

This is the second step in the inference scheme. It takes the term-topic matrix $W$ and uses WordNet and the set of hashtags $H$ to update the counts of different terms according to their importance to different topics. Our intuition is that among the most probable terms for a topic, those that are semantically similar are the most distinctive ones to that topic. Therefore, we boost their counts based on their importance to the topic and their semantic similarities. Basically, we take the top $l$ probable terms $L^{k_i}$ for every topic $k_i$. Then for every pair of terms in $L^{k_i}$, we boost their counts based on their similarity in WordNet. More specifically, we use WordNet to retrieve the shortest path $dist$ between the two terms. For example, let $k_i$ be the current topic of interest. Then for every pair of terms $(v_x, v_j) \in L^{k_i}$, we use WordNet to retrieve the distance between them $dist(v_x, v_j)$ based on their lexical category. We only consider nouns and verbs in our work. Then the counts of both terms are boosted as follows:

$$WN(v_x, k_i) = W(v_x, k_i) + \frac{W(v_j, k_i)}{dist(v_x, v_j)} \tag{10}$$

$$WN(v_j, k_i) = W(v_j, k_i) + \frac{W(v_x, k_i)}{dist(v_x, v_j)} \tag{11}$$

where $WN$ represents the updated term-topic count matrix based on the relationships between terms in WordNet.

We further boost the counts of terms in $L^{k_i}$ for every topic $k_i$ by taking advantage of the set

of hashtags $H$ in $D$. Since hashtags are strong indicators of topics, our intuition is that among the most probable terms of a topic, those that appear in the topics hashtags are the most representative ones of that topic. Therefore, we boost their counts based on how often these hashtags are used to tag that topic. Let $k_i$ be the current topic of interest. Then for every term $v_j \in L^{k_i}$, we check if it appears in at least one of the hashtags $H_{k_i}$ associated with $k_i$. Let $H_{v_j}$ be the set of hashtags that contain the term $v_j$. The count of $v_j$ is boosted as follows:

$$WH(v_j, k_i) = WN(v_j, k_i) + \left[ WN(v_j, k_i) * \left( \frac{\sum_{h \in H_{v_j}} hashFreq[k_i][h]}{TotalHashFreq[k_i]} * 100 \right) \right] \quad (12)$$

where $WH$ represents the updated term-topic count matrix based on the set of hashtags, $hashFreq[k_i][h]$ denotes the frequency of hashtag $h$ in tweets about topic $k_i$, and $TotalHashFreq[k_i]$ denotes the sum of the frequencies of all hashtags in $H_{k_i}$.

Furthermore, this module builds a customized version of WordNet to include domain-related terms. Adding a new term to customize WordNet for a specific domain should comply with the following criteria: for a term $v_j$ to be added and connected to a set of terms $V_j$ in WordNet, the term $v_j$ should be related to all terms in $V_j$ in the context of that domain. We use the *maximal frequent itemsets MFI* found in the corpus as a guide to determine where to add these terms and how to connect them to other terms in WordNet.

**Definition 8** (Maximal Frequent Itemset (MFI) [19])**.** *Let the vocabulary $V$ be the set of all distinct terms. Let $I \subseteq V$ be an itemset. Let the collection of tweets $D$ be a multiset of subsets of the vocabulary $V$. The support of an itemset, $support(I)$, is the percentage of tweets in $D$ containing $I$. An itemset $I$ is a* frequent itemset *if $support(I) \geq minSup$, where $minSup$ is a user-defined minimum support. A frequent itemset $I$ is a* maximal frequent itemset *(*MFI*) if there is no superset of $I$ that is frequent.*

For the customization purposes, we assume that all terms to be added to WordNet are nouns, and all relationships are of type SIMILAR-TO. We then use MAFIA [19] to mine the MFI from the corpus $D$. Next, for every term $v_j$ in the top probable terms for topic $k_i$, $v_j \in L^{k_i}$, if it does not exist in WordNet, we find the maximal frequent itemset $MFI^{v_j}$ that contains $v_j$. If more than one is found, we use the one with the maximum support. Then, if at least one term in $MFI^{v_j}$ exists in

WordNet, we add $v_j$ to WordNet. Next, for every item (term) $v_x \in MFI^{v_j}$, we check if it exists in WordNet. If this is the case, we customize WordNet by adding a SIMILAR-TO relationship between $v_x$ and $v_j$.

### 4.4.3 Posterior Distribution Calculation

This is the final step in the inference where we actually compute the posterior distribution on the random variables. After boosting the counts, the computation of the posterior distribution on the random variables $\Phi$ and $\Theta$ is a straightforward step. Given the updated count matrix $WH$, the distributions of topics over the vocabulary of terms $\phi_{WH}^{v_j k_i}$ is calculated directly from Equation 8 as follows:

$$\phi_{WH}^{v_j k_i} = \frac{WH(v_j, k_i) + \beta}{\sum_{v'_j} WH(v'_j, k_i) + V\beta} \tag{13}$$

Similarly, the author's distributions over topics $\Theta$ is calculated from Equation 9 directly.

### 4.4.4 Improved Topic Clustering

Most parametric topic models assume a fixed number of topics, which is unknown in advance in most cases. We propose a new method that uses the *agglomerative hierarchical clustering* algorithm [43] to adjust the number of topics to be presented to the user based on the *Kullback-Leibler (KL)* divergence. The KL divergence is used to calculate the similarity between the distributions over vocabulary for every pair of topics. Let $D_{KL}(\rho||\tau)$ be the KL distance between the distributions of two topics $\rho$ and $\tau$. Since KL divergence is not symmetric, we calculate the distance $Dst(\rho||\tau)$ as follows:

$$Dst(\rho||\tau) = \frac{(D_{KL}(\rho||\tau) + D_{KL}(\tau||\rho))}{2} \tag{14}$$

At every step, the clustering algorithm calculates the KL divergence between every pair of topics and merges the pair with the lowest KL divergence value. To determine the best number of topics to be returned to the user we employed the *L method* [85]. The L method builds a two-dimensional evaluation graph where the x-axis represents the number of topics and the y-axis represents the KL-divergence. It then calculates and returns the "knee" of the evaluation graph, which represents the best number of topics that represent the corpus.

Table 4.3: Datasets statistics

| Dataset | Tweets | Authors | Vocabulary |
|---|---|---|---|
| **OffAcc dataset** | 83,404 | 51 | 29,155 |
| **FashionKW dataset** | 38,038 | 943 | 35,016 |

## 4.5 Experiments

We evaluated our method in terms of Perplexity, Topics' coherence, and their quality. We also analyzed the topics trends and interests of the users over time and show examples of customizing WordNet. In the experiments, we set the number of Gibbs sampling iterations of each topic model at 500 and fixed the hyperparameters at $\alpha = 50/K$ and $\beta = 0.01$.

### 4.5.1 Datasets

For evaluation purposes we collected two fashion datasets using Twitter API, namely, **OffAcc** and **FashionKW**. **OffAcc** has 100,099 tweets collected over 20 months, from September 2013 to May 2015. The tweets were retrieved from the official accounts of 51 fashion designers and magazines. **FashionKW** has 122,579 tweets collected over a period of 13 days from March 4, 2015 to March 16, 2015. The tweets were collected by sending search queries that contained 110 fashion-related hashtags (keywords) to Twitter API. The resulting corpus was written by 48,643 different users[2].

### 4.5.2 Data Preprocessing

We cleaned the tweets by removing URLs, emoticons, punctuation marks, mentions, stop words, and words that appear in more than 70% of the tweets. The remaining words were then stemmed using a *Porter Stemmer* [78]. The corpus was further processed so that duplicates and tweets with fewer than 3 words were removed. Furthermore, users with fewer than 10 tweets were removed, along with their tweets. Table 4.3 shows the statistics of the two datasets after the preprocessing.

[2] For repeatability, the tweet IDs are available on http://dmas.lab.mcgill.ca/fung/research/data/AFRH16tweetIDs.txt

### 4.5.3 Perplexity

To compare the predictive performance of our method with Twitter-LDA, we performed a 10-fold cross-validation and calculated the perplexity on hold-out testing data for $K = 3, 6, 9, 12, 15$. The perplexity is a widely used measurement to evaluate the ability of a probabilistic topic model to handle unseen documents. Lower perplexity implies better predictive performance of the model. It is defined as a decreasing function of the log-likelihood $\ell(D_{unseen})$ of unseen documents $D_{unseen}$, as follows:

$$Perplexity(D_{unseen}) = exp\left[\frac{-\ell(D_{unseen})}{M}\right] \tag{15}$$

where $M$ denotes the total number of words in the corpus.

We compared the results obtained from applying *Twitter-LDA* and our proposed method *Twitter-LDA with WordNet and Hashtags* (*Twitter-LDA-WNH*) on the OffAcc dataset. To evaluate the effect of including the set of hashtags, we also included our method with WordNet only (*Twitter-LDA-WN*) in the comparison. Figure 4.3 shows that Twitter-LDA-WNH has the lowest perplexity for all values of K, followed by Twitter-LDA-WN, while Twitter-LDA has the higher perplexity values. The results also show that the perplexity values for all three models increased when $k = 15$. Because we have a low number of topics in fashion, usually between 5 to 12, providing the model with a larger $K$ typically results in a complicated model with many vague topics that are difficult to interpret, and it reduces the model's performance when handling new documents. Similarly, when $K = 3$, we got a complicated model with very general topics, which in turn reduces the ability of the model to handle new documents.

The obtained results suggest that our method, Twitter-LDA-WNH, has the best predictive performance compared to Twitter-LDA and Twitter-LDA-WN for all values of $K$. The results also reveal that the inclusion of the set of hashtags in our method can further improve the results in terms of handling unseen documents.

### 4.5.4 Topics Coherence

We further evaluated the quality of our results in terms of topics coherence. Our goal is to show how incorporating WordNet and the set of hashtags in our method helps increase the coherence of

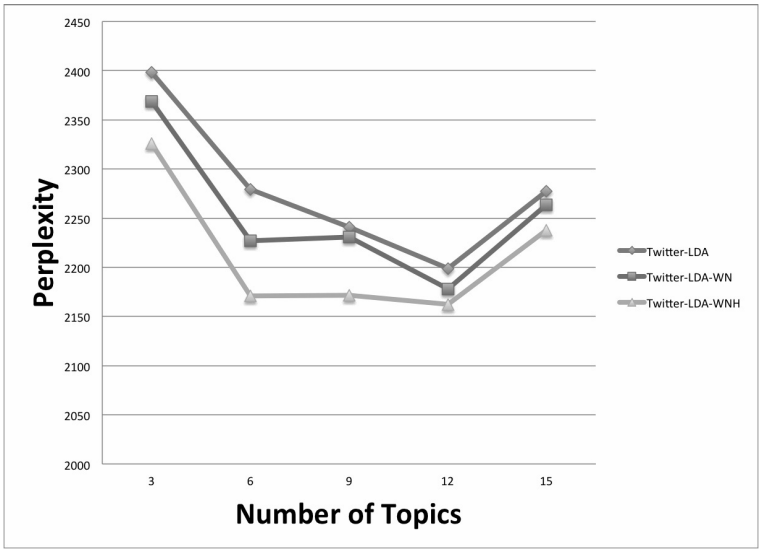


Figure 4.3: Perplexity on OffAcc dataset

the learned topics over topics learned by Twitter-LDA. The employment of the semantic relations in WordNet and the hashtags as an intermediate step helps emphasize the importance of distinctive terms during the inference process itself. Consequently, this helps minimize the effect of the overlapping vocabulary in a specific domain and distinguish the learned topics from each other.

We used the *Normalized Pointwise Mutual Information (NPMI)* [3] and the *CP* [81] in our experiment to measure the coherence of the topics learned by *Twitter-LDA*, our method, *Twitter-LDA-WNH*, and the results of our method after clustering the topics, *Clustered-TLDA-WNH*. In their study, Röder et al. [81] evaluated several coherence measures in terms of their correlation to human ratings. Their study shows that NPMI has the strongest correlation to human ratings among all already existing coherence measures while CP outperforms all coherence measures that use direct confirmation, including NPMI. This justifies the reason for using CP and NPMI for our experiment.

CP uses a one-preceding segmentation of the top keywords to calculate the coherence of a topic. For every keyword, the confirmation to its preceding keyword is calculated using Fitelson's coherence [30] as follows:

$$\varrho\left(w_i, w_j\right) = \left(\frac{p\left(w_j|w_i\right) - p\left(w_j|\neg w_i\right)}{p\left(w_j|w_i\right) + p\left(w_j|\neg w_i\right)} + \frac{p\left(w_i|w_j\right) - p\left(w_i|\neg w_j\right)}{p\left(w_i|w_j\right) + p\left(w_i|\neg w_j\right)}\right)/2 \tag{16}$$

The arithmetic mean of the Fitelson's coherence results is the CP value of that topic.

NPMI uses a one-one segmentation of the top keywords to calculate the coherence of a topic. For every pair of keywords, the confirmation is calculated as follows:

$$NPMI\left(w_i, w_j\right) = \frac{PMI\left(w_i, w_j\right)}{-log\left(p\left(w_i, w_j\right)\right)} \tag{17}$$

The arithmetic mean of the NMPI results is the overall NPMI value of that topic.

In this experiment we represented each topic as a set of the top 10 most probable keywords in its distribution over terms and used *Palmetto*[3] to calculate the NPMI and CP values for each topic. A two-samples t-test was conducted to compare the NPMI and CP coherence values of topics learned by Twitter-LDA and Twitter-LDA-WNH. The obtained results show that there is a significant difference in the NPMI coherence for topics learned by Twitter-LDA ($M = -.03, SD = .04$) and by Twitter-LDA-WNH ($M = .02, SD = .06$); $t(24) = 2.45, p = .01$. The results also show that there is a significant difference in the CP coherence for topics learned by Twitter-LDA ($M = -.09, SD = .20$) and by Twitter-LDA-WNH ($M = .26, SD = .21$); $t(28) = 4.76, p < .001$. These results suggest that topics in our model are more coherent. Specifically, our results suggest that when we incorporate WordNet and hashtags into the generative model of Twitter-LDA, the coherence of the topics increases.

To evaluate the results of topics' clustering, we also compared the coherence of topics learned by Twitter-LDA and our model after clustering, Clustered-TLDA-WNH. We started with an initial value of $K = 15$ and $K = 8$ for the OffAcc and FashionKW datasets, respectively. For the OffAcc dataset, the clustering algorithm performed 6 merges and reduced the number of topics to 9. Similarly, for the FashionKW dataset, the clustering algorithm performed 2 merges and reduced the number of topics to 6.

We conducted a two-samples t-test to compare the NPMI and CP coherence values of topics learned by Twitter-LDA and Clustered-TLDA-WNH. The obtained results show that there is a significant difference in the NPMI coherence for topics learned by Twitter-LDA ($M = -.03, SD = .04$) and by Clustered-TLDA-WNH ($M = .08, SD = .09$); $t(13) = 3.90, p < .001$. Similarly, the

[3]Source: https://github.com/AKSW/Palmetto, retrieved on January 26, 2019

results show that there is a significant difference in the CP coherence for topics learned by Twitter-LDA ($M = -.09, SD = .20$) and by Clustered-TLDA-WNH ($M = .27, SD = .08); t(19) = 6.61, p < .001$. These results suggest that merging topics learned by our model yields more coherent topics than the original topics learned by Twitter-LDA. The results also suggest that the best number of topics for the OffAcc and FashionKW datasets are 9 and 6, respectively.

### 4.5.5 Users' Evaluation

The objective of the users' evaluation is to compare the quality of the results obtained through the application of our method, Twitter-LDA-WNH, and Twitter-LDA in terms of both interpretation and representation of the topic from the perspective of human users. Due to the fact that judging the quality of a topic is subjective, and to avoid our bias interpretation, we conducted an online survey in March 2017 and asked participants to judge the quality of the top probable keywords generated by the two methods.

#### 4.5.5.1 Evaluating the Interpretation of Topics

The perplexity results suggest that the best number of topics for the OffAcc dataset is in the range of $6 - 12$ topics. The topics coherence results also suggest that the best number of topics for the OffAcc dataset is $9$ topics and $6$ topics for the FashionKW dataset. Therefore, in this experiment we set $K = 9$ and $K = 6$ for the OffAcc and FashionKW datasets, respectively, and applied Twitter-LDA and our proposed method Twitter-LDA-WNH. This resulted in a total of 30 topics, each method yielding 15 topics. We also carefully prepared a set of 16 labels to cover the most popular topics in the fashion industry as follows: First, we systematically went through the top 10 popular fashion magazines and identified the common topics. Then, we reviewed these topics with a fashion expert and merged them into 16 topics with minimal overlapping. We then took the top $10$ probable keywords in the distribution of every topic and prepared the test so we had $30$ sets of keywords generated by the two methods and a set of 16 labels representing the topics. The sets of keywords were mixed together in random order so that the participants did not know which set was generated by which method. We then asked $105$ participants to assign a label to each set of keywords. Our participants were undergraduate students from Ryerson University in Canada with

academic backgrounds in fashion.

To evaluate the results, we prepared a standard answer that represents the true topics' labels for each set of keywords based on the judgment of a fashion expert. We would like to emphasize that we did not ask the fashion expert to evaluate the performance of our method. To avoid any bias, we provided the expert with 30 sets of keywords mixed in random order and asked him to assign a label to each set of keywords. The expert did not know the method that generated each set. The expert's interpretations for topics were only used as the gold standard for comparing responses gathered from other participants to the true answer and recording the percentage of the correct answers for each set of keywords. For evaluation purposes, even if a participant selected a label that was different from the golden answer, it does not necessarily mean the answer is wrong in practice. However, we consider the chosen label as incorrect. We acknowledge that this evaluation process is harsh. Given such a harsh evaluation setting, we can still show that our proposed method yields good results.

Table 4.4 shows that the interpretation of the true topic label of the sets of keywords improved for the OffAcc and FashionKW datasets after applying Twitter-LDA-WNH by an average of $14\%$ and $22\%$, respectively. A two-samples t-test was conducted to compare the average of users' interpretation of topics learned by Twitter-LDA and Twitter-LDA-WNH. The obtained results show that there is a significant difference in the users' interpretation for topics learned by Twitter-LDA $(M = .38, SD = .05)$ and by Twitter-LDA-WNH $(M = .56, SD = .01); t(2) = 4.94, p = .02$. These results suggest that topics in our model become more interpretable by users.

We further analyzed the results and noticed that topics represented by a lot of acronyms, fashion brands, and names did not improve by applying our method. Table 4.5 shows some examples of such topics after applying Twitter-LDA-WNH on the OffAcc dataset. As shown in the table, the set of keywords representing topic 7 did not improve after applying our method, while topic 1, on the other hand, fell under the topic of *Media* instead of *Celebrities*. Another finding was that if some fashion terms do not exist in WordNet, the ability of our method to improve some topics is diminished. Furthermore, since only $31\%$ of tweets in the OffAcc dataset contain hashtags, the influence of emphasizing the importance of terms based on the set of hashtags was limited. As a result, the interpretation of some of these topics did not improve, while the interpretation of others changed completely.

Table 4.4: Average percentage of the correct answers for both models

| Dataset | Twitter-LDA | Twitter-LDA-WNH |
|---|---|---|
| **OffAcc** | 41% | 55% |
| **FashionKW** | 34% | 56% |

Table 4.5: Examples of topics in OffAcc and FashionKW datasets

| | Twitter-LDA | | Twitter-LDA-WHN | |
|---|---|---|---|---|
| **Topic #** | **Keywords** | **Label** | **Keywords** | **Label** |
| **7** | dolce, gabbana, amp, dgwomen, versace, dolcegabbana, dgeditorials, wear, discover, fashion | Brands | dolce, gabbana, amp, versace, wear, fashion, dgwomen, dolcegabbana, summer, dgeditorials | Brands |
| **1** | kim, kardashian, tylor, video, beyonce, swift, west, jenner, watch, kany | Celebrities | fashion, time, song, love, thing, kendal, dress, video, west, watch | Media |

#### 4.5.5.2 Evaluating the Quality of Topics' Representations

We further evaluated the quality of the keywords used to represent each topic in terms of the number of representative keywords of each topic after applying Twitter-LDA and our method, Twitter-LDA-WNH.

Table 4.6 and Table 4.7 show some examples of different sets of keywords resulting from applying the two methods and how they were interpreted by users. Table 4.6 shows how both sets of keywords were interpreted to be about the topic *Jewelry*. The set of keywords resulting from applying Twitter-LDA-WNH was better than the one resulting from applying Twitter-LDA in terms of the number of related keywords to that topic. Table 4.7 shows how the interpretation of the set of keywords has changed. The label *Celebrities* was assigned for the set of keywords resulting from applying Twitter-LDA. This assignment has shifted to the topic *Events* after applying our method. The obtained results show that the average number of improved representative keywords are 5.3 keywords for our method, in contrast to 3.3 for Twitter-LDA. The results also suggest that although the interpretation of some topics has been completely changed, our method provides meaningful topics with a reasonable number of representative keywords.

Table 4.6: Improvements of the sets of keywords resulting from Twitter-LDA-WNH over Twitter-LDA

| Model | Keywords | Label | # Related words |
|---|---|---|---|
| **Twitter-LDA** | jewelry, fashion, menstyle, hair, photo, ring, kingsjames, tbt, home, interiordesign | Jewelry | 3 |
| **Twitter-LDA-WNH** | ring, jewelry, fashion, diamond, silver, gold, photo, hair, home, vintage | Jewelry | 7 |

Table 4.7: Different interpretation of two sets of keywords resulting from Twitter-LDA and Twitter-LDA-WNH

| Model | Keywords | Label | # Related words |
|---|---|---|---|
| **Twitter-LDA** | red, oscars, carpet, dress, kate, gown, kardashian, kim, wed, celebr | Celebrities | 9 |
| **Twitter-LDA-WNH** | oscars, dress, gown, carpet, photo, red, fashion, night, middleton, jenner | Events | 9 |

### 4.5.6 Topical Trends Over Time

To illustrate how different topics are covered by tweets over time, we counted the number of tweets written in each time slot for every topic. Figure 4.4 shows how the 15 topics in the OffAcc dataset were covered by the collection of tweets over a time span of 20 months. The result shows that the number of tweets about topics, such as *Shopping*, *Brands*, *Seasons & Collections*, and *Men's Wear*, was mostly stable throughout the year. Tweeting about *Customers & Services*, *Trends & Styles*, and *Jobs* increased slightly in the period of June-July 2014 and November-December 2014, which reflects the heavy shopping periods such as the annual sale season, Christmas, and New Year. On the other hand, topics such as *Media*, *Fashion Week*, *Celebrities*, and *Beauty & Appearance* were heavily covered by tweets in February-March 2014, June 2014, November-December 2014, and February-March 2015, reflecting major fashion events such as the *Fashion Week*, *Oscars*, *Golden Globe*, and *Grammy Awards*. Knowing how different topics are covered by Twitter during the year can be of great importance to marketing and advertising.

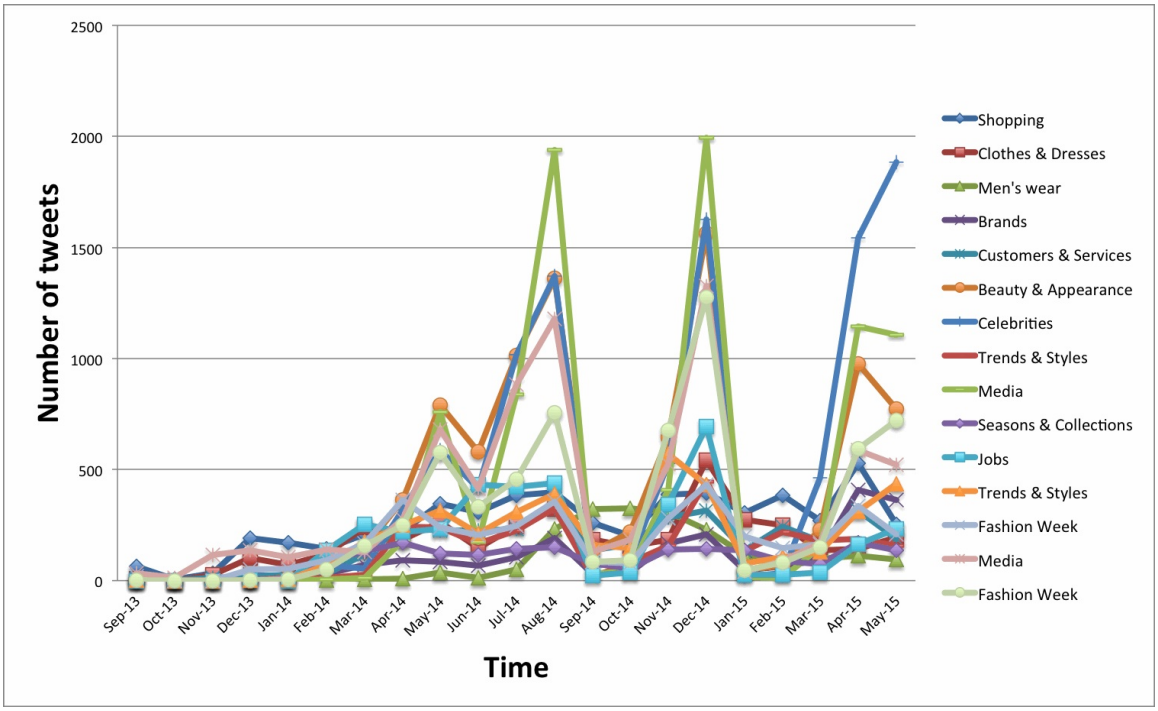


Figure 4.4: Topical trends over time

### 4.5.7 Users' Interests Over Time

To illustrate how the interests of users in these topics have changed over time, we chose two fashion magazine accounts, namely, *LuckyMagazine* and *StyleForum*, and two fashion designers' accounts, namely, *Prada* and *YSL*. Figure 4.5 shows how the tweets written by StyleForum were mainly about *Trends & Styles*, followed by *Beauty & Appearance*, *Celebrities*, and *Events* such as fashion week. These topics were heavily covered during March-April 2014, June-July 2014, November-December 2014, and March-April 2015, reflecting major fashion events during the year. Figure 4.6 shows that the LuckyMagazine tweets were mainly about *Beauty & Appearance*, followed by *Celebrities*. Tweeting about such topics noticeably increased during March-April 2014, June-July 2014, November-December 2014, and March-April 2015, which also reflects the major events in fashion. Fashion designers' interests are shown in Figure 4.7 and Figure 4.8. As shown in Figure 4.7, most of Prada's tweets were about *Customers & Services*. These tweets increased in April 2014, August 2014, February-March 2015, and April 2015. These are usually the times when new seasonal collections are launched by designers. YSL's tweets, on the other hand, were mainly

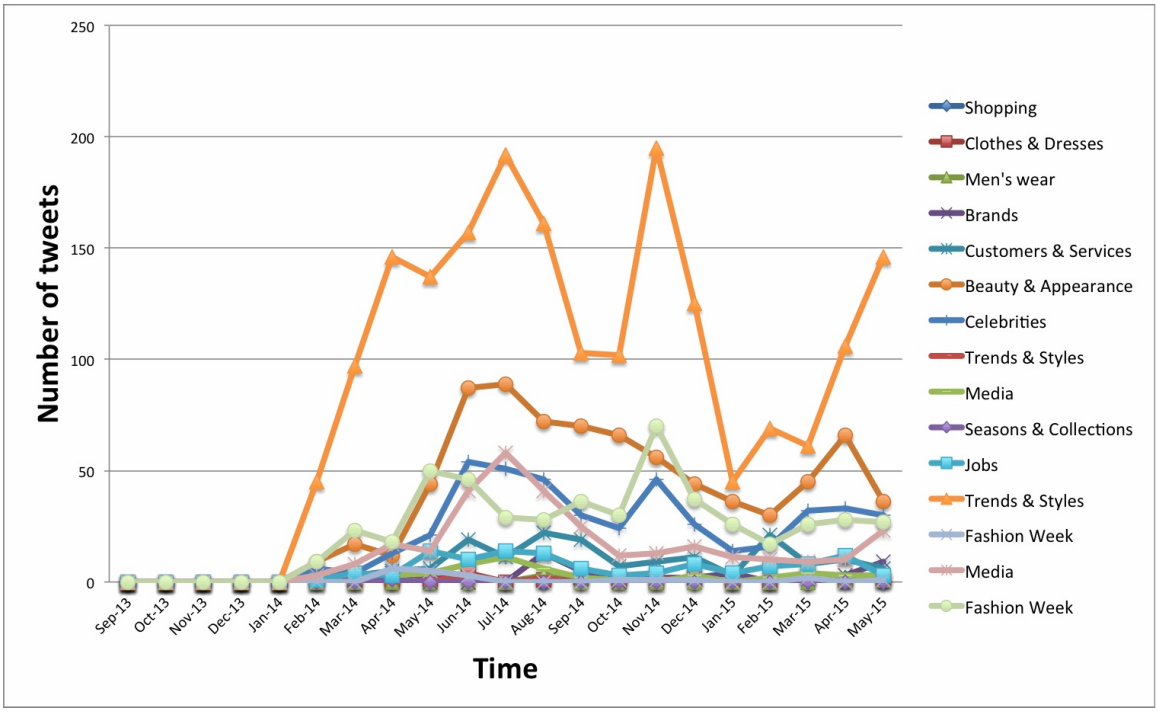


Figure 4.5: StyleForum's tweets over time

about *Seasons & Collections*, as shown in Figure 4.8. Similar to Prada, YSL's tweets were heavily about *Seasons & Collections* during April 2014, July 2014, November-December 2014, and January 2015. In general, we noticed that the use of Twitter by fashion designers is somehow limited, while fashion magazines' tweets are more about fashion events, icons, and trends. Knowing the timing and topics of the fashion designers' and magazines' tweets can greatly help marketing and advertising agencies know when, how, and through which account they can target potential customers.

### 4.5.8 Customized Taxonomy

In this section we demonstrate the results of WordNet customization to include domain-specific terms. Figure 4.9 shows some examples of the fashion-related terms that were added to WordNet. Each sub-figure represents one addition. The new term is represented by the node at the top, while the nodes at the bottom represent the terms already existing in WordNet.

In our experiment, WordNet was customized to include fashion acronyms, brands, communities, and other domain-specific terms. Figure 4.9.a shows how *tbt*[4], a widely used fashion acronym, was

[4]"Stands for (throwback to) to indicate an old photo, idea, etc." Retrieved December 20, 2016, from http://www.

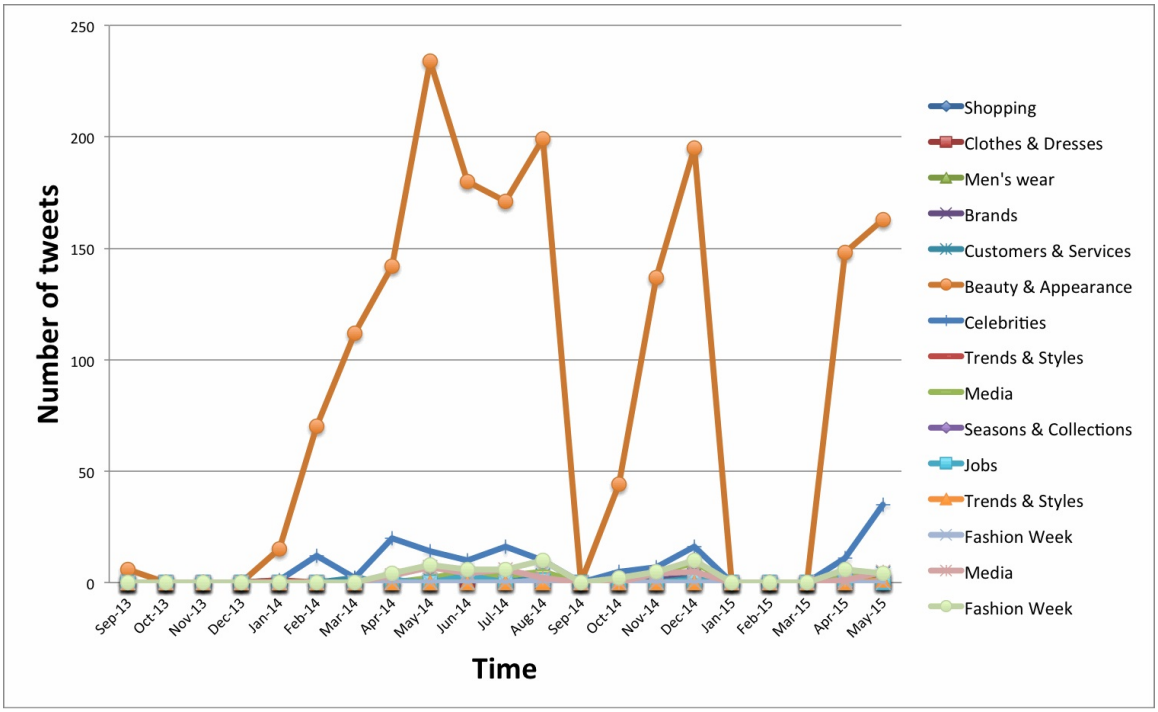


Figure 4.6: LuckyMagazine's tweets over time

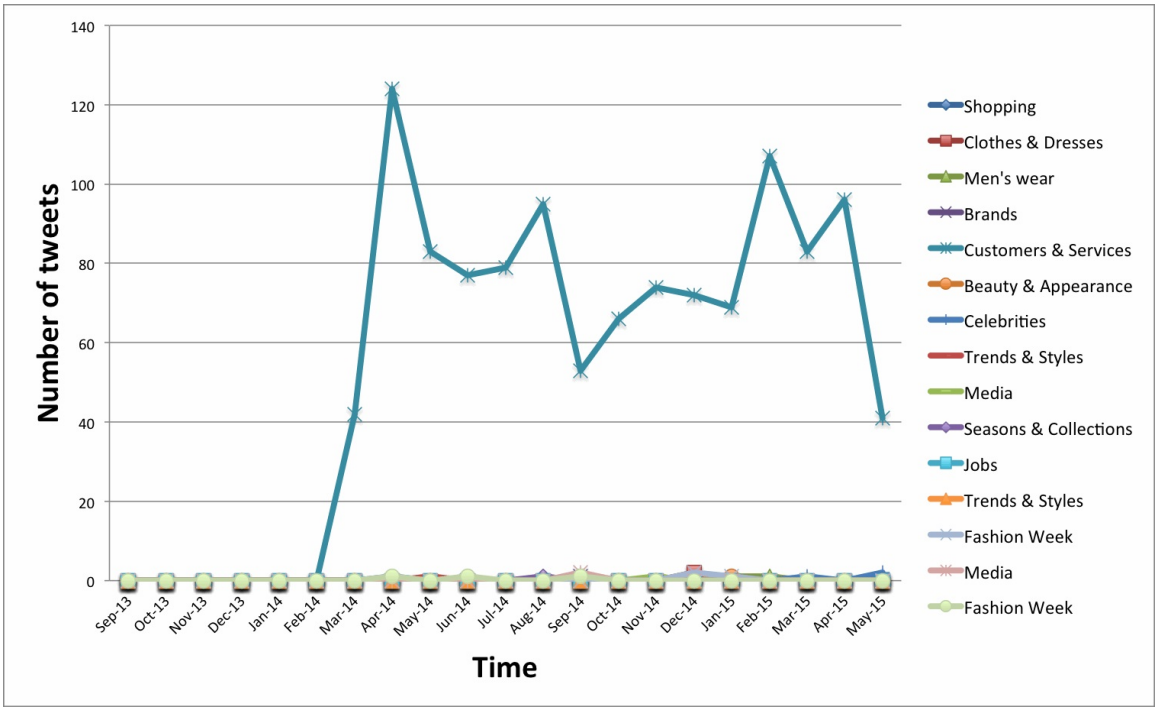


Figure 4.7: Prada's tweets over time

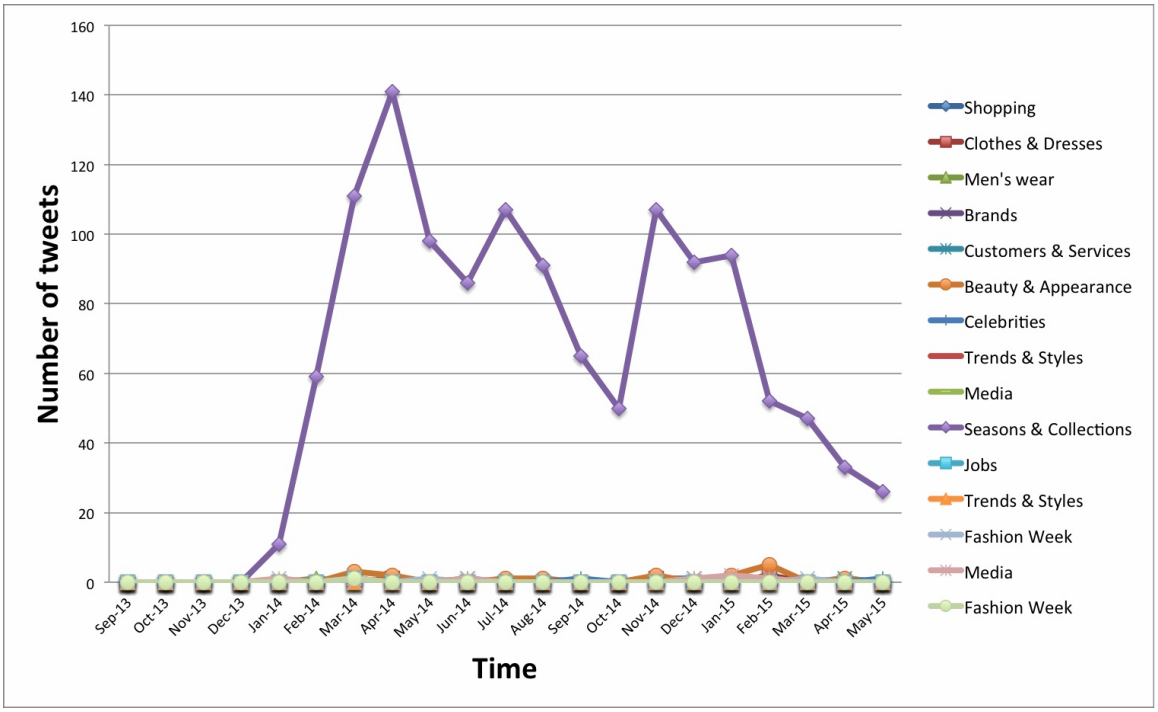


Figure 4.8: YSL's tweets over time

connected to *photo, hair*, and *style*. Similarly, figure 4.9.b shows how *ootd*[5], another acronym, was connected to *style* and *trend*. Figure 4.9.c and figure 4.9.d show how the brand, *Gucci*[6], and the fashion community, *Hijabers*[7], were connected to the term *fashion*. Other domain-specific terms such as *lurex*[8], and *moda*[9] were also added, as shown in figures 4.9.e, and 4.9.f. These examples show how terms were added to WordNet and connected to related terms in the context of fashion. This can be generalized to dynamically build a domain-specific taxonomy for any domain that shares the same characteristics.

urbandictionary.com/define.php?term=TBT

[5]"Outfit Of the Day." Retrieved December 20, 2016, from http://www.urbandictionary.com/define.php?term=OOTD

[6]"An international fashion company." Retrieved December 20, 2016, from http://www.urbandictionary.com/define.php?term=Gucci

[7]"A fashion community for trends in hijab." Retrieved December 20, 2016, from http://erpub.org/siteadmin/upload/8991ER815006.pdf

[8]"A type of fabric." Retrieved December 20, 2016, from https://en.oxforddictionaries.com/definition/lurex

[9] "Fashion, trend, style." Retrieved December 20, 2016, from https://en.wiktionary.org/wiki/moda

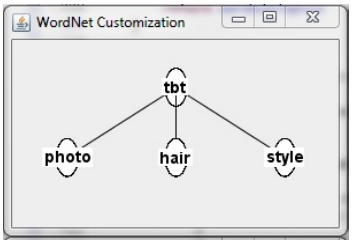


Figure 4.9.a

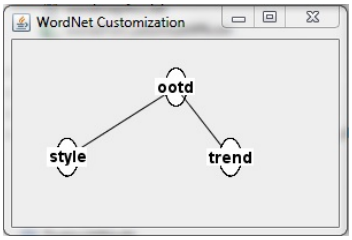


Figure 4.9.b

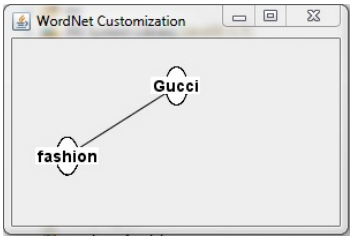


Figure 4.9.c

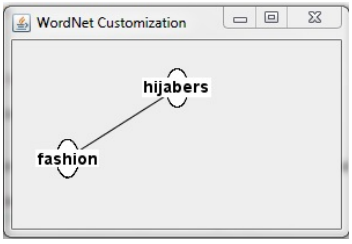


Figure 4.9.d

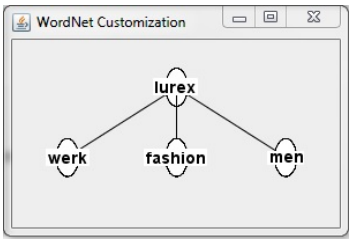


Figure 4.9.e

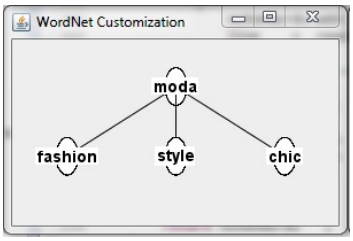


Figure 4.9.f

Figure 4.9: Examples of WordNet customization

## 4.6 Conclusion

In this chapter, we propose a new method that incorporates Twitter-LDA, WordNet, and the set of hashtags available in the corpus with the objective of improving the top probable keywords that represent each topic. Based on the semantic relationships in WordNet and the set of hashtags available in the corpus, the importance of different keywords to different topics is emphasized in the effort of providing the user with a higher quality representation of each topic. A customized version of WordNet is also built to include domain-related terms based on the maximal frequent itemsets found in the corpus. Furthermore, we propose to find the best number of topics covered by the corpus by employing a clustering algorithm to cluster topics based on their similarities in order to get more coherent topics. We further analyze how topics' coverage and users' interests change over time. The proposed method is applied on two real-life fashion datasets collected from Twitter. The obtained results suggest that our method is better than Twitter-LDA in terms of the perplexity, topics' coherence and their quality.

# Chapter 5

# Detecting Breaking News Rumors of Emerging Topics in Social Media

The materials in this Chapter are published in an international journal called Information Processing and Management - A Special Issue on Mining Social Influence and Actionable Insights from Social Networks [5], with impact factor 3.444.

## 5.1 Introduction

Twitter is considered one of the most widely adopted social media platforms for spreading breaking news worldwide. In fact, a recent survey from Pew Research Center stated that "As of August 2017, two-thirds (67%) of Americans get news from social media" and that "about three-quarters (74%) of Twitter users have reported getting news on the site" [88]. The importance of social media, especially Twitter, as a major source of up-to-date information arises from the fact that anyone can instantly post, share, and gather information related to breaking news. This flexibility of sharing and exchanging information comes with a drawback of overwhelming readers with a huge volume of new information every second. Unfortunately, the information is not always trustworthy. This nature of social media provides a fertile ground for rumormongers to post and spread rumors that may result in major chaos and unpredictable reactions from involved individuals.

A real-life example is the single tweet reporting an "Explosion at White House" in 2013. Although this rumor was debunked very quickly, tweets about it spread to millions of users causing an intense impact and a dramatic plunge in the stock market within only six minutes[1]. Such a major impact could have been avoided, or at least minimized, if there were a way to flag that single tweet as a rumor. This example as well as many other real-life examples show how the explosive spread of rumors in social media can lead to extremely damaging impacts on people and society.

Different definitions of rumors have been used in the literature. However, one of the most adopted definitions is in [6] where a rumor is defined as "a story or a statement whose truth value is unverified". Definitions of rumors in major dictionaries also coincide with that. According to these definitions, rumors do not have to be false; they can be deemed later to be true or false. The main characteristic of a rumor is that its truth value is unverified at the time of posting. In relevant studies, there are two types of rumors on social media based on the temporal characteristic: long-standing rumors and breaking news rumors [108]. Long-standing rumors are well-discussed for long periods of time, and one can easily collect related training data under the given topics. In contrast, breaking news rumors generally have not been observed before and require zero-shot learning for real-time detection.

Breaking news refers to "newly received information about an event that is currently occurring or developing"[2]. Most regular news evolves slowly, and more details are expected to be revealed over time. In contrast, breaking news is often unexpected events that evolve dramatically fast without many details on what happened or what will happen next. It covers an unexpected sequence of sub-topics that mostly do not occur in existing data. A typical example was the earthquake of magnitude 9.0 that happened in 2011, followed by a tsunami and the failure of three nuclear reactors in Fukushima. This severe consequence is outside of most people's expectations. The nature of breaking news associates it with a lot of rumors on social media. In fact, the volume of rumors is directly proportional to the importance of and interest in the topic to individuals [6]. Therefore, sensitive topics and breaking news tend to be associated with a huge volume of rumors. This is especially true in the early stages of diffusion when the topic is hot, unclear, and attracting a lot of

[1]Source: http://www.cnbc.com/id/100646197, retrieved on April 2, 2018

[2]Source: https://en.oxforddictionaries.com/definition/breaking_news, retrieved on April 3, 2018

attention.

Real-life incidents of damage and chaos, caused by the spread of rumors in social media during breaking news, have highlighted the urgent need of automatically identifying rumors and verifying their contents. Rumor detection is the task of determining which pieces of information spreading in the social media have unverifiable truth values at the time of posting. This is a crucial and non-trivial task. For long-standing rumors, one can detect or fact-check the incoming text with a training dataset that covers the related events. For breaking news rumors, this data is non-existent and requires zero-shot learning with respect to its temporally evolving topics. It is more challenging to detect breaking news rumors than long-standing ones. First, breaking news covers topics and events that we may not find in the training dataset, which requires a cross-topic consideration in supervised learning. Otherwise, the detection model will very likely overfit the training dataset. Second, breaking news tends to contain new words such as new hashtags or entity names that do not exist in the training dataset. The issue of *Out-of-vocabulary* (OOV) words is another challenge. Emerging rumors contain words that are not in the training samples, especially for the hashtags. Using pre-trained word embedding cannot address this issue because of the new terms that have not been observed before. Moreover, the same terms may have very different meanings when compared to the past, given their context.

To address these challenges, we jointly train a *word2vec* [68] model with an unsupervised objective to learn the word embedding and train a recurrent neural network model with a supervised objective of rumor detection. We propose to train a word2vec model on the fly with the input of a recurrent neural network. Typically, one uses the recurrent neural network to update the word embedding layer. In contrast, we keep a word2vec model parallel to the recurrent neural network and use it to update the embedding space. In this way, our model can incrementally learn the distributed vector representations of words in the input text, capture the deep latent features and their correlations from it, and use them to build a detection model of breaking news rumors. Furthermore, learning the distributed vector representations of terms allows our model to better handle new OOV words of emerging topics of breaking news that were not seen during the training process. We find such a simple design effective to address the aforementioned challenges.

Related research mostly focuses on long-standing rumors. These rumors usually spread on

social media websites for a while, causing streams of posts questioning their truth and looking for a confirmation. Thus, long-standing rumors are already known to be rumors and detecting them is relatively straightforward. For that reason, existing work handling long-standing rumors aims at tracking the diffusion of rumors, classifying opinions expressed toward them, or predicting their veracity [108]. In contrast, we aim at detecting emerging rumors of breaking news, which is more challenging. During early stages of breaking news diffusion, when the topic is still hot, emerging rumors spread very fast in social media, with not many posts discussing their truth. In contrast, people tend to spread these rumors and act upon them immediately, which can be extremely damaging. Furthermore, because breaking news tends to generate new unseen topics, work on detecting emerging rumors of breaking news has to be able to handle topic shift issues.

Most existing studies on rumor detection also suffer from another issue: they assume that rumors are always false and aim at predicting these false rumors [108]. This is demonstrated by the design of their experiments where they train their detection models on datasets of long-lasting rumors with the objective of detecting false rumors. This assumption is invalid because rumors are not always false. The term 'rumor' refers to unverified information that can be deemed later to be true or false. Instead, we aim at detecting emerging rumors regardless of their truth value. The goal is to flag micro-posts as rumors, i.e., micro-posts that contain unverified information during the rapid diffusion, and thus minimize their harmful consequences.

In this chapter we study the problem of automatically identifying rumors spreading in social media during breaking news diffusion. We propose a new method that incorporates deep learning and representation learning algorithms to automatically identify rumors in social media. The main contributions of this work can be summarized as follows:

- We propose a new semi-supervised learning solution for breaking news rumor detection by combining an unsupervised learning objective with a supervised learning objective. To the best of our knowledge, this is the first work that employs representation learning with a deep learning model for the purpose of emerging breaking news rumor detection on social media.

- We propose a new strategy to update word embeddings on the fly with the training process to mitigate the cross-topic and OOV issue in breaking news rumor detection. In contrast

to existing work, we do not train our model based on hand-crafted features. Instead, our proposed model learns distributed representations on parallel to the supervised training.

- Experimental results on real-life datasets suggest that our proposed method outperforms the state-of-the-art sequential classifier [107], as well as other classifiers in terms of precision, recall, and F1.

There is very limited work targeting the challenge of identifying unverified information circulating social media. The work in [105] is one of the earliest works in this category. The authors proposed to first identify "signal tweets" based on a hand-crafted list of regular expressions. Our proposed model learns the features automatically rather than using a predefined hand-crafted regular expressions list. Recently, a sequential classifier model based on the *Conditional Random Fields (CRF)* was proposed to learn the context of an event from the sequence of tweets seen so far and use it to classify the current tweets [107]. Our model predicts the class of the micro-post solely based on its text. It does not need historical data or a trail of micro-posts regarding the information in question.

The rest of the chapter is organized as follows. Section 5.2 provides the research problem followed by an overview of the proposed model. Section 5.3 describes the experiments and provides a detailed discussion of the results.

## 5.2 Deep Learning Model for Breaking News Rumors Detection

This section first formally defines the research problem and then presents the proposed deep learning model for detecting breaking news rumors in social media.

### 5.2.1 Problem Statement

The research problem of breaking news rumors detection can be defined as follows: for a given micro-post regarding a specific piece of information, the task is to determine if it is a rumor or not. This problem can be formulated as a binary classification problem as follows: let $mp = \langle w_1, ..., w_T \rangle$ be a sequence of words in a micro-post $mp$ of length $T$. Given $mp$ as an input, the goal is to classify it as a rumor or not by assigning a label from $RL = \{R, NR\}$.

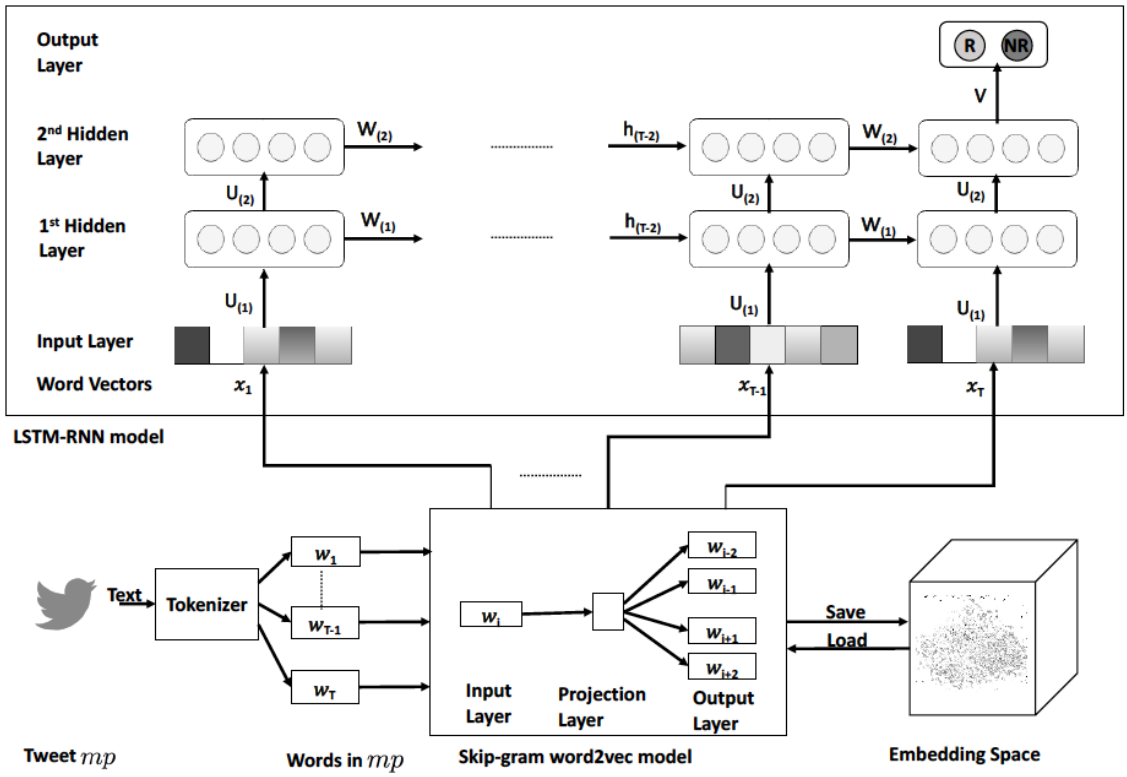


Figure 5.1: The proposed breaking news rumors detection model. A micro-post $mp$ is first tokenized into a sequence of words $mp = \langle w_1, ..., w_T \rangle$. Next, the word2vec model converts the sequence of words into a sequence of vectors $x = \langle x_1, \ldots, x_T \rangle$ and passes it through weighted connections to the LSTM-RNN model. Finally, the LSTM-RNN model predicts the class as the output vector at the last time step $T$

### 5.2.2 Proposed Model

This section provides an overview of the proposed breaking news rumors detection model. The proposed model jointly trains a word2vec model with an unsupervised objective to learn the word embedding and train a recurrent neural network model with a supervised objective of rumor detection. Fig. 5.1 illustrates the architecture of the proposed model. We will start by describing two main components of our model, namely word2vec and LSTM-RNN, followed by a brief description on how the two models are jointly trained using the input data.

#### 5.2.2.1 word2vec

A word2vec model is a neural network that takes a text corpus as an input and produces real-valued low-dimensional vector representations for words that appear in that corpus. Thus, it converts textual data into distributed vector representations that can be then fed into deep neural networks for different purposes. These vector representations are called *word embeddings*. In this work, we use a technique called *skip-gram* to train the word2vec model [68] given its better effectiveness compared to the *cbow* model. Given a corpus of text, skip-gram builds the word2vec model as follows. Let $w_i$ be a word in the corpus, and let the set of words surrounding $w_i$ within a specified window size in a sentence be the context of $w_i$. To build the word2vec model, skip-gram takes each word $w_i$ along with its context words and learns their word representations. The learning objective here is to find useful representations of these words in the embedding space so that the model can, given any other word $w_t$, predict its surrounding context words with high probabilities and the others with low probability [68]. Formally, given a sequence of words $mp = \langle w_1, ..., w_T \rangle$ and a context window of size $\mathfrak{z}$, the objective of a skip-gram model is to maximize the following average log probability function:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-\mathfrak{z}\leq j\leq \mathfrak{z}, j\neq 0} \log p\left(w_{t+j}|w_t\right) \tag{18}$$

where $\log p\left(w_{t+j}|w_t\right)$ is approximated using negative sampling as follows:

$$\log p\left(w_{t+j}|w_t\right) = \log \sigma({\mathbf{v}'_{w_{t+j}}}^{\top}\mathbf{v}_{w_t}) + \sum_{i=1}^{\mathbf{k}} E_{w_i\sim P_n(w_{t+j})}\left[\log \sigma(-{\mathbf{v}'_{w_i}}^{\top}\mathbf{v}_{w_t})\right] \tag{19}$$

where $\mathbf{v}_{w_t}$ and $\mathbf{v}'_{w_{t+j}}$ denote the input and output vector representations of words $w_t$ and $w_{t+j}$; $\mathbf{k}$ denotes the number of negative samples for each data sample, and $P_n(w_{t+j})$ denotes the noise distribution [68].

#### 5.2.2.2 LSTM-RNN

*Long Short-Term Memory (LSTM)* is an extended *Recurrent Neural Network (RNN)* architecture designed to overcome the limitation of standard RNNs in storing information about previous input [36, 41]. In standard RNNs, the gradient of the current time stamp completely depends on the

next time stamp during the back-propagation step, which will cause the gradient to either vanish or explode [63]. LSTM-RNN provides more channels for the gradient to flow back from time step $t$ to time step $t-1$ by introducing the concept of gates. The gradients do not completely depend on a single time stamp and the vanishing or exploding issues are mitigated by gating.

LSTM-RNN introduces a memory cell $m_t$ at each time step $t$. In this case, the algorithm iterates over the following equations to update the hidden states of the network and generate the outputs [36]:

$$\mathsf{i}_t = \sigma\left(\mathsf{W}_\mathsf{i} x_t + \mathsf{U}_\mathsf{i} \mathsf{h}_{t-1} + \mathsf{V}_\mathsf{i} m_{t-1} + \mathsf{b}_\mathsf{i}\right) \tag{20}$$

$$f_t = \sigma\left(\mathsf{W}_f x_t + \mathsf{U}_f \mathsf{h}_{t-1} + \mathsf{V}_f m_{t-1} + \mathsf{b}_f\right) \tag{21}$$

$$m_t = f_t m_{t-1} + \mathsf{i}_t tanh\left(\mathsf{W}_m x_t + \mathsf{U}_m \mathsf{h}_{t-1} + \mathsf{b}_m\right) \tag{22}$$

$$\mathsf{o}_t = \sigma\left(\mathsf{W}_\mathsf{o} x_t + \mathsf{U}_\mathsf{o} \mathsf{h}_{t-1} + \mathsf{V}_\mathsf{o} m_t + \mathsf{b}_\mathsf{o}\right) \tag{23}$$

$$\mathsf{h}_t = \mathsf{o}_t tanh\left(m_t\right) \tag{24}$$

where $\sigma$ is the logistic sigmoid function, and $\mathsf{i}$, $f$, $\mathsf{o}$, and $m$ are the input, forget, output gates, and the cell input activation vector, respectively. The terms $\mathsf{W}$, $\mathsf{U}$, and $\mathsf{V}$ denote weight matrices connecting hidden to hidden, input to hidden, and hidden to output layers, respectively; the term $\mathsf{h}$ denotes a hidden state, and the term $\mathsf{b}$ denotes a bias vector.

For each training micro-post, the predicted class is calculated using a softmax layer with the objective of minimizing the following cross entropy loss:

$$\mathsf{L} = -\frac{1}{\mathsf{N}} \sum_{i=1}^{\mathsf{N}} \left[y \log(p) + (1-y) \log\left(1-p\right)\right] \tag{25}$$

where $\mathsf{N}$ represents the number of training samples, $y$ represents the actual class, and $p$ represents the predicted class. Softmax layers are used to calculate the output of a neural network as a probability for each class in the range from 0 to 1. The main advantage of the softmax layer is that it provides the flexibility of predicting each class with a certain probability rather than predicting only one class.

#### 5.2.2.3 Model training

We train our model as follows. We first feed the training corpus of micro-posts to the combined skip-gram-word2vec model, which automatically learns the distributed vector representation of each word, i.e., word embedding. This converts the sequence of words in $mp$ into a sequence of vectors $x = \langle x_1, \ldots, x_T \rangle$ that is passed through weighted connections to a stack of LSTM hidden layers to compute the hidden vector sequences $\mathsf{h} = \langle \mathsf{h}_1, \ldots, \mathsf{h}_\mathsf{T} \rangle$. The predicted class is then calculated as the output vector at the last time step $o_T$ of the LSTM-RNN model.

To help the training process mitigate the cross-topic and OOV issues in breaking news rumor detection, we keep the word2vec model parallel to the recurrent neural network model and use it to update the embedding space on the fly. By designing our model in this way, we incrementally learn the distributed vector representations of words in the input text, capture the latent features and their correlations from the text, and use them to build a detection model of breaking news rumors. We compare the performance of different embedding training strategies in Section 5.3.5.3. The experimental result shows that this approach significantly outperforms the typical methods of embedding training.

## 5.3 Experiment

This section first describes the datasets, baseline methods, features sets, and experimental settings. Next, the obtained results are discussed in detail. Finally, two case studies of real-life breaking news events are presented.

### 5.3.1 Datasets

In real-life, breaking news tends to generate new unseen topics that have not been observed before and do not exist in the training data. Thus, a representative dataset does not exist and new breaking news stories will always bring previously unseen events and be associated with a sequence of unexpected sub-topics. Consequently, work on detecting emerging rumors of breaking news has to be able to handle topic shift issues and this is what makes our problem more challenging.

In our experiments, we used five sets of real-life tweets from *PHEME* [106], where each set is

related to a different piece of breaking news. *PHEME* is publicly accessible. Table 5.1 summarizes the percentages of rumors and non-rumors tweets in each of them.

Table 5.1: Percentages of rumors and non-rumors tweets in the PHEME datasets

| **Breaking News** | **Rumors** | **Non-rumors** |
|---|---|---|
| **Charlie Hebdo** | 458 (22.0%) | 1,621 (78.0%) |
| **Ferguson** | 284 (24.8%) | 859 (75.2%) |
| **Germanwings Crash** | 238 (50.7%) | 231 (49.3%) |
| **Ottawa Shooting** | 470 (52.8%) | 420 (47.2%) |
| **Sydney Siege** | 522 (42.8%) | 699 (57.2%) |

### 5.3.2 Baselines and Feature Sets

To evaluate our model, we compared it with the state-of-the-art sequential classifier proposed in [107]. We also compared our model with other non-sequential classifiers that were used extensively as baselines in the literature, including *Support Vector Machine (SVM)*, *Naive Bayes (NB)*, *Random Forest (RF)*, and *Maximum Entropy (ME)*.

To train the baseline classifiers, we used the same sets of content-based and social-based features that yielded the state-of-the-art performance in [107]. Table 5.2 summarizes these two sets of features.

### 5.3.3 Experimental Settings

To simulate a real-life cross-topic emerging rumor detection scenario, we performed a 5-fold cross-validation as follows. In each run, we used the datasets of four breaking news stories to train our model as well as the baseline classifiers. The fifth dataset was then used to evaluate the performance of these classifiers in terms of precision, recall, and F1. Thus, in each of the five runs, the dataset used for the evaluation represents breaking news rumors of unseen topics. Furthermore, to insure the stability of the reported results and get a more robust estimation of the classification performance of our deep learning model, we repeated each run of the 5-fold cross-validation for each model configuration five times. In the rest of this chapter, the classification performance of the proposed model is reported as the $mean \pm variance$ of five repetitions of the 5-fold cross-validation

Table 5.2: Content-based and social-based features

| Category | Features |
|---|---|
| **Content-based** | word vectors |
| | Capital ratio: ratio of capital letters |
| | #Qmark: number of question marks |
| | #Emark: number of exclamation marks |
| | #Periods: number of periods |
| | #Words: number of words |
| **Social-based** | #Tweets: number of tweets written by the author |
| | #Lists: number of lists that include the author's account |
| | Follow ratio: the following ratio of the author's account |
| | Age: the age of the author's account |
| | Verified: whether the account of the author is verified or not |

instead of a single 5-fold cross-validation run.

The proposed model was implemented using *JetBrains IntelliJ IDEA*[3] development environment and *Deeplearning4j*[4] machine learning library. We ran our experiments on a machine running *Windows server 2016 Datacenter*. The machine is powered by an Intel Xeon E5-1650 v4 processor at 3.60 GHz with 32GB of RAM.

### 5.3.4 Evaluation Measures

To evaluate the classification performance of the proposed model as well as the baseline models in our experiments, we used the *precision*, *recall*, and *F1 score*. The precision (positive predictive value) is the ratio of the correctly classified positive micro-posts by the model to the total classified positive micro-posts. It is calculated as follows:

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{26}$$

The recall (sensitivity) is the ratio of the correctly classified positive micro-posts to the all micro-posts in actual class. It is calculated as follows:

[3]Source: https://www.jetbrains.com/idea/, retrieved on September 28, 2018
[4]Source: https://deeplearning4j.org/, retrieved on September 28, 2018

$$Precision = \frac{TruePositive}{TruePositive + FalseNegative} \tag{27}$$

The F1 score is the harmonic mean of the precision and recall. It balances the precision and recall of the classification model and is calculated as follows:

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{28}$$

### 5.3.5 Results

#### 5.3.5.1 Comparison with baseline classifiers

To compare the performance of our proposed model with the baseline classifiers, we performed a 5-fold cross-validation and reproduced the results of [107], as shown in Table 5.3. The reported values are the micro-averaged scores across all five runs in terms of precision, recall, and F1 for both classes: rumors and non-rumors. Bold values indicate the best classification performance among all classifiers. For our proposed model, the reported values are the micro-averaged $\pm$ variance scores across five repetitions of the 5-fold cross-validation. As shown in the table, results for the rumors class suggest that among all baseline classifiers, NB had the best performance in terms of recall, while Conditional Random Fields (CRF) performed the best in terms of precision and F1. This is consistent with the results reported in [107]. Table 5.3 also shows that our proposed model outperformed CRF in terms of precision, recall, and F1.

For the non-rumors class, similar results were obtained. Among all baseline classifiers, CRF had the best performance in terms of precision and F1, while NB performed the best in terms of recall. Our proposed model also outperformed all baselines in terms of precision and F1. It achieved a high recall as well.

Table 5.3 also shows that our proposed model had the best overall performance for both classes: rumors and non-rumors, compared to all baseline classifiers in terms of F1. These results suggest that our model outperformed all baseline classifiers, including the state-of-the-art model, in detecting breaking news rumors using *only the text of tweets* as input without any social-based features.

Table 5.3: Micro-averaged precision (p), recall (R), and F1 scores of detecting rumors and non-rumors across all five runs for baseline classifiers and our proposed model

| Classifier | Features | Rumors | | | Non-rumors | | | All Classes |
|---|---|---|---|---|---|---|---|---|
| | | P | R | F1 | P | R | F1 | F1 |
| **Support Vector Machine (SVM)** | **Content-based** | 0.351 | 0.431 | 0.387 | 0.668 | 0.590 | 0.626 | 0.536 |
| | **Social-based** | 0.347 | 0.479 | 0.402 | 0.666 | 0.536 | 0.594 | 0.517 |
| | **Combined** | 0.353 | 0.457 | 0.399 | 0.671 | 0.569 | 0.616 | 0.531 |
| **Random Forest (RF)** | **Content-based** | 0.299 | 0.092 | 0.141 | 0.655 | 0.889 | 0.754 | 0.618 |
| | **Social-based** | 0.343 | 0.460 | 0.393 | 0.662 | 0.545 | 0.598 | 0.495 |
| | **Combined** | 0.326 | 0.104 | 0.158 | 0.658 | 0.889 | 0.757 | 0.622 |
| **Naive Bayes (NB)** | **Content-based** | 0.402 | **0.767** | 0.527 | 0.775 | 0.412 | 0.538 | 0.533 |
| | **Social-based** | 0.259 | 0.011 | 0.020 | 0.659 | **0.984** | 0.789 | 0.653 |
| | **Combined** | 0.402 | 0.767 | 0.527 | 0.775 | 0.412 | 0.538 | 0.533 |
| **Maximum Entropy (ME)** | **Content-based** | 0.362 | 0.473 | 0.410 | 0.678 | 0.570 | 0.619 | 0.537 |
| | **Social-based** | 0.368 | 0.495 | 0.422 | 0.684 | 0.563 | 0.617 | 0.540 |
| | **Combined** | 0.364 | 0.472 | 0.411 | 0.679 | 0.575 | 0.623 | 0.540 |
| **Conditional Random Field (CRF)** | **Content-based** | 0.687 | 0.544 | 0.607 | 0.788 | 0.872 | 0.828 | 0.761 |
| | **Social-based** | 0.467 | 0.259 | 0.333 | 0.690 | 0.848 | 0.761 | 0.648 |
| | **Combined** | 0.665 | 0.548 | 0.601 | 0.787 | 0.858 | 0.821 | 0.752 |
| **Proposed model** | **words** | **0.728** ±0.002 | 0.706 ±0.0005 | **0.716** ±0.001 | **0.833** ±0.0003 | 0.847 ±0.001 | **0.839** ±0.0004 | **0.795** ±0.001 |
| | **Combined** | 0.619 ±0.005 | 0.670 ±0.003 | 0.639 ±0.001 | 0.821 ±0.0003 | 0.778 ±0.008 | 0.796 ±0.003 | 0.741 ±0.002 |

#### 5.3.5.2 Experimenting with syntactic representations of posts

To further evaluate the classification performance of our model, we experimented with the following syntactic representations of tweets as our input:

- *Part-Of-Speech tags (POS)*. Inspired by work on sensitive text detection [67], we wanted to explore whether or not representing a tweet as a sequence of POS tags can lead to better classification performance. We used GATE Twitter part-of-speech tagger, known as *"Twittie"*[5], to tag words in our datasets. Then, we replaced each word in every tweet by its POS tag and used the sequences of POS tags as our input.

- *N-gram words and N-gram characters*. We also represented each input tweet as a sequence of N-gram words or N-gram characters to further explore whether or not such representations can improve the classification performance of our model.

In this experiment, we set $N = 1, 2, 3$ for N-gram words and $N = 3, 5, 7$ for N-gram characters. We then performed 5 repetitions of a 5-fold cross-validation and evaluated our model using different

[5]Source: https://gate.ac.uk/wiki/twitter-postagger.html, retrieved on January 24, 2018

Table 5.4: Micro-averaged mean ± variance of precision (p), recall (R), and F1 scores of detecting rumors and non-rumors across all five runs for our proposed model using other syntactic features

| Features | Rumors | | | Non-rumors | | | All Classes |
|---|---|---|---|---|---|---|---|
| | **P** | **R** | **F1** | **P** | **R** | **F1** | **F1** |
| **1-gram words** | 0.728 ±0.002 | 0.706 ±0.0005 | 0.716 ±0.001 | **0.833** ±0.0003 | 0.847 ±0.001 | 0.839 ±0.0004 | **0.795** ±0.001 |
| **2-gram words** | 0.478 ±0.002 | 0.431 ±0.007 | 0.447 ±0.002 | 0.706 ±0.004 | 0.737 ±0.009 | 0.719 ±0.005 | 0.631 ±0.004 |
| **3-gram words** | **0.884** ±0.0002 | **0.740** ±0.001 | **0.806** ±0.0002 | 0.542 ±0.001 | 0.759 ±0.003 | 0.632 ±0.002 | 0.746 ±0.0004 |
| **3-gram characters** | 0.420 ±0.002 | 0.612 ±0.009 | 0.494 ±0.002 | 0.734 ±0.001 | 0.555 ±0.014 | 0.626 ±0.007 | 0.575 ±0.003 |
| **5-gram characters** | 0.496 ±0.003 | 0.589 ±0.008 | 0.533 ±0.002 | 0.778 ±0.001 | 0.700 ±0.011 | 0.732 ±0.004 | 0.662 ±0.003 |
| **7-gram characters** | 0.316 ±0.031 | 0.199 ±0.022 | 0.239 ±0.026 | 0.646 ±0.001 | 0.788 ±0.004 | 0.709 ±0.001 | 0.583 ±0.002 |
| **Part Of Speech (POS) tags** | 0.433 ±0.021 | 0.154 ±0.004 | 0.207 ±0.005 | 0.793 ±0.001 | **0.927** ±0.005 | **0.853** ±0.0002 | 0.752 ±0.0004 |

input representations. Table 5.4 shows the micro-averaged ± variance scores across five repetitions of the 5-fold cross-validation in terms of precision, recall, and F1 for both classes: rumors and non-rumors. Bold values indicate which input representation yielded the best classification performance of our model. For the rumors class, the results suggest that representing the input tweets as sequences of 3-gram words yielded the best classification performance over all other representations in terms of precision, recall, and F1. 2-gram words also yielded a good classification performance. On the other hand, representing tweets as sequences of N-gram characters did not yield as good of a performance as N-gram words. The results also suggest that using POS tags representations yielded high classification performance in terms of precision, but the recall and F1 scores were low. For the non-rumors class, among all input representations, using the text of the tweet yielded the best classification performance in terms of precision, while the POS tags representation yielded the best classification performance in terms of recall and F1.

Table 5.4 also shows that our proposed model yielded the best overall performance in terms of F1 for both classes when the input is simply the text of the tweets. The results also suggest that the classification performance for the non-rumors class is better than the rumors class. By observing many examples of the non-rumors micro-posts, we noticed that they were written in a more formal

way and have a better syntactical construction than the rumors micro-posts.

#### 5.3.5.3 Comparison of Different Embedding Training Strategies

To assess if knowledge transfer can help improve the classification performance of our deep learning model, we compared the performance of our model using three different settings for learning the distributed vector representation of words via the word2vec model:

- *Static word2vec model*. In this setting, during the training phase we used the training datasets to jointly learn the word2vec and LSTM-RNN models. Then, to evaluate our model, the word2vec model was used as a lookup table to transform every new tweet in the testing dataset into a sequence of vector representations of its words, which was then fed into the LSTM-RNN model.

- *Dynamic word2vec model*. In this setting, during the training phase we used the training datasets to jointly learn the word2vec and LSTM-RNN models. Then, to evaluate our model the word2vec model was incrementally up-trained and updated while classifying every new tweet in the testing dataset.

- *Up-trained Google word2vec model*[6]. In this setting, instead of learning the distributed vector representations of words from scratch, we used a general word2vec model as our initial distributed victor representations of words. This model was trained on Google's news dataset to contain three million words and phrases, each represented as a 300-dimensional vector in the embedding space. During the training phase, Google's word2vec model was first up-trained using our training datasets in parallel with building the LSTM-RNN model. Then, to evaluate our model, this word2vec model was incrementally up-trained and updated while classifying every new tweet in the testing dataset.

Table 5.5 shows the micro-averaged $\pm$ variance scores of our model under each of the three settings across five repetitions of the 5-fold cross-validation in terms of precision, recall, and F1 for both classes: rumors and non-rumors. Bold values indicate which setting yielded the best classification performance of our model. The results suggest that using a dynamic word2vec setting yielded a

[6]Source: https://code.google.com/archive/p/word2vec/, retrieved on May 11, 2018

Table 5.5: Micro-averaged mean ± variance of precision (p), recall (R), and F1 scores of detecting rumors and non-rumors across all five runs for our proposed model under different settings of training word2vec model

| **Word2vec model** | **Rumors** | | | **Non-rumors** | | | **All Classes** |
|---|---|---|---|---|---|---|---|
| | **P** | **R** | **F1** | **P** | **R** | **F1** | **F1** |
| **Static model** | 0.710 ±0.003 | 0.696 ±0.002 | 0.703 ±0.002 | 0.716 ±0.0005 | 0.747 ±0.001 | 0.731 ±0.0002 | 0.734 ±0.001 |
| **Dynamic model** | **0.728** ±0.002 | **0.706** ±0.0005 | **0.716** ±0.001 | **0.833** ±0.0003 | **0.847** ±0.001 | **0.839** ±0.0004 | **0.795** ±0.001 |
| **Up-trained Google model** | 0.668 ±0.001 | 0.552 ±0.0003 | 0.604 ±0.003 | 0.751 ±0.002 | 0.816 ±0.002 | 0.782 ±0.0007 | 0.719 ±0.002 |

significantly better classification performance than the static word2vec for the rumors class in terms of recall and F1, while it improved the performance on the non-rumors class in terms of precision, recall, and F1. In the experiment the size of the testing dataset is smaller than the training set. Since the quality of the distributed vector representation of words tends to increase significantly with the amount of the input data, the dynamic word2vec setting should yield even better classification performance in the long term. The results also suggest that although the idea of transfer knowledge using a pre-trained embedding from Google seems promising, it did not improve the classification performance of our model in terms of precision, recall, or F1. These results suggest that building the word2vec model in parallel with building the LSTM-RNN model helps the rumors detection model learn the latent features and their correlations from the input text. Furthermore, updating the word2vec model incrementally with every new tweet helps the model mitigate the topic-shifts and OOV issues associated with emerging breaking news rumors.

Table 5.6: Precision scores of different classifiers before and after using social-based features associated with each dataset

| **Dataset** | **NB** | | **ME** | | **RF** | | **SVM** | | **CRF** | | **Proposed Model** | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Before** | **After** | **Before** | **After** | **Before** | **After** | **Before** | **After** | **Before** | **After** | **Before** | **After** |
| **Charlie Hebdo** | 0.756 | 0.756 | 0.568 | **0.578** | 0.687 | 0.687 | 0.571 | 0.556 | 0.823 | 0.807 | 0.684 | 0.630 |
| **Ferguson** | 0.253 | **0.254** | 0.563 | **0.578** | 0.714 | 0.704 | 0.519 | **0.543** | 0.778 | 0.773 | 0.680 | **0.714** |
| **Germanwings Crash** | 0.508 | 0.508 | 0.520 | 0.510 | 0.484 | **0.505** | 0.582 | 0.548 | 0.731 | 0.702 | 0.806 | 0.802 |
| **Ottawa Shooting** | 0.527 | 0.527 | 0.501 | 0.491 | 0.478 | **0.484** | 0.512 | 0.508 | 0.697 | 0.709 | 0.896 | 0.844 |
| **Sydney Siege** | 0.428 | 0.428 | 0.494 | 0.488 | 0.564 | **0.582** | 0.491 | 0.488 | 0.697 | 0.691 | 0.803 | 0.779 |

Table 5.7: Importance scores of each of the features in each dataset measured as the gain ratio between this feature and the true class label

| Dataset | Content-based Features | | | | | Social-based Features | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Capital Ratio | #Qmark | #Emark | #Periods | #Words | #Tweets | #Lists | Follow Ratio | Age | Verified |
| **Charlie Hebdo** | 0.010 | **0.032** | **0.024** | 0.008 | 0.005 | 0.015 | 0.020 | 0.000 | 0.009 | **0.038** |
| **Ferguson** | 0.011 | **0.020** | 0.005 | 0.010 | **0.018** | 0.010 | **0.014** | 0.000 | 0.003 | 0.000 |
| **Germanwings Crash** | 0.010 | **0.019** | 0.004 | **0.029** | 0.007 | 0.004 | 0.008 | 0.000 | 0.005 | **0.022** |
| **Ottawa Shooting** | **0.031** | **0.127** | **0.054** | 0.003 | 0.019 | 0.020 | 0.023 | 0.000 | 0.016 | 0.003 |
| **Sydney Siege** | **0.054** | 0.047 | **0.056** | 0.004 | 0.005 | 0.029 | **0.105** | 0.000 | 0.008 | 0.044 |

#### 5.3.5.4 Characterizing datasets

During our experiments we observed that using social-based features in addition to content-based features as our input did not always improve the classifiers. In this section we aim to assess the effect of adding the social-based features to the content-based features of each of the datasets on the classification performance. We started by evaluating the precision of each classifier on each dataset twice: once using only content-based features and another using both social-based features and content-based features as our input. Table 5.6 shows the obtained results. Bold values indicate cases where the precision of a classifier was improved after adding social-based features. The results show that the precisions of four classifiers were improved after adding the social-based features for the *Ferguson* dataset compared to only one classifier for the rest of the datasets.

These results led us to analyze the social-based and the content-base features of each of the datasets. We started by measuring the importance of each of the features in predicting the true class of tweets in each of the datasets using the *gain ratio* feature selection algorithm [2]. Table 5.7 shows the obtained results. Bold values indicate the top important features in each case. The results show that the number of lists that include the author's account, denoted by *#Lists*, is an important social-based feature for the *Ferguson* and the *Sydney Siege* datasets, while *verified* (whether the author's account is verified or not) is an important social-based feature for *Charlie Hebdo* and *Germanwings Crash* datasets. We further analyzed the social-based features of each of the datasets and used the *Standard Deviation (SD)* to measure the amount of variation in their values. Table 5.8 shows the obtained results. Bold values indicate cases where the SD value of the feature in a dataset varies significantly from the rest of the datasets. The standard deviation values in the table show the

sparsity in the values of each social-based feature in each one of the five datasets. Each column represents the amount of variation in one social-based feature. The different scales are due to the fact that different features have very different value scales. As shown in the table, among the four datasets with important social-based features, the *Ferguson* dataset can be characterized by the very low SD value of the *#Lists* feature compared to the rest of the datasets. Similarly, the *Sydney Siege* dataset can be characterized by the high SD value of the *#Lists*. On the other hand, the SD values of the *Verified* feature in the *Charlie Hebdo* and *Germanwings Crash* datasets are almost the same as the rest of the datasets, which does not help characterize these datasets.

By comparing our results in Tables 5.6, 5.7, and 5.8, we observed that although the *Ferguson* and the *Sydney Siege* datasets can be distinguished from the other datasets by having a social-based feature with high important score and very different SD value, adding the social-based features improved the classification performance for most classifiers for the first dataset, compared to only one classifier for the second one. The very high SD value of the *#Lists* feature in the *Sydney Siege* dataset suggests much higher sparsity in its values. Consequently, instead of improving the classification performance, adding this feature actually worsened it.

Table 5.8: Standard Deviation values of social-based features for the PHEME datasets

| **Dataset** | **#Tweets** | **#Lists** | **Follow Ratio** | **Age** | **Verified** |
|---|---|---|---|---|---|
| **Charlie Hebdo** | 56305.081 | 33348.537 | 1.552 | 1.950 | 0.498 |
| **Ferguson** | 58165.469 | **12054.331** | 1.094 | 1.783 | 0.483 |
| **Germanwings Crash** | 67650.101 | 30550.214 | 1.438 | 2.158 | 0.483 |
| **Ottawa Shooting** | 55850.439 | 32896.770 | 1.489 | 1.604 | 0.468 |
| **Sydney Siege** | 53221.181 | **71941.379** | 1.549 | 1.952 | 0.483 |

In general, the nature of breaking news and its diffusion patterns reduce the effect of using social-based features to distinguish rumors from non-rumors micro-posts for many reasons. First, breaking news mainly spreads on Twitter as trending stories and hashtags. Taking a glance at any trending breaking news hashtag clearly shows the high diversity in social-based features of the participants. Furthermore, predefined features are known to be data or domain dependent. Meaning that the effect of different types of features depends on the quality of the dataset and how informative

these features are in that specific dataset. For instance, many works in the literature on veracity classification and stance classification of long-standing rumors have experimented with social-based features as well as many other types of features and have reported contrasting results on different datasets. Finally, predefined lists of features need to be periodically revised and updated in order for the model to better handle new data. In the case of emerging breaking news rumors, even when a model is trained on high quality data where the social-based features are very informative, the model may not perform well with new data. This is a major advantage of our proposed model, which will learn the latent features and their correlations from the input text itself, rather than depending on a predefined list of features. Our design also allows the model to automatically learn new features from every new data it receives and dynamically update itself to better handle it.

### 5.3.6 Case Studies

In this section, two case studies of real-life breaking news events are first presented and followed by a brief discussion of the obtained results[7].

#### 5.3.6.1 Case Study 1: Detecting rumors of emerging sub-topics of a breaking news

To demonstrate the performance of our model on a real-time Twitter stream of a breaking news sub-topics, we collected tweets about an emerging breaking news story stating that the U.S. government lost track of almost 1,500 unaccompanied immigrant children after placing them in sponsors' homes[8]. This breaking news has recently become viral in Twitter with thousands of people wondering in the hashtag *#WhereAreTheChildren* about many aspects of the news. Although this news was verified in general, many tweets are spreading rumors about different aspects and details of the story. These rumors are not yet confirmed nor refuted by the government. We collected 50 tweets about this breaking news and manually fact-checked each of them and kept only the 34 tweets we

[7] Labeled data available at: http://dmas.lab.mcgill.ca/data/RumorsNonRumorsCaseStudyData.zip.

[8] Source: https://www.cnn.com/2018/05/26/politics/hhs-lost-track-1500-immigrant-children/index.html, retrieved on May 28, 2018

Table 5.9: The classification performance of our model on a real-life breaking news case study in terms of precision (p), recall (R), and F1

| | **P** | **R** | **F1** |
|---|---|---|---|
| **Rumor** | 0.786 | 0.647 | 0.710 |
| **Non-rumor** | 0.700 | 0.824 | 0.757 |
| **Both classes** | 0.743 | 0.735 | 0.757 |

Table 5.10: Examples of tweets collected from real-life breaking news and how it was classified by our model.

| **Tweet text** | **Truth** | **Classified** |
|---|---|---|
| So, about that prison bus for babies..., it actually takes charter school kids on field trips. | rumor | rumor |
| This administration is a real beauty. HOW in hades do you lose almost FIFTEEN HUNDRED CHILDREN? | non-rumor | non-rumor |
| How is it fake news? It's from their website and is literally a prison bus for babies. Why do you think the babies are there? | rumor | non-rumor |

know belong to one of the two classes: rumors[9] and non-rumors[10]. We then fed those tweets into our model to classify each of them as a rumor or not. Table 5.10 shows examples of the collected tweets and how they were classified by our model. Table 5.9 shows the classification performance of our rumor detection model when applied on these tweets in terms of precision, recall, and F1. These results suggest that our model is capable of detecting breaking news rumors of unseen topics with high accuracy.

#### 5.3.6.2 Case Study 2: Detecting rumors of multiple emerging breaking news topics

We performed another case study to demonstrate the performance of our model on detecting different emerging topics of multiple breaking news in a real-time Twitter stream. We started by collecting tweets about the following three unverified breaking news stories that have recently emerged and are not yet confirmed nor refuted by the government:

[9]Source: https://www.snopes.com/fact-check/prison-bus-for-babies/, retrieved on May 29, 2018

[10]Source: https://www.snopes.com/fact-check/1475-immigrant-children-missing/, retrieved on May 29, 2018

Table 5.11: The classification performance of our model on a real-life multiple breaking news case study in terms of precision (p), recall (R), and F1

| | **P** | **R** | **F1** |
|---|---|---|---|
| **Rumor** | 0.810 | 0.756 | 0.782 |
| **Non-rumor** | 0.766 | 0.818 | 0.791 |
| **Both classes** | 0.788 | 0.787 | 0.791 |

- *"449,000 California residents turned down jury duty because they are not U.S. citizens, despite being registered to vote"*[11]. This news spread very fast in social media and even more claims were added by users overtime. Nevertheless, this news is not verified yet.

- *"Guatemalan authorities rescued a group of minors from human smugglers in the migrant caravan"*[12]. This news is still unverified regardless of the claims about the existence of exclusive information and photos from a high-level Guatemalan government official.

- *"The U.S. Attorney for the Southern District of New York has begun the prosecution of President Trump's inauguration committee as of December 2018*[13]. Although this claim was published by reputable news organizations, it is still unverified and is based only on information from unnamed sources.

Furthermore, to demonstrate a real-life scenario where Twitter streams are not limited to predefined events or topics, we collected general streams of tweets from the following two major sources of breaking news:

- *An official Twitter account of a well-known news agency.* We collected all tweets in the first 2 pages of the timeline of the *CNN's* Twitter account[14]. These tweets represent a real-time stream of micro-posts about unspecified topics of regular as well as breaking news and events currently occurring all over the world.

- *A general all-time trending hashtag.* We also collected all tweets in the first 2 pages of

[11]Source: https://www.snopes.com/fact-check/did-449000-californians-turn-down-jury-duty-because-they-are-undocumented-immigrants/, retrieved on Dec 24, 2018
[12]Source: https://www.snopes.com/fact-check/guatemala-smugglers-children/, retrieved on Dec 25, 2018
[13]Source: https://www.snopes.com/fact-check/trump-entities-criminal-probe/, retrieved on Dec 20, 2018
[14]Source: https://twitter.com/CNN, retrieved on Dec 25, 2018

the timeline of a general widely-adopted fashion hashtag, namely *#OOTD*[15]. We choose this hashtag for two main reasons. First, fashion data in this hashtag represents unseen general topics that are not news-related. This simulates an everyday general real-time Twitter stream. Second, similar to a trending breaking news hashtag, trending fashion hashtags always contain tweets with many new and emerging topics, vocabulary, and named entities.

Next, we manually fact-checked each of the collected tweets and kept only the 89 ones we know belong to one of the two classes: rumors and non-rumors. We then randomly shuffled these tweets and fed them into our detection model. Table 5.11 shows the classification performance of our rumor detection model when applied on these tweets in terms of precision, recall, and F1. These results suggest that our model is capable of detecting multiple breaking news rumors of unseen topics in an everyday Twitter stream with high accuracy.

#### 5.3.6.3 Discussion of case studies results

To further understand the obtained results of our rumor detection model, we closely inspected the text of tweets that were correctly classified and compared it with tweets that were misclassified in the two case studies. We had two main observations. First, we noticed a high similarity in the writing styles among most rumor tweets. Similarly, most non-rumor tweets also have their own writing style. This observation can be further inspected in the future by proposing a breaking news rumor detection model that is conditioned on the different writing styles of tweets. Second, we noticed the existence of many new OOV terms and named entities that were not originally trained by our model such as *Inauguration*, *Guatemala*, *smugglers*, *Trump*, *immigrants*, and *outfit*. The results of the case studies suggest that our model can adaptively capture the drift and mitigate the OOV and topic-shift issues in breaking news rumor detection.

## 5.4 Limitation

According to our adopted definition where a rumor is defined as "a story or a statement whose truth value is unverified", rumors do not have to be false; they can be deemed later to be true or

[15]Source: https://twitter.com/search?vertical=default&q=%23OOTD&src=typd, retrieved on Dec 25, 2018

false. This definition implies that an emerging tweet that was flagged as rumor can later be non-rumor. However, our proposed model does not explicitly model or memorize the facts across time. To address this issue, the proposed model can be combined with a long-lasting rumor detection model. The proposed model is responsible for flagging and storing the emerging rumors, and the long-lasting rumor detection model can be trained when facts are checked.

However, our experiment and case studies show that although our model does not explicitly model and memorize facts across time, it performs fairly well by just looking at a tweet in the current moment. We suspect that there may be two reasons. First, the word2vec model is incrementally updated. It may memorize new concepts and drift over time. Secondly, the proposed model may memorize to distinguish how rumors and non-rumors are conveyed in natural language. They may correspond to a very different writing style, which coincides with our observations in case studies.

# Chapter 6

# Identifying High-engaging Breaking News Rumors in Social Media

The materials in this Chapter are currently under review in the 28th International Joint Conference on Artificial Intelligence (IJCAI 2019).

## 6.1 Introduction

Social media highly impacts people's knowledge and perception of the world. The convenience, speed, accessibility of real-time information, and the diversity of the available sources from every corner of the world have attracted more people to gather their news in social media every day [65]. According to a recent study from Pew Research Center, the global median for getting news from social media at least once a day has become 42% in 2018 [71]. The wide adoption of social media for news gathering comes with a drawback of overwhelming readers with lots of new information that cannot always be trusted. The lack of fact-checking and source-verification in social media facilitates the spread of huge volumes of rumors every day. These rumors, when become viral, may result in extremely damaging consequences in just a few minutes.

Allport and Postman [6] define a rumor as "a story or a statement whose truth value is unverified". According to this definition, a rumor does not have to be false; it can be deemed later to

be true or false. Rumors in social media can fall into one of two categories based on their temporal characteristics [108]: *long-standing rumors* that are well-discussed for long periods of time, and *breaking news rumors* that are generally unseen before and emerge extremely fast during the breaking news evolution.

Breaking news refers to "information that is being received and broadcast about an event that has just happened or just begun"[1]. Several characteristics of breaking news distinguish it from regular news, such as its dramatic evolution over time, the lack of sufficient details about what happened and what will happen, and the unexpected sequence of sub-topics that mostly do not occur in existing data. A typical example of breaking news is the *Thoku* earthquake of magnitude 9.0 that hit the east coast of Japan in 2011. This earthquake was followed by an abnormal sequence of events including a tsunami and the failure of three nuclear reactors in Fukushima. These severe consequences as well as the earthquake itself were outside of most people's expectations.

According to the basic law of rumors [6], the more the sensitivity, importance, and uncertainty of a topic, the more it is associated with rumors. This explains why breaking news is usually associated with many rumors, especially at the early stages of diffusion. Consequently, identifying and acting upon breaking news rumors in a timely fashion to minimize their harmful effect becomes an extremely difficult and crucial task. However, not all rumors have the potential to spread in social media. High-engaging breaking news rumors are those written in a manner that ensures they achieve the highest prevalence among the recipients. These rumors are extremely difficult to detect and have the potential to become extremely viral in social media for several reasons. First, the mental state of recipients during breaking news is one that is ready to accept any information without thinking or analyzing its contents [73]. This mental state reduces the recipients' ability to judge the quality of the information received. Furthermore, during breaking news and emergency situations, people closely follow up with any information update regarding the current development of the breaking news. Moreover, as a general rule, it is emotions that govern the sharing act in social media [13]. Rumors are intended to touch and satisfy the primal emotional needs of recipients, such as fear, anger, anxiety, sadness, or happiness [73]. More importantly, rumors are intentionally designed

[1] Source: https://dictionary.cambridge.org/dictionary/english/breaking-news, retrieved on Jan 3, 2019

and written to mimic how verified breaking news information is reported. Thus, in addition to the compelling writing, they are believable, expressive, informative, and answer questions people want to know [6, 24, 26].

Identifying high-engaging breaking news rumors in social media can be extremely helpful in prioritizing the rumors verification process during breaking news and emergencies to reduce their damaging consequences. However, the nature of breaking news and high-engaging rumors has posed many challenges to this task. First, breaking news covers topics and events that may not exist in the training dataset. The existing data may also lack similar, or related topics and events. In this case, the task of identifying high-engaging breaking news rumors requires zero-shot learning for real-time detection. This is much more challenging than handling regular news and long-standing rumors where the training dataset usually covers related events and topics as well as historical observations about the information in question. Second, information diffusion in social media during breaking news does not follow the regular flow of the network structure. Meaning that a breaking news rumor does not require the existence of explicit links (relations) between recipients to propagate throughout social media networks. In this case, the task of predicting the popularity of a rumor micro-post based on the structure of the social network or the patterns of information diffusion is not applicable. Also, handling breaking news rumors is a time-critical task. In such a case, even waiting for a few minutes for enough data to be available might render the results of a rumor detection and popularity prediction models useless in terms of minimizing the harmful effect of rumors during breaking news and emergencies. This is because, normally, more than 50% of the sharing act happens within the first ten minutes after posting the micro-post in social media [101]. However, during breaking news this percentage becomes much higher. Thus, after a few minutes, the damaging consequences of a high-engaging breaking news rumor is more likely to have already happened. Finally, the characteristics of high-engaging rumors are very much in line with high-engaging posts in social media in general. This makes the process of distinguishing high-engaging rumors from high-engaging non-rumors much more challenging than identifying rumors in general.

To address these challenges, we propose a multi-task deep learning model that can incrementally learn the shared latent features among the two tasks of *breaking news rumors detection* and *breaking news rumors popularity prediction* and use these features to train the model with the objective of

identifying high-engaging breaking news rumors in social media. In our design, we use a *word embedding learning* model to learn the distributed vector representations of terms in the input text. This helps our proposed model better handle the issues of emerging topics of breaking news such as *Out-Of-Vocabulary (OOV)* and topic-shifts. Furthermore, we use a *Convolutional Neural Network (CNN)* model and a *Self-attention mechanism* as shared feature extractors in the model. The CNN model helps capture several classes of semantic features while the attention mechanism guides the model to weight differently to the input sequence and locates important features for predicting the final class. This helps the proposed model learn the salient semantic similarities among important words and phrases for identifying high-engaging breaking news rumors in social media.

In this chapter, we tackle the problem of identifying breaking news rumors that are most likely to become viral and achieve high engagement rates in social media. The main contributions of this work can be summarized as follow:

- To the best of our knowledge, this is the first work that tackles the problem of identifying high-engaging breaking news rumors in social media.
- We propose a new multi-task CNN-attention-based neural network architecture to *jointly* learn the two tasks of breaking news rumors detection and breaking news rumors popularity prediction in social media. The proposed model learns the salient semantic similarities among important features for identifying high-engaging breaking news rumors and separates them from the rest of the input text.
- Extensive experiments on five real-life breaking news datasets suggest that our proposed model is capable of identifying high-engaging breaking news rumors in social media, and it outperforms all baselines in terms of precision, recall, and F1.

Most existing work on rumor detection focuses on long-standing rumors rather than breaking news rumors and aims at tracking the diffusion of rumors, classifying opinions expressed toward them, or predicting their veracity [108]. In contrast, this work aims at detecting high-engaging breaking news rumors, which is more challenging because of the lack of sufficient data and the need for addressing the OOV and topic shift issues. Furthermore, most existing studies assume that rumors are always false and propose models to detect these false rumors [108]. In these studies,

the detection models are trained on datasets of long-lasting rumors with the objective of detecting which of these rumors are false. According to the definition of rumors, they refer to unverified information that can be deemed later to be true or false. Thus, assuming that rumors are always false is invalid. In this work, we aim at detecting rumors regardless of their truth value. The goal is to flag high-engaging micro-post rumors during the rapid diffusion of breaking news to help reduce their damaging consequences.

Most existing work on popularity prediction aims at estimating the future popularity of a micro-post in social media based on the early observations of its dynamics, the network structure, or both. These methods are not applicable when dealing with breaking news rumors for several reasons. First, the spread of breaking news through social media does not necessarily follow the network structure. Second, waiting for enough early observations of the micro-post dynamics to be available is not an option. Also, in contrast to long-standing rumors, important breaking news rumors have a distinctive life-cycle. Studies have shown that a single verification tweet from an official governmental authority account will drastically increase the judgment ability of individuals and quickly curb the diffusion of a breaking news rumor [77, 90, 93]. Accordingly, the interest in posting and sharing rumors regarding the breaking news fades out quickly over time and the awareness of the associated verified rumors becomes very high among individuals. Thus, using early observation of a rumor's dynamics might not give a reliable estimation of its future popularity. In contrast, our proposed model does not need a collection of early observations nor is it based on the network structure.

The rest of the chapter is organized as follows. Section 6.2 provides an overview of the research problem and the details of the proposed model. Section 6.3 shows the experimental result with extensive discussions.

## 6.2 Joint Learning Model for Identifying High-engaging Breaking News Rumors

In this section, we first formally define the research problem followed by the proposed model.

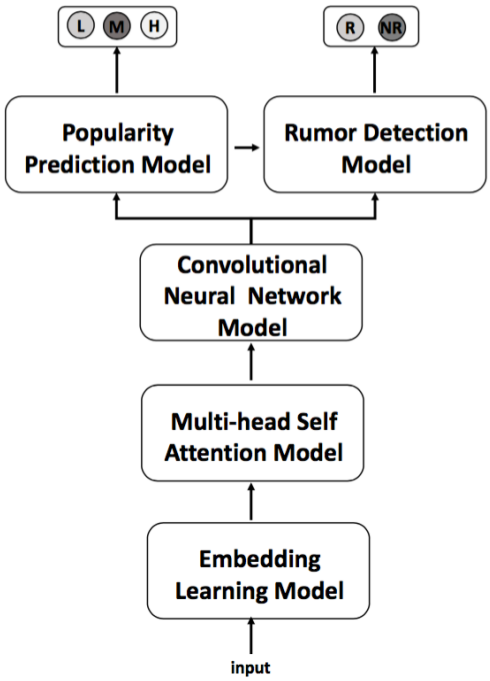


Figure 6.1: The proposed joint learning model for identifying high-engaging breaking news rumors in social media

### 6.2.1 Problem Statement

The research problem of identifying high-engaging breaking news rumors can be defined as follows. For a given micro-post regarding a specific piece of information, the task is to determine whether or not it is a rumor and predict its engagement rate among recipients in social media. Let $mp = \langle w_1, ..., w_T \rangle$ be a sequence of words in a micro-post $mp$ of length $T$, and let $F$ be the set of its associated features. Given $mp$ and $F$ as inputs, the goal is to simultaneously classify $mp$ as either a rumor or a non-rumor by assigning a label from $RL = \{R, NR\}$ and predicting the engagement rate it will achieve by assigning a label from $PL = \{High, Moderate, Low\}$.

### 6.2.2 Proposed Model

Figure 6.1 illustrates an overview of our proposed joint learning model. The proposed model consists of five main components, namely the embedding learning model, the multi-head self-attention model, the convolutional neural network model, the popularity prediction model, and the rumor detection model. The following subsections cover each component in details.

#### 6.2.2.1 Embedding Learning Model

This model takes a text corpus as an input and produces real-valued low-dimensional vector representations for words that appear in that corpus. These vector representations are called *word embeddings*.

In this work, we train the word2vec model using a technique called *skip-gram* [68]. Skip-gram takes a corpus of text as input and uses it to build a word2vec model as follows. Let the set of words surrounding a word $w_i$ within a specified window size in a sentence be the context of $w_i$. To build the word2vec model, skip-gram takes each word $w_i$ in the corpus along with the words representing its context and learns their word embeddings. The learning objective of skip-gram is to find useful representations of the input words in the embedding space so that, given any other word $w_t$, the word2vec model can predict the surrounding context words of $w_t$ with high probabilities and other words with low probabilities [68]. Formally, given a sequence of words $mp = \langle w_1, ..., w_T \rangle$ and a context window of size $\mathfrak{z}$, the objective of a skip-gram model is to maximize the following average log probability function:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-\mathfrak{z}\leq j\leq \mathfrak{z}, j\neq 0} \log p\left(w_{t+j}|w_t\right) \tag{29}$$

where $\log p\left(w_{t+j}|w_t\right)$ is approximated using negative sampling as follows:

$$\log p\left(w_{t+j}|w_t\right) = \log \sigma({\mathbf{v}'_{w_{t+j}}}^{\top}\mathbf{v}_{w_t}) + \sum_{i=1}^{\mathbf{k}} E_{w_i\sim P_n(w_{t+j})}\left[\log \sigma(-{\mathbf{v}'_{w_i}}^{\top}\mathbf{v}_{w_t})\right] \tag{30}$$

where $\mathbf{v}_{w_t}$ and $\mathbf{v}'_{w_{t+j}}$ denote the input and output vector representations of words $w_t$ and $w_{t+j}$, respectively; $\mathbf{k}$ denotes the number of negative samples for each data sample, and $P_n(w_{t+j})$ denotes the noise distribution [68].

By the end of this model, the sequence of words in each micro-post $mp$ is converted into a sequence of vectors $X = \langle x_1, \ldots, x_T \rangle$ that is passed to the subsequent multi-head self-attention model.

#### 6.2.2.2 Multi-head Self-attention Model

Attention mechanisms have been used mainly to guide a deep learning model to attend differently to the input sequence and locate important features to predict the final class. Recently, a *Self-attention* or *Scaled Dot-Product Attention* mechanism has been proposed as a part of a machine translator architecture called the *Transformer* [95].

*Scaled Dot-Product Attention* or *Self-attention* mechanism works as follows. First, it calculates the dot-product of a weight matrix $W_{attn} \in \mathbb{R}^{\mathcal{D} \times 3\mathcal{D}}$ by each word embedding $x_i$ in $X$. Next, it splits the result through dimension to generate three matrices of size $\mathcal{D}$ known as the query $\mathcal{Q}$, the key $\mathcal{K}$, and the value $\mathcal{V}$ matrices. Finally, the attention is calculated as follows:

$$Self - attention(\mathcal{Q}, \mathcal{K}, \mathcal{V}) = softmax(\frac{\mathcal{Q}\mathcal{K}^\top}{\sqrt{\mathcal{D}}})\mathcal{V} \tag{31}$$

In our work, instead of performing a single Self-attention function with $\mathcal{D}$-dimensional keys, values, and queries, we use a *Multi-head Self-attention* mechanism to allow the learning model to "jointly attend to information from different representation subspaces at different positions" [95]. *Multi-head Self-attention* mechanism works as follows. First, the queries, keys, and values are linearly projected $\mathcal{H}$ times with different learned linear projections to $\mathcal{D}_\mathcal{K}$, $\mathcal{D}_\mathcal{K}$, and $\mathcal{D}_\mathcal{V}$ dimensions, respectively [95]. Next, the self-attention value for each projected version, i.e., head, is calculated as follows:

$$head_i = Self - attention(\mathcal{Q}\mathcal{P}_i^\mathcal{Q}, \mathcal{K}\mathcal{P}_i^\mathcal{K}, \mathcal{V}\mathcal{P}_i^\mathcal{V}) \tag{32}$$

where $\mathcal{P}_i^\mathcal{Q} \in \mathbb{R}^{\mathcal{D} \times \mathcal{D}_\mathcal{K}}$, $\mathcal{P}_i^\mathcal{K} \in \mathbb{R}^{\mathcal{D} \times \mathcal{D}_\mathcal{K}}$ and $\mathcal{P}_i^\mathcal{V} \in \mathbb{R}^{\mathcal{D} \times \mathcal{D}_\mathcal{V}}$ are the projection parameters. Finally, these attentions are concatenated and projected again to compute the final multi-head attention as follows:

$$Multi - headAttention(\mathcal{Q}, \mathcal{K}, \mathcal{V}) = concat(head_1, \ldots, head_\mathcal{H})\mathcal{P}^O \tag{33}$$

where $\mathcal{P}^O \in \mathbb{R}^{\mathcal{H}\mathcal{D}_\mathcal{V} \times \mathcal{D}}$ is the projection parameter.

By the end of this model, the sequence of word embedding vectors $X = \langle x_1, \ldots, x_T \rangle$ of each micro-post is converted into a sequence of attended-vectors $X^{att} = \langle x_1^{att}, \ldots, x_T^{att} \rangle$. This sequence of attended-vectors $X^{att}$ is then transformed into a matrix representation $\mathcal{A} \in \mathbb{R}^{T * \mathcal{D}}$ that is passed

to the subsequent convolutional neural network model.

#### 6.2.2.3 Convolutional Neural Network Model

*Convolutional Neural Networks (CNNs)* are feed forward neural networks that typically consist of convolutional layers followed by pooling layers. According to Goldberg and Hirst [34], a CNN is basically a feature extractor model that is useful only as a substructure of larger networks. These feature extractor models have been used extensively in the literature for image-related tasks. However, recent studies have shown promising results of using CNN models for text classification [49, 102].

In our work, a CNN model is used to extract the latent features from the input textual data as follows. First, the convolutional layers apply convolutional filters over the input matrix to produce different feature maps. Then, these feature maps are fed through pooling layers to induce a fixed length features vector of the micro-post. Formally, given an input matrix $\mathcal{A} \in \mathbb{R}^{T*\mathcal{D}}$ that represents a micro-post consisting of $T$ words, each represented by a $\mathcal{D}$-dimensional vector of real values, the convolutional layer will repeatedly apply the linear filters on sub-matrices of $\mathcal{A}$ as follows [102]:

$$\mathcal{O}^i_{cnv} = \mathbf{W} \cdot \mathcal{A}\left[i : i + r - 1\right] \tag{34}$$

where $\mathcal{O}^i_{cnv}$ is the output of the convolutional operator after applying the $i^{th}$ filter, $r$ is the region size or the height of the filter, $\mathbf{W}$ is the weight matrix of the filter, and $\mathcal{A}\left[i : i + r - 1\right]$ represents the sub-matrix of $\mathcal{A}$ from row $i$ to row $i + r - 1$. A feature map $c_i$ of the $i^{th}$ filter is then calculated as follows:

$$c_i = \mathbf{a}\left(\mathcal{O}^i_{cnv} + \mathbf{b}\right) \tag{35}$$

where $\mathbf{a}$ is the activation function and $\mathbf{b} \in \mathbb{R}$ is a bias term.

These feature maps $C$ are then fed into a 1-max pooling layer that extracts a scalar value and generates a univariate feature vector from each feature map $c_i$. The univariate feature vectors are then concatenated to form a fixed-size feature vector $FV$ that is passed to the popularity prediction and rumor detection models for the final prediction task.

#### 6.2.2.4 Popularity Prediction Model

This model learns the future popularity of a micro-post in social media as a function of its *Engagement Rate*, which is a widely-adopted measure to evaluate the quality of a micro-post on different social media platforms. It is generally calculated as the number of received engagements of a micro-post divided by the number of users or events that triggered the engaging action. Formally, let the *Engagement Volume* denote the total impact of a micro-post $mp$, which is calculated as the total count of *likes*, *shares*, and *comments* received by $mp$; let the *Base Volume* denote the number of users with direct exposure to the content of $mp$. The engagement rate of $mp$ is then calculated as follows:

$$Engagement\ Rate(mp) = \frac{Engagement\ Volume(mp)}{Base\ Volume(mp)}, \tag{36}$$

In our work, due to the difficulty of performing classification or regression for the whole engagement rate range, we reformulate the popularity prediction task as a multi-classification task that assigns each micro-post to one of the three levels of popularity as follows:

- *Low popularity* if the engagement rate of a micro-post is below $0.02\%$.
- *Moderate popularity* if the engagement rate of a micro-post is between $0.02\%$ and $0.33\%$.
- *High popularity* if the engagement rate of a micro-post is above $0.33\%$.

This model learns the popularity prediction task as follows. First, it takes the feature vector $FV$ generated by the convolutional neural network model and passes it through a fully connected layer to generate an internal feature vector of the popularity prediction model, namely $FV_p$. Then, $FV_p$ is passed through a softmax layer with *sigmoid* function to predict the popularity level of a micro-post. The objective function is optimized by minimizing the following cross-entropy popularity prediction loss function $L_P$:

$$L_P = -\frac{1}{\mathtt{N}} \sum_{i=1}^{\mathtt{N}} \sum_{j=1}^{\mathtt{M}_p} \left[ \left( \mathtt{y}_p^i = j \right) F_p \left( \mathtt{x}^i, \Delta \right) \right] \tag{37}$$

where $\mathtt{N}$, $\mathtt{M}_p$, $\mathtt{x}^i$, and $\mathtt{y}_p^i$ represent the number of training samples, number of classes, the $i^{th}$ training example, and its actual class, respectively. $\Delta$ denotes the model parameters, and $F_p\left(\mathtt{x}^i, \Delta\right)$ represents the predicted popularity class.

#### 6.2.2.5 Rumor Detection Model

This model takes two inputs: the feature vector $FV$ generated by the convolutional neural network model and the feature vector $FV_p$ generated by the popularity prediction model. Then, it learns the rumor detection task as follows. First, the feature vector $FV$ is fed through fully connected layers and the output is concatenated with the $FV_p$ feature vector. The merged vector is then fed through a softmax layer with *tanh* function to predict whether or not the input micro-post is a rumor. The objective function here is to minimize the cross-entropy rumor detection loss function $L_R$ as follows:

$$L_R = -\frac{1}{\mathtt{N}} \sum_{i=1}^{\mathtt{N}} \sum_{j=1}^{\mathtt{M}_R} \left[\left(\mathtt{y}_R^i = j\right) F_R\left(\mathtt{x}^i, \Omega\right)\right] \tag{38}$$

where $\mathtt{N}$, $\mathtt{M}_R$, $\mathtt{x}^i$, and $\mathtt{y}_R^i$ represent the number of training samples, number of classes, the $i^{th}$ training example, and its actual class, respectively. $\Omega$ denotes the model parameters and $F_R\left(\mathtt{x}^i, \Omega\right)$ represents the predicted rumor class.

Finally, the joint loss for the entire joint learning model is calculated as the following unified loss:

$$L_{uni} = \lambda L_P + L_R \tag{39}$$

where $\lambda$ is a weighting factor to be learned.

## 6.3 Experiments

The objectives of the experiments are to evaluate the classification performance of the proposed joint learning model and to evaluate the effect of joint learning on the single task of breaking news rumors detection and the task of breaking news rumors popularity prediction. Below, we first describe the datasets, features, and experimental settings, followed by the results and discussions.

### 6.3.1 Dataset

In our experiments, we used five sets of real-life, publicly accessible tweets from *PHEME* [106], where each set is related to a different breaking news story and contains both rumors and non-rumors tweets as shown in Table 6.1.

Table 6.1: Percentages of rumors and non-rumors tweets in the PHEME datasets

| **Breaking News** | **Rumors** | **Non-rumors** |
|---|---|---|
| **Charlie Hebdo** | 458 (22.0%) | 1,621 (78.0%) |
| **Ferguson** | 284 (24.8%) | 859 (75.2%) |
| **Germanwings Crash** | 238 (50.7%) | 231 (49.3%) |
| **Ottawa Shooting** | 470 (52.8%) | 420 (47.2%) |
| **Sydney Siege** | 522 (42.8%) | 699 (57.2%) |

### 6.3.2 Feature Sets

To evaluate the effect of using different features on the classification performance of our proposed model as well as the baseline models, we experimented with the following feature sets as our input:

- **Text.** In this case, the input of the model is simply the text of the micro-posts with no additional predefined features.

- **Text and stylometric features.** In this case, the model takes the text as well as the stylometric features of the micro-posts as input. Table 6.2 shows the set of stylometric features used in this work.

- **Text and emotional triggers features.** In this case, the model takes the text as well as the emotional triggers of the micro-posts as input. For the emotional trigger words, we employed the *NRC Word-Emotion Association Lexicon (EmoLex)* [72]. This lexicon covers words that are associated with the eight primal emotions: *anger, fear, disgust, sadness, anticipation, surprise, joy, and trust*. It also covers words associated with *positive* as well as *negative* sentiments. Furthermore, we leverage from the *emoticons* associated with these primal emotions

Table 6.2: List of the stylometric features

| Feature | Description |
|---|---|
| **Capital ratio** | ratio of capital letters in the text. |
| **#QuestionMarks** | number of question marks (?) in the text. |
| **#ExclamationMarks** | number of exclamation marks (!) in the text. |
| **#Periods** | number of periods (.) in the text. |
| **#DoubleQuotes** | number of double quotation marks (") in the text. |
| **#SingleQuotes** | number of single quotation marks (') in the text. |
| **#Words** | number of words in the text. |
| **#URLs** | number of URLs (http://) in the text. |
| **#Hashtags** | number of hashtags (#) in the text. |
| **#Mentions** | number of mentions (@) in the text. |
| **#Emojis** | number of emoicons in the text. |
| **#Commas** | number of commas (,) in the text. |
| **#AndMarks** | number of ampersands (&) in the text. |
| **#SemicolonMarks** | number of semicolons (;) in the text. |
| **#ColonMarks** | number of colons (:) in the text. |

in social media. An emoticon refers to "an image made up of symbols such as punctuation marks, used in text messages, emails, etc. to express a particular emotion"[2]. The list of emotional triggers emoticons we adopt in this work is summarized in Table 6.3.

Table 6.3: The list of primal emotions and the associated emotional triggers emoticons used in social media

| Emotion | Emotional triggers emoticons |
|---|---|
| **Anger** | >:S , >:{ , >: , >:[ , >:— , x-@ , :@ , :-@ , :-/ , :-\ , :/ , :\ |
| **Disgust** | :&, , :-& |
| **Fear** | :-o , :-O , :o , :O , :-$ , :$ , o_O , O_o |
| **Joy** | :-) , :) , ;-) , ;) , :-)) , :-D , :D , ;-D , ;D , :-p , :p , ;-p<br>:^) , ;^) , :o) , ;o) , :') , :-] , :] , ;-] , ;] , :-> , :> , :∼)<br>;p , =-D , =D |
| **Sadness** | :-( , :( , :-(( , =( , :-[ , :[ , :-< , :< , :∼( , :^( , :o( , :'( |
| **Surprise** | :-o , :-O , :o , :O , :-$ , :$ , o_O , O_o |

- **Text, stylometric, and emotional triggers features.** In this case, the model takes the text, stylometric, and the emotional triggers features of the micro-posts as input.

[2]Source: https://dictionary.cambridge.org/dictionary/english/emoticon, retrieved on Jan 7, 2019

### 6.3.3 Experimental Settings

To simulate a real-life breaking news scenario, we performed a 5-fold cross-validation as follows. In each run, we used the datasets of four breaking news stories as our training data. Then, we used the fifth dataset to evaluate the classification performance of the proposed model and the baseline models in terms of precision, recall, and F1. By designing our experiment this way, we insure that the dataset used for the evaluation in each of the five runs represents breaking news rumors of unseen topics. Furthermore, to insure the stability of the reported results of the deep learning models and get a more robust estimation of their classification performances, we did five repetitions of each run of the 5-fold cross-validation for each model. Then, we reported their classification performance as the $mean \pm variance$ of the precision, recall, and F1 scores across the five repetitions of the 5-fold cross-validation instead of a single 5-fold cross-validation run.

The proposed model was implemented using *JetBrains PyCharm*[3] development environment for Python and the *TensorFlow*[4] open source machine learning framework . We ran our experiments on a *Windows server 2016 Datacenter*. The machine has a 32 GB of RAM and is powered by an Intel Xeon E5-1650 v4 processor at 3.60 GHz.

### 6.3.4 Experimental Results

#### 6.3.4.1 Classification Performance of the Proposed Joint Learning Model

To the best of our knowledge, this is the first work that tackles the problem of identifying high-engaging breaking news rumors in social media. To evaluate the classification performance of our proposed joint learning model, we compared the following variations of the proposed model:

- **Multi-task CNN-based joint learning model.** We implemented a multi-task joint learning model with only the convolutional neural network model as the feature extractor. We then trained the model to simultaneously learn the two tasks of breaking news rumor detection and breaking news popularity prediction.

- **Multi-task CNN-Attn-based joint learning model.** We implemented a multi-task joint

[3]Source: https://www.jetbrains.com/pycharm/, retrieved on December 28, 2018
[4]Source: https://www.tensorflow.org/, retrieved on December 28, 2018

learning model with both self-attention and convolutional neural network models as the feature extractors. We then trained the model to simultaneously learn the two tasks of breaking news rumor detection and breaking news popularity prediction.

In this experiment, we used different feature sets as inputs to each model and performed 5 repetitions of 5-fold cross-validations. Then we reported the classification performance results of each model as the $mean \pm variance$ of precision, recall, and F1.

Table 6.4 shows the obtained results. Bold values indicate the best classification performance among all models. As shown in the table, the *Multi-task CNN-Attn-based* model along with the *Text and Emotional* features outperformed all other models in identifying high-engaging breaking news rumors in terms of precision and F1, while it outperformed all other models in terms of recall when the *Text and Stylometric* features are used as its input.

Table 6.4: Mean $\pm$ variance of the precision (P), recall (R), and F1 scores of identifying high-engaging breaking news rumors using different features sets and variations of our proposed joint learning model

| **Model** | **Features** | **P** | **R** | **F1** |
|---|---|---|---|---|
| **Multi-task CNN-based model** | Text | 0.711 $\pm$0.002 | 0.775 $\pm$0.007 | 0.742 $\pm$0.005 |
| | Text & Stylometric | 0.712 $\pm$0.002 | 0.784 $\pm$0.001 | 0.746 $\pm$0.001 |
| | Text & Emotional | 0.725 $\pm$0.000 | 0.772 $\pm$0.002 | 0.748 $\pm$0.001 |
| | Text & Stylometric & Emotional | 0.710 $\pm$0.002 | 0.790 $\pm$0.010 | 0.748 $\pm$0.006 |
| **Multi-task CNN-Attn-based model** | Text | 0.712 $\pm$0.001 | 0.794 $\pm$0.006 | 0.753 $\pm$0.002 |
| | Text & Stylometric | 0.716 $\pm$0.001 | **0.797** $\pm$0.001 | 0.756 $\pm$0.001 |
| | Text & Emotional | **0.731** $\pm$0.000 | 0.791 $\pm$0.001 | **0.761** $\pm$0.000 |
| | Text & Stylometric & Emotional | 0.721 $\pm$0.001 | 0.792 $\pm$0.001 | 0.755 $\pm$0.001 |

The results also show that, for each features set, our proposed *Multi-task CNN-Attn-based* model yielded a better classification performance than the *Multi-task CNN-based* model. This suggests that using both the convolutional neural network model and the self-attention model as shared feature

extractors helps the proposed model to better capture the salient features and the semantic similarities among important words and phrases in the input text for the task of identifying high-engaging breaking news rumors in social media.

We further evaluated the classification performance of the two variations of the proposed joint learning model: the *Multi-task CNN-based* model and the *Multi-task CNN-Attn-based* model using the *Area Under Curve - Receiver Operating Characteristic (AUC-ROC)* curves. AUC-ROC curves are used to evaluate the classification performance of classification models at various threshold settings. The *Receiver Operating Characteristic (ROC)* curve is plotted in a 2-dimensional space where the x-axis represents the *False Positive Rate (FPR)* and the y-axis represents the *True Positive Rate (TPR)*. The *Area Under Curve (AUC)* score measures the ability of the classification model to separate or distinguish between the different classes. The higher the AUC score, the better the classification performance of the model is.

In this experiment, we performed a 5-fold cross-validation and plotted the ROC curve for each run as well as the mean ROC curve across all five runs for the two models. We also calculated the AUC score for each run as well as the Mean $\pm$ variance of the AUC scores across all five runs for each model. Figure 6.2 shows the obtained results. As shown in Figure 6.2.a, the mean AUC score is $0.63 \pm 0.14$ for the *Multi-task CNN-based* model while the mean AUC score is $0.69 \pm 0.08$ for the *Multi-task CNN-Attn-based* model as shown in Figure 6.2.b. These results shows that our proposed *Multi-task CNN-Attn-based* model significantly outperformed the baseline *Multi-task CNN-based* model in identifying high-engaging breaking news rumors in terms of the AUC-ROC score. This suggests that the design of our proposed joint learning model helps it better capture the features for the task of identifying high-engaging breaking news rumors in social media.

#### 6.3.4.2 Joint Learning Effect on the Single Tasks of Breaking News Rumors Detection and Breaking News Rumors Popularity Prediction

To evaluate the effect of the joint learning on the classification performance of the single tasks of breaking news rumors detection as well as breaking news rumors popularity prediction, we compared our joint learning model with the following single-task baseline classifiers:

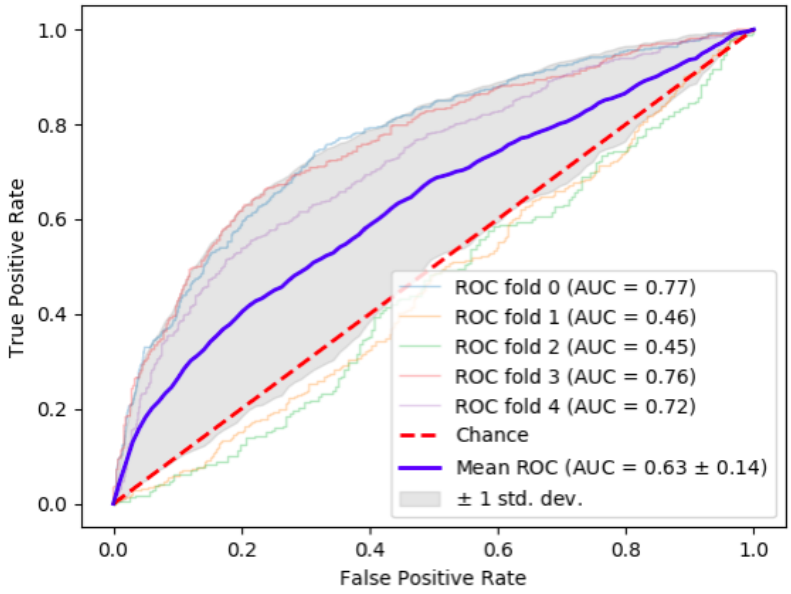


Figure 6.2.a: AUC-ROC for the Multi-task CNN-based model

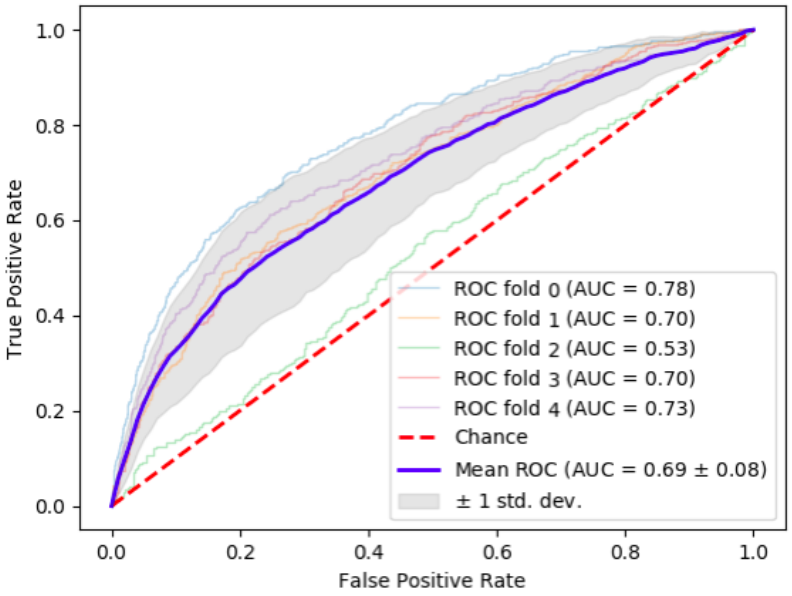


Figure 6.2.b: AUC-ROC for the Multi-task CNN-Attn-based model

Figure 6.2: Receiver Operating Characteristic (ROC) curves for the two variations of the proposed joint learning model showing the ROC curves and the Area Under Curve (AUC) scores for each of the five runs and the Mean $\pm$ variance of the AUC scores across all five runs. Figure 6.2.a shows the obtained results for the Multi-task CNN-based model and Figure 6.2.b shows the obtained results for the Multi-task CNN-Attn-based model

- **Naive Bayes (NB), Support Vector Machine (SVM), and Random Forest (RF) classifiers.** We implemented a NB classifier, a SVM classifier, and a RF classifier using the *R* language [80] along with the *RStudio* development environment[5]. Then, we used a pretrained word2vec model to replace the text of each micro-post with its *word embeddings* representation. The word embeddings representation of each micro-post is then used as input instead of the text of the micro-post.

- **CNN-based model.** We implemented a single-task model with a convolutional neural network model as the feature extractor.

- **CNN-Attn-based model.** We implemented a single-task model with self-attention and convolutional neural network models as feature extractors.

In this experiment, we used different features sets as our inputs. For each feature set, we trained two models of each baseline single-task classifier. One to learn the task of breaking news rumors detection and another to learn the task of breaking news rumors popularity prediction. To evaluate the classification performance of the NB, SVM, and RF classifiers, we performed a 5-fold cross-validation and reported the results of each model on each task in terms of precision, recall, and F1. To evaluate the classification performance of the deep learning models, we performed 5 repetitions of 5-fold cross-validations and reported the classification performance results of each model on each task as the $mean \pm variance$ of precision, recall, and F1. Table 6.5 shows the obtained results. Bold values indicate the best classification performance among all models.

By looking at the table one can see that, in general, the deep learning models yielded better classification performance than the BN, SVM, and the RF classifiers. This suggests that incrementally learning the word embeddings and the latent feature from the input text helps the deep learning models effectively capture the drift in topics and learn the important features for the two single tasks of breaking news rumors detection and breaking news popularity prediction. This is especially true with the breaking news rumors popularity prediction task where the deep learning models yielded significantly better classification performance than the BN, SVM, and the RF classifiers. Thus, it helps the popularity prediction models overcome the need for collecting sufficient data about the

[5]Source: https://www.rstudio.com/, retrieved on January 12, 2019

early observations, the network structure, or the related topics to predict the future popularity of micro-posts in social media.

The following subsections discuss the effect of the joint learning on the classification performance of each task.

#### 6.3.4.2.1 Joint learning effect on breaking news rumors detection task

The results in Table 6.5 show that the *Multi-task CNN-Attn-based* model along with the *Text and Emotional* features had the best classification performance in terms of precision and F1, while the *CNN-Attn-based* model along with the *Text, Stylometric, and Emotional* features yielded the best classification performance in terms of recall.

These results suggest that jointly learning the two tasks of breaking news rumors detection and breaking news rumors popularity prediction by our proposed model had significantly improved the classification performance of the breaking news rumors detection in terms of precision and F1 over all the single-task classifiers in our experiment. This shows how the design of our joint learning model helps leverage from the shared characteristics between the two tasks in improving the breaking news rumors detection task.

#### 6.3.4.2.2 Joint learning effect on breaking news rumors popularity prediction task

The results in Table 6.5 show that the *CNN-based* model along with the *Text, Stylometric, and Emotional* features had the best performance in terms of precision, while the *CNN-Attn-based* model along with the *Text, Stylometric, and Emotional* features yielded the best classification performance in terms of recall and F1.

These results suggest that the joint learning did not improve the task of breaking news rumors popularity prediction. In fact, single-task deep learning models yielded the best overall classification performance. Nevertheless, our proposed joint learning model still achieved a high classification performance in terms of precision and recall and a comparable classification performance in terms of F1 to all the single-task deep learning classifiers in our experiment. This shows that our joint learning model is capable of learning important latent features for predicting the popularity of breaking news rumors in social media with high accuracy without the need for gathering the early

observations of its dynamics or the need for a collection of related or similar topics and posts.

#### 6.3.4.3 Discussion on Feature Sets

In this section, we discuss the effect of including the *emotional triggers* and the *stylometric* features, in addition to the text, on the classification performance of breaking news rumors detection and breaking news popularity prediction. We start by inspecting the results in Table 6.5 to determine, for each model, which feature sets yielded the best classification performance in terms of precision, recall, and F1. Table 6.6 shows the summarized results for the breaking news rumors detection task and Table 6.7 shows the summarized results for the breaking news rumors popularity prediction task. We observed that using the *Text, Stylometric, and Emotional* features sets yielded the best classification performance in terms of precision of the rumors detection task and in terms of recall and F1 of the rumors popularity prediction task in all models except the *Multi-task CNN-Attn-based model*. It also yielded the best classification performance in terms of precision of the rumors detection task for three out of the five single-task models and the best classification performance in terms of recall and F1 of the rumors popularity prediction task for four out of the five single-task models. We also observed that, for the proposed joint learning model, using the *Text and Emotional* features sets yielded the best classification performance in terms of precision and F1 of the rumors detection task and in terms of precision, recall, and F1 for the rumors popularity prediction task.

These observations suggest the following. First, using the *emotional triggers* as well as the *stylometric* features can effectively help a classification model better learn the tasks of breaking news rumors detection and the breaking news rumors popularity prediction in social media. Second, in most of the cases, the best classification performances of the two tasks were achieved using the same sets of features. Hence, the high accuracy of jointly learning the two tasks. Finally, although incrementally learning the word embeddings of the input text helps mitigate the topic-shift and OOV issues of breaking news rumors detection, including the emotional triggers features and the stylometric features helps improve the task of identifying high-engaging breaking news rumors in social media.

Table 6.5: Precision (P), recall (R), and F1 scores of the two tasks of breaking news rumors detection and breaking news rumors popularity prediction across all five runs for the single-task baseline classifiers and our proposed joint learning models using different features sets

| **Model** | **Features** | **Breaking News Rumors Detection** | | | **Breaking News Rumors Popularity Prediction** | | |
|---|---|---|---|---|---|---|---|
| | | **P** | **R** | **F1** | **P** | **R** | **F1** |
| **Naive Bayes** | Text | 0.512 | 0.523 | 0.517 | 0.222 | 0.335 | 0.267 |
| | Text & Stylometric | 0.536 | 0.527 | 0.531 | 0.315 | 0.347 | 0.330 |
| | Text & Emotional | 0.530 | 0.529 | 0.529 | 0.299 | 0.300 | 0.300 |
| | Text, Stylometric & Emotional | 0.546 | 0.536 | 0.541 | 0.319 | 0.360 | 0.338 |
| **Support Vector Machine** | Text | 0.307 | 0.500 | 0.380 | 0.224 | 0.333 | 0.268 |
| | Text & Stylometric | 0.494 | 0.533 | 0.513 | 0.347 | 0.346 | 0.347 |
| | Text & Emotional | 0.437 | 0.534 | 0.481 | 0.257 | 0.334 | 0.291 |
| | Text, Stylometric & Emotional | 0.502 | 0.534 | 0.518 | 0.361 | 0.357 | 0.359 |
| **Random Forest** | Text | 0.320 | 0.500 | 0.390 | 0.224 | 0.333 | 0.268 |
| | Text & Stylometric | 0.321 | 0.499 | 0.391 | 0.234 | 0.338 | 0.277 |
| | Text & Emotional | 0.340 | 0.398 | 0.367 | 0.257 | 0.321 | 0.285 |
| | Text, Stylometric & Emotional | 0.332 | 0.504 | 0.400 | 0.307 | 0.379 | 0.339 |
| **CNN-based model** | Text | 0.514 $\pm$0.000 | 0.554 $\pm$0.016 | 0.533 $\pm$0.009 | 0.811 $\pm$0.000002 | 0.959 $\pm$0.0003 | 0.879 $\pm$0.00004 |
| | Text & Stylometric | 0.519 $\pm$0.000 | 0.559 $\pm$0.004 | 0.538 $\pm$0.001 | 0.810 $\pm$0.00001 | 0.988 $\pm$0.00007 | 0.890 $\pm$0.000003 |
| | Text & Emotional | 0.532 $\pm$0.000 | 0.564 $\pm$0.004 | 0.548 $\pm$0.0009 | 0.834 $\pm$0.00004 | 0.886 $\pm$0.003 | 0.859 $\pm$0.0006 |
| | Text, Stylometric & Emotional | 0.551 $\pm$0.000 | 0.599 $\pm$0.001 | 0.574 $\pm$0.0002 | **0.841** $\pm$0.00003 | 0.943 $\pm$0.0002 | 0.889 $\pm$0.00002 |
| **CNN-Attn-based model** | Text | 0.514 $\pm$0.000 | 0.618 $\pm$0.052 | 0.561 $\pm$0.045 | 0.805 $\pm$0.00001 | 0.963 $\pm$0.007 | 0.878 $\pm$0.001 |
| | Text & Stylometric | 0.523 $\pm$0.00001 | 0.635 $\pm$0.001 | 0.574 $\pm$0.0002 | 0.805 $\pm$0.000 | 0.999 $\pm$0.000 | 0.890 $\pm$0.000 |
| | Text & Emotional | 0.519 $\pm$0.00002 | 0.666 $\pm$0.011 | 0.583 $\pm$0.0009 | 0.794 $\pm$0.0007 | 0.999 $\pm$0.000003 | 0.885 $\pm$0.0003 |
| | Text, Stylometric & Emotional | 0.518 $\pm$0.00005 | **0.728** $\pm$0.029 | 0.605 $\pm$0.003 | 0.806 $\pm$0.000 | **1.000** $\pm$0.000 | **0.892** $\pm$0.000 |
| **Multi-task CNN-based model** | Text | 0.616 $\pm$0.003 | 0.576 $\pm$0.014 | 0.595 $\pm$0.003 | 0.811 $\pm$0.00003 | 0.974 $\pm$0.001 | 0.885 $\pm$0.0001 |
| | Text & Stylometric | 0.613 $\pm$0.003 | 0.589 $\pm$0.002 | 0.601 $\pm$0.002 | 0.811 $\pm$0.00003 | 0.978 $\pm$0.0005 | 0.887 $\pm$0.00003 |
| | Text & Emotional | 0.638 $\pm$0.001 | 0.565 $\pm$0.003 | 0.599 $\pm$0.0004 | 0.812 $\pm$0.00002 | 0.979 $\pm$0.0005 | 0.888 $\pm$0.0001 |
| | Text, Stylometric & Emotional | 0.605 $\pm$0.004 | 0.618 $\pm$0.020 | 0.612 $\pm$0.002 | 0.815 $\pm$0.00003 | 0.973 $\pm$0.0003 | 0.887 $\pm$0.00003 |
| **Multi-task CNN-Attn-based model** | Text | 0.613 $\pm$0.003 | 0.609 $\pm$0.011 | 0.611 $\pm$0.0003 | 0.811 $\pm$0.0002 | 0.979 $\pm$0.0002 | 0.887 $\pm$0.00004 |
| | Text & Stylometric | 0.621 $\pm$0.002 | 0.613 $\pm$0.002 | 0.617 $\pm$0.0004 | 0.811 $\pm$0.00003 | 0.981 $\pm$0.001 | 0.888 $\pm$0.00004 |
| | Text & Emotional | **0.647** $\pm$0.0002 | 0.600 $\pm$0.001 | **0.623** $\pm$0.001 | 0.814 $\pm$0.00001 | 0.982 $\pm$0.001 | 0.890 $\pm$0.0001 |
| | Text, Stylometric & Emotional | 0.630 $\pm$0.002 | 0.600 $\pm$0.001 | 0.615 $\pm$0.0001 | 0.813 $\pm$0.0005 | 0.978 $\pm$0.0005 | 0.888 $\pm$0.0003 |

Table 6.6: Best performing feature sets with each model for the single task of breaking news rumors detection in terms of precision (P), recall (R), and F1

| Model | P | R | F1 |
|---|---|---|---|
| **Naive Bayes** | Text, Stylometric, & Emotional | Text, Stylometric, & Emotional | Text, Stylometric, & Emotional |
| **Support Vector Machine** | Text, Stylometric, & Emotional | Text, Stylometric, & Emotional | Text, Stylometric, & Emotional |
| **Random Forest** | Text & Emotional | Text, Stylometric, & Emotional | Text, Stylometric, & Emotional |
| **CNN-based model** | Text, Stylometric, & Emotional | Text, Stylometric, & Emotional | Text, Stylometric, & Emotional |
| **CNN-Attn-based model** | Text & Stylometric | Text, Stylometric, & Emotional | Text, Stylometric, & Emotional |
| **Multi-task CNN-based model** | Text & Emotional | Text, Stylometric, & Emotional | Text, Stylometric, & Emotional |
| **Multi-task CNN-Attn-based model** | Text & Emotional | Text & Stylometric | Text & Emotional |

Table 6.7: Best performing feature sets with each model for the single task of breaking news rumors popularity prediction in terms of precision (P), recall (R), and F1

| Model | P | R | F1 |
|---|---|---|---|
| **Naive Bayes** | Text, Stylometric, & Emotional | Text, Stylometric, & Emotional | Text, Stylometric, & Emotional |
| **Support Vector Machine** | Text, Stylometric, & Emotional | Text, Stylometric, & Emotional | Text, Stylometric, & Emotional |
| **Random Forest** | Text, Stylometric, & Emotional | Text, Stylometric, & Emotional | Text, Stylometric, & Emotional |
| **CNN-based model** | Text, Stylometric, & Emotional | Text & Stylometric | Text & Stylometric |
| **CNN-Attn-based model** | Text, Stylometric, & Emotional | Text, Stylometric, & Emotional | Text, Stylometric, & Emotional |
| **Multi-task CNN-based model** | Text, Stylometric, & Emotional | Text & Emotional | Text & Emotional |
| **Multi-task CNN-attn-based model** | Text & Emotional | Text & Emotional | Text & Emotional |

# Chapter 7

# Conclusions

With the explosive growth of the Internet and the huge number of existing social media websites and applications, nearly half the world's inhabitants are on social media nowadays. This number is dramatically increasing over time. Consequently, the exponential growth of data is impacting people's knowledge and perception of the world. Textual data in social media contains valuable real-time information from every corner of the globe. However, overwhelming users with such volumes of unstructured data complicates the task of extracting useful information and assessing its veracity. In this thesis, we present three research problems to address major challenges of handling textual data in social media.

First, overwhelming the user with big volumes of short, noisy, and unstructured textual data complicates the task of understanding what topics are discussed at a given time and extracting useful information from the data. This thesis tackles the problem of improving topic modeling of short text documents in social media by proposing a new method that incorporates the Twitter-LDA topic model, WordNet, and the set of hashtags available in the corpus of micro-posts. The objective is to improve the top probable keywords that represent each topic. Based on the semantic relationships in WordNet and the set of hashtags available in the corpus, the importance of different keywords to different topics is emphasized in the effort of providing the user with a higher quality representation of each topic. A customized version of WordNet is also built to include domain-related terms based on the maximal frequent itemsets found in the corpus. Furthermore, we propose to find the best number of topics covered by the corpus by employing a clustering algorithm to

cluster topics based on their similarities in order to get more coherent topics. We further analyze how topics' coverage and users' interests change over time. The proposed method is applied on two real-life fashion datasets collected from Twitter. The obtained results suggest that our method is better than Twitter-LDA in terms of perplexity, topics' coherence, and their quality.

Second, several characteristics of social media facilitate the process of posting information with unestablished truth values and its fast diffusion among users all over the world. Therefore, the task of distinguishing verified information from unverified rumors spreading in social media becomes an extremely difficult and crucial task. Breaking news rumors, if not identified as early as possible, may have extremely damaging consequences. This thesis tackles the problem of identifying breaking news rumors of emerging topics spreading in social media by proposing a model that jointly builds the word2vec model and the LSTM-RNN rumor detection model. The proposed model is capable of accurately identifying breaking news rumors solely based on the text of a tweet. Our experiments on real-life datasets show that the performance of our proposed model outperforms the state-of-the-art classifier as well as other baseline classifiers in terms of precision, recall, and F1.

Finally, the uncertainty and chaos associated with hot and sensitive breaking news and emergency situations facilitates the explosive spread of high-engaging rumors that might be extremely damaging. Overwhelming the authorities with huge volumes of breaking news rumors makes the process of verifying their contents and acting upon them quickly to reduce their damaging consequences an extremely challenging task. Fortunately, not all breaking news rumors will spread in social media. This thesis tackles the problem of identifying high-engaging breaking news rumors of emerging topics spreading in social media by proposing a multi-task neural network model that jointly learns the breaking news rumor detection and breaking news rumors popularity prediction tasks. Our experiments on real-life datasets show that the performance of the joint learning model outperforms other baseline classifiers in terms of precision, recall, and F1 and is capable of identifying high-engaging breaking news rumors with high accuracy.

In this thesis, we show that incorporating the semantic relations in a lexical database and leveraging from the strong topics' indicators in social media, such as the set of hashtags, helps topic modeling algorithms overcome the lack of co-occurrence patterns and the high sparseness challenges of modeling micro-posts in social media. Furthermore, it helps topic modeling algorithms

produce an improved and cohesive set of keywords to represent each extracted topic. We also show that training a word2vec model in parallel to a deep learning model and using it to update the embedding space on the fly with every new textual data it receives significantly outperformed the typical methods of embedding training in mitigating the cross-topic and Out-Of-Vocabulary (OOV) issues associated with emerging breaking news rumors. The simplicity and effectiveness of this training technique can help a deep learning model adaptively capture the drift in the data and mitigate the well-known OOV issues in many natural language processing tasks. Furthermore, we show that it is feasible to achieve the task of high-engaging breaking news rumors detection by leveraging from the shared characteristics between well-written rumors and popular posts in social media. We also show that the inclusion of emotional triggers and stylometric features can effectively help improve the detection of potentially high-engaging unverified information circulating in social media. Moreover, these features along with a word2vec model and a deep learning model can be used to build a highly accurate model for predicting the future popularity of a post in social media without monitoring its early dynamics in the social network or collecting related and similar posts.

# Chapter 8

# Future Directions

With the vast number of users who initiate and spread information in social media websites, the next question is to what extent can we trust the sources of such information and be willing to spread it? People tend to build their trust of information posted or reposted by other users based on personal experience. They also tend to follow popular accounts, repost, and comment on the information posted by them. Although such popular accounts are not always trustworthy sources of information, information posted by them spreads very fast in social media.

Trust has been defined differently throughout the existing literature. For example, Jøsang et al. [48] defined trust as "the extent to which one party is willing to depend on something or somebody in a given situation with a feeling of relative security, even though negative consequences are possible." Similarly, Hamdi et al. [38] considered trust as a subjective measure that describes "how far, a given entity A considers another entity B as trustworthy."

Several works in the literature have tackled the problem of inferring the trustworthiness of users in online social networks such as [33, 38, 44, 92]. However, most existing work calculates the trust values in social media based on the structure of the network and the existence of direct paths and interactions between users. The existing work does not fully address the problem of assessing trust in social media for several reasons. First, a social media user has to face the risk of dealing with other social media users who are unknown to him/her. This raises the need of having tools to help a user assess the trustworthiness of other users in social media, regardless of his/her connection to that user. Furthermore, following a user in social media does not necessarily mean that the follower

trusts the followee. In fact, many social media users are keen to pursue controversial social media accounts that they do not trust. Also, existing studies did not take into consideration the contents posted by users to assess their trustworthiness.

As a future work, we aim at building upon our work in this thesis and study the challenges of automatically assessing the trustworthiness of users in social media based on their characteristics and the contents posted by them while taking time into consideration. We want to explore the usage of representation learning to learn richer feature representations that capture the lexical, network, and user features. Our objective is to automatically learn more hidden features for a more insightful analysis of social media users.

# Bibliography


[1] Sofiane Abbar, Carlos Castillo, and Antonio Sanfilippo. To post or not to post: Using online trends to predict popularity of offline content. In *Proceedings of the 29th on Hypertext and Social Media*, HT '18, pages 215–219, New York, NY, USA, 2018. ACM.

[2] Thomas Abeel, Yves Van de Peer, and Yvan Saeys. Java-ml: A machine learning library. *The Journal of Machine Learning Research*, 10:931–934, June 2009. ISSN 1532-4435.

[3] Nikolaos Aletras and Mark Stevenson. Evaluating topic coherence using distributional semantics. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS'13) Long Papers*, pages 13–22, 2013.

[4] Sarah A. Alkhodair, Benjamin C. M. Fung, Osmud Rahman, and Patrick C. K. Hung. Improving interpretations of topic modeling in microblogs. *Journal of the Association for Information Science and Technology (JASIST)*, 69(4):528–540, April 2018. ISSN 2330-1635.

[5] Sarah A. Alkhodair, Steven H. H. Ding, Benjamin C. M. Fung, and Junqiang Liu. Detecting breaking news rumors of emerging topics in social media. *Information Processing and Management Journal - A Special Issue on Mining Social Influence and Actionable Insights from Social Networks (in press)*, 2019.

[6] Gordon W. Allport and Leo J. Postman. *The psychology of rumor*. Russell & Russell, 1965.

[7] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv e-prints*, abs/1409.0473, September 2014.

[8] Roja Bandari, Sitaram Asur, and Bernardo A. Huberman. The pulse of news in social media: Forecasting popularity. *CoRR*, abs/1202.0332, 2012.

[9] Abdurrahman M. A. Basher and Benjamin C. M. Fung. Analyzing topics and authors in chat logs for crime investigation. *Knowledge and Information Systems (KAIS)*, 39(2):351–381, May 2014.

[10] Kayhan N. Batmanghelich, Ardavan Saeedi, Karthik Narasimhan, and Samuel Gershman. Nonparametric spherical topic modeling with word embeddings. *Computing Research Repository*, abs/1604.00126, 2016.

[11] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *The Journal of Machine Learning Research*, 3:1137–1155, March 2003.

[12] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8): 1798–1828, August 2013.

[13] Jonah A. Berger and Katherine Milkman. What makes online content viral? *Journal of Marketing Research*, 49, December 2009.

[14] David M. Blei and John D. Lafferty. Dynamic topic models. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 113–120, New York, NY, USA, 2006. ACM.

[15] David M. Blei and John D. Lafferty. A correlated topic model of science. *The Annals of Applied Statistics*, 1(1):17–35, 2007.

[16] David M. Blei and John D. Lafferty. Topic models. In Ashok N. Srivastava and Mehran Sahami, editors, *Text Mining: Classification, Clustering, and Applications*, chapter 4, pages 71–94. Chapman and Hall/CRC, New York, 2009.

[17] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, March 2003.

[18] Jordan L. Boyd-Graber, David M. Blei, and Xiaojin Zhu. A topic model for word sense disambiguation. In *Conference on Empirical Methods in Natural Language Processing Conference on Computational Natural Language Learning (EMNLP-CoNLL)*, pages 1024–1033. ACL, 2007.

[19] Douglas Burdick, Manuel Calimlim, and Johannes Gehrke. Mafia: A maximal frequent itemset algorithm for transactional databases. In *Proceedings of the 17th International Conference on Data Engineering*, pages 443–452, Washington, DC, USA, 2001. IEEE Computer Society.

[20] Feng Chen and Wai Hong Tan. Marked self-exciting point process modelling of information diffusion on twitter. *The Annals of Applied Statistics*, 12:2175–2196, December 2018.

[21] Tong Chen, Lin Wu, Xue Li, Jun Zhang, Hongzhi Yin, and Yang Wang. Call attention to rumors: Deep attention based recurrent neural networks for early rumor detection. *CoRR*, abs/1704.05973, 2017.

[22] KyungHyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *CoRR*, abs/1409.1259, 2014.

[23] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, pages 160–167, New York, NY, USA, 2008. ACM.

[24] Massimo Crescimbene, Federica La Longa, and Tiziana Lanza. The science of rumors. *Annals of geophysics*, 55, July 2012.

[25] Andrew M. Dai and Amos J. Storkey. Author disambiguation: A nonparametric topic and co-authorship model. In *NIPS Workshop on Applications for Topic Models Text and Beyond*, pages 1–4, 2009.

[26] Nicholas Difonzo and Prashant Bordia. Rumors influence: Toward a dynamic social impact

theory of rumor. *Science of social influence: Advances and future progress*, pages 271–296, January 2007.

[27] Steven H. H. Ding, Benjamin C. M. Fung, and Philippe Charland. Asm2vec: Boosting static representation robustness for binary clone search against code obfuscation and compiler optimization. In *Proc. of the 40th International Symposium on Security and Privacy (S&P)*, pages 38–55, San Francisco, CA, May 2019. IEEE Computer Society.

[28] Steven H. H. Ding, Benjamin C. M. Fung, Farkhund Iqbal, and William K. Cheung. Learning stylometric representations for authorship analysis. *IEEE Transactions on Cybernetics (CYB)*, 49(1):107–121, January 2019.

[29] Avinava Dubey, Ahmed Hefny, Sinead Williamson, and Eric P. Xing. A nonparametric mixture model for topic modeling over time. In *SDM*, 2013.

[30] Branden Fitelson. A probabilistic theory of coherence. *Analysis*, 63(279):194–199, 2003.

[31] W. R. Gilks, S. Richardson, and D. Spiegelhalter. *Markov Chain Monte Carlo in Practice*. Chapman & Hall/CRC Interdisciplinary Statistics. CRC Press LLC, 1996. ISBN 9780412055515.

[32] Fréderic Godin, Viktor Slavkovikj, Wesley De Neve, Benjamin Schrauwen, and Rik Van de Walle. Using topic models for twitter hashtag recommendation. In *Proceedings of the 22Nd International Conference on World Wide Web*, pages 593–596, Republic and Canton of Geneva, Switzerland, 2013. International World Wide Web Conferences Steering Committee.

[33] Jennifer Ann Golbeck. *Computing and Applying Trust in Web-based Social Networks*. PhD thesis, University of Maryland, College Park, MD, USA, 2005.

[34] Yoav Goldberg and Graeme Hirst. *Neural Network Methods in Natural Language Processing*. Morgan & Claypool Publishers, 2017. ISBN 1627052984, 9781627052986.

[35] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[36] Alex Graves. Generating sequences with recurrent neural networks. *CoRR*, abs/1308.0850, 2013.

[37] Weiwei Guo and Mona Diab. Modeling sentences in the latent space. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, pages 864–872, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[38] Sana Hamdi, Alda Lopes Gancarski, Amel Bouzeghoub, and Sadok Ben Yahia. Tison: Trust inference in trust-oriented social networks. *ACM Transactions on Information Systems (TOIS)*, 34(3):17:1–17:32, April 2016.

[39] Sardar Hamidian and Mona Diab. Rumor identification and belief investigation on twitter. In *Proceedings of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, 2016.

[40] Naeemul Hassan, Chengkai Li, and Mark Tremayne. Detecting check-worthy factual claims in presidential debates. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, pages 1835–1838, New York, NY, USA, 2015. ACM.

[41] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computing*, 9 (8):1735–1780, November 1997.

[42] Dichao Hu. An introductory survey on attention mechanisms in NLP problems. *CoRR*, abs/1811.05544, 2018.

[43] Anil K. Jain and Richard C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, Inc., 1988. ISBN 013022278X.

[44] Wenjun Jiang, Guojun Wang, and Jie Wu. Generating trusted graphs for trust evaluation in online social networks. *Future Generation Computer Systems*, 31:48–58, February 2014.

[45] Zhiwei Jin, Juan Cao, Yu-Gang Jiang, and Yongdong Zhang. News credibility evaluation on microblog with a hierarchical propagation model. In *Proceedings of the 2014 IEEE International Conference on Data Mining*, ICDM '14, pages 230–239, Washington, DC, USA, 2014. IEEE Computer Society.

[46] Zhiwei Jin, Juan Cao, Yongdong Zhang, and Jiebo Luo. News verification by exploiting conflicting social viewpoints in microblogs. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 2972–2978. AAAI Press, 2016.

[47] Zhiwei Jin, Juan Cao, Han Guo, Yongdong Zhang, and Jiebo Luo. Multimodal fusion with recurrent neural networks for rumor detection on microblogs. In *Proceedings of the 2017 ACM on Multimedia Conference*, MM '17, pages 795–816, New York, NY, USA, 2017. ACM.

[48] Audun Jøsang, Roslan Ismail, and Colin Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 43(2):618–644, March 2007.

[49] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751. Association for Computational Linguistics, 2014.

[50] Dan Knights, Michael C. Mozer, and Nicolas Nicolov. Detecting topic drift with compound topic models. In *Proceedings of the 3rd International Conference on Weblogs and Social Media (ICWSM)*. The AAAI Press, 2009.

[51] Ryota Kobayashi and Renaud Lambiotte. Tideh: Time-dependent hawkes process for predicting retweet dynamics. In *Proceedings of the 10th International AAAI Conference on Web and Social Media (ICWSM 2016)*, May 2016.

[52] Sejeong Kwon, Meeyoung Cha, and Kyomin Jung. Rumor detection over varying time windows. *PLOS ONE*, 12(1):1–19, January 2017.

[53] Yann Lecun, Lon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998.

[54] Haitao Li, Xiaoqiang Ma, Feng Wang, Jiangchuan Liu, and Ke Xu. On popularity prediction of videos shared in online social networks. In *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*, CIKM '13, pages 169–178, New York, NY, USA, 2013. ACM.

[55] Yitan Li, Linli Xu, Fei Tian, Liang Jiang, Xiaowei Zhong, and Enhong Chen. Word embedding revisited: A new representation learning and explicit matrix factorization perspective. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pages 3650–3656. AAAI Press, 2015.

[56] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. Learning context-sensitive word embeddings with neural tensor skip-gram model. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pages 1284–1290. AAAI Press, 2015.

[57] Xiaomo Liu, Armineh Nourbakhsh, Quanzhi Li, Rui Fang, and Sameena Shah. Real-time rumor debunking on twitter. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1867–1870, New York, NY, USA, 2015. ACM.

[58] Xiaozhong Liu and Howard Turtle. Real-time user interest modeling for real-time ranking. *Journal of the American Society for Information Science and Technology*, 64(8):1557–1576, 2013.

[59] Yang Liu, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. Topical word embeddings. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 2418–2424. AAAI Press, 2015.

[60] Hsin-Min Lu. Wordnet-enhanced topic models. In *Mining Data Semantics in Heterogeneous Information Networks Workshop (MDS'2013; in conjunction with ACM SIGKDD Conference on Knowledge Discovery and Data Mining)*, Chicago, Illinois, USA, 2013.

[61] Michal Lukasik, Kalina Bontcheva, Trevor Cohn, Arkaitz Zubiaga, Maria Liakata, and Rob Procter. Using gaussian processes for rumour stance classification in social media. *CoRR*, 2016.

[62] Jing Ma, Wei Gao, Zhongyu Wei, Yueming Lu, and Kam-Fai Wong. Detect rumors using time series of social context information on microblogging websites. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1751–1754, New York, NY, USA, 2015. ACM.

[63] Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J. Jansen, Kam-Fai Wong, and Meeyoung Cha. Detecting rumors from microblogs with recurrent neural networks. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 3818–3824. AAAI Press, 2016.

[64] Zhiqiang Ma, Wenwen Dou, Xiaoyu Wang, and Srinivas Akella. Tag-latent dirichlet allocation: Understanding hashtags and their relationships. In *Proceedings of the 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, pages 260–267, Washington, DC, USA, 2013. IEEE Computer Society.

[65] Katerina Eva Matsa and Elisa Shearer. News use across social media platforms 2018. Report, Pew Research Center, September 2018.

[66] Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, December 1943.

[67] Graham McDonald, Craig Macdonald, and Iadh Ounis. Using part-of-speech n-grams for sensitive-text classification. In *Proceedings of the 2015 International Conference on The Theory of Information Retrieval*, ICTIR '15, pages 381–384, New York, NY, USA, 2015. ACM.

[68] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, pages 3111–3119, USA, 2013. Curran Associates Inc.

[69] George A. Miller. Wordnet: A lexical database for english. *Communications of the ACM*, 38 (11):39–41, November 1995.

[70] Swapnil Mishra, Marian-Andrei Rizoiu, and Lexing Xie. Feature driven and point process approaches for popularity prediction. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, pages 1069–1078, New York, NY, USA, 2016. ACM.

[71] Amy Mitchell, Katie Simmons, Katerina Eva Matsa, and Laura Silver. People in poorer countries just as likely to use social media for news as those in wealthier countries. Report, Pew Research Center, January 2018.

[72] Saif M. Mohammad and Peter D. Turney. Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, CAAGET '10, pages 26–34, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[73] Ebony S. Muhammad. The psychology of rumors: Why they spread & how to avoid being misled, October 2017. URL http://hurt2healingmag.com/the-psychology-of-rumors-why-they-spread-how-to-avoid-being-misled/.

[74] Claudiu Musat, Julien Velcin, Marian-Andrei Rizoiu, and Stefan Trausan-Matu. Concept-based topic model improvement. In Dominik Ryżko, Henryk Rybiński, Piotr Gawrysiak, and Marzena Kryszkiewicz, editors, *Emerging Intelligent Technologies in Industry*. Springer Berlin Heidelberg, 2011.

[75] David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. Automatic evaluation of topic coherence. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 100–108, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[76] Michael A. Nielsen. *Neural Networks and Deep Learning*. Determination Press, 2015.

[77] Pinar Ozturk, Huaye Li, and Yasuaki Sakamoto. Combating rumor spread on social media: The effectiveness of refutation and warning. In *Proceedings of the 2015 48th Hawaii International Conference on System Sciences*, HICSS '15, pages 2406–2414, Washington, DC, USA, 2015. IEEE Computer Society.

[78] Martin F. Porter. An algorithm for suffix stripping. In Karen Sparck Jones and Peter Willett, editors, *Readings in Information Retrieval*, pages 313–316. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.

[79] Vahed Qazvinian, Emily Rosengren, Dragomir R. Radev, and Qiaozhu Mei. Rumor has it: Identifying misinformation in microblogs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1589–1599, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[80] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013. URL http://www.R-project.org/.

[81] Michael Röder, Andreas Both, and Alexander Hinneburg. Exploring the space of topic coherence measures. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, pages 399–408, New York, NY, USA, 2015. ACM.

[82] Michal Rosen-Zvi, Chaitanya Chemudugunta, Thomas Griffiths, Padhraic Smyth, and Mark Steyvers. Learning author-topic models from text corpora. *ACM Transactions on Information Systems (TOIS)*, 28(1):4:1–4:38, January 2010.

[83] Frank Rosenblatt. *Principles of neurodynamics: perceptrons and the theory of brain mechanisms*. Report (Cornell Aeronautical Laboratory). Spartan Books, 1962.

[84] Natali Ruchansky, Sungyong Seo, and Yan Liu. Csi: A hybrid deep model for fake news detection. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, pages 797–806, New York, NY, USA, 2017. ACM.

[85] Stan Salvador and Philip Chan. Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. In *Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence*, pages 576–584, Washington, DC, USA, 2004. IEEE Computer Society.

[86] Kentaro Sasaki, Tomohiro Yoshikawa, and Takeshi Furuhashi. Online topic model for twitter considering dynamics of user interests and topic trends. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1977–1985, Doha, Qatar, October 2014. Association for Computational Linguistics.

[87] Jieying She and Lei Chen. Tomoha: Topic model-based hashtag recommendation on twitter. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 371–372, Republic and Canton of Geneva, Switzerland, 2014. International World Wide Web Conferences Steering Committee.

[88] Elisa Shearer and Jeffrey Gottfried. News use across social media platforms 2017. Report, Pew Research Center, September 2017.

[89] Jun Song, Yu Huang, Xiang Qi, Yuheng Li, Feng Li, Kun Fu, and Tinglei Huang. Discovering hierarchical topic evolution in time-stamped documents. *Journal of the Association for Information Science and Technology*, 2015.

[90] Jiahao Su, Ming Fang, Jiang Jiang, and Ying-Wu Chen. An evolutionary game model of multi-topics diffusion in social network. *ITM Web of Conferences*, 12:03045, January 2017.

[91] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 3104–3112, Cambridge, MA, USA, 2014. MIT Press.

[92] Mohsen Taherian, Morteza Amini, and Rasool Jalili. Trust inference in web-based social networks using resistive networks. In *Proceedings of the 2008 Third International Conference on Internet and Web Applications and Services*, pages 233–238, Washington, DC, USA, 2008. IEEE Computer Society.

[93] Misako Takayasu, Kazuya Sato, Yukie Sano, Kenta Yamada, Wataru Miura, and Hideki Takayasu. Rumor diffusion and convergence during the 3.11 earthquake: A twitter case study. *PLoS ONE*, 10, April 2015.

[94] Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101, 2004.

[95] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike Von

Luxburg, Samy Bengio, Hanna Wallach, Rob Fergus, S.V.N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017.

[96] William Yang Wang. "liar, liar pants on fire": A new benchmark dataset for fake news detection. *CoRR*, abs/1705.00648, 2017.

[97] Xuerui Wang and Andrew McCallum. Topics over time: A non-markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 424–433, New York, NY, USA, 2006. ACM.

[98] Daniel Xie, Jiejun Xu, and Tsai-Ching Lu. What's trending tomorrow, today: Using early adopters to discover popular posts on tumblr. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 2168–2176, December 2017.

[99] Yan Yan, Zhaowei Tan, Xiaofeng Gao, Shaojie Tang, and Guihai Chen. Sth-bass: A spatial-temporal heterogeneous bass model to predict single-tweet popularity. In Shamkant B. Navathe, Weili Wu, Shashi Shekhar, Xiaoyong Du, Sean X. Wang, and Hui Xiong, editors, *Database Systems for Advanced Applications*, pages 18–32, Cham, 2016. Springer International Publishing.

[100] Zaihan Yang, Alexander Kotov, Aravind Mohan, and Shiyong Lu. Parametric and non-parametric user-aware sentiment topic models. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 413–422, New York, NY, USA, 2015. ACM.

[101] Tauhid Zaman, Emily B. Fox, and Eric T. Bradlow. A bayesian approach for predicting the popularity of tweets. *The Annals of Applied Statistics*, 8(3):1583–1611, September 2014.

[102] Ye Zhang and Byron Wallace. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 253–263. Asian Federation of Natural Language Processing, 2017.

[103] Qingyuan Zhao, Murat A. Erdogdu, Hera Y. He, Anand Rajaraman, and Jure Leskovec. Seismic: A self-exciting point process model for predicting tweet popularity. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pages 1513–1522, New York, NY, USA, 2015. ACM.

[104] Wayne Xin Zhao, Jing Jiang, Jianshu Weng, Jing He, Ee-Peng Lim, Hongfei Yan, and Xiaoming Li. Comparing twitter and traditional media using topic models. In *Proceedings of the 33rd European Conference on Advances in Information Retrieval*, pages 338–349, Berlin, Heidelberg, 2011. Springer-Verlag.

[105] Zhe Zhao, Paul Resnick, and Qiaozhu Mei. Enquiring minds: Early detection of rumors in social media from enquiry posts. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1395–1405, Republic and Canton of Geneva, Switzerland, 2015. International World Wide Web Conferences Steering Committee.

[106] Arkaitz Zubiaga, Geraldine W. S. Hoi, Maria Liakata, and Rob Procter. Pheme dataset of rumours and non rumours, November 2016.

[107] Arkaitz Zubiaga, Maria Liakata, and Rob Procter. Learning reporting dynamics during breaking news for rumour detection in social media. *CoRR*, 2016.

[108] Arkaitz Zubiaga, Ahmet Aker, Kalina Bontcheva, Maria Liakata, and Rob Procter. Detection and resolution of rumours in social media: A survey. *ACM Computing Surveys*, 51(2):32:1–32:36, February 2018.