Macro-Micro Approach for Mining Public Sociopolitical Opinion from Social Media

by

Bo Wang

Thesis

Submitted to the University of Warwick

to obtain the degree of

Doctor of Philosophy

Department of Computer Science

October 2017

This thesis is dedicated to my parents, Jianbao Wang and Xia Liu.

Acknowledgments

First and foremost, I would like to express my sincere gratitude to my supervisors, Dr. Maria Liakata and Prof. Rob Procter. Many thanks to their invaluable guidance and encouragement for guiding me to tackle all the research challenges during my PhD. I could not have asked for better mentors.

I would like to extend my gratitude to Arkaitz Zubiaga for his advice throughout my PhD. I am also grateful to have Dr. Theo Damoulas, Prof. Mike Joy, and Prof. Alexandra I. Cristea as my annual reviewers, for their valuable suggestions on my PhD progress.

I would also like to thank my friends, colleagues at the department, fellow members of WarwickNLP, and comrades at CS229 for their support and friendship. I hope part of us never grows up.

Furthermore, I am thankful to all my co-authors.

I would like to thank the most important people in my life, my mother Xia Liu, and my father Jianbao Wang. I can never thank them enough for their love and support. I would also like to thank my grandparents and other members of my family.

Finally, I would like to thank everyone who believed in me and saw something in me when I did not see myself.

Declarations

I hereby declare that the work presented in this thesis entitled *Macro-Micro Approach for Mining Public Sociopolitical Opinion from Social Media* is an original work and has not been submitted to any college, university or any other academic institution for the purpose of obtaining an academic degree.

Abstract

During the past decade, we have witnessed the emergence of social media, which has prominence as a means for the general public to exchange opinions towards a broad range of topics. Furthermore, its social and temporal dimensions make it a rich resource for policy makers and organisations to understand public opinion. In this thesis, we present our research in understanding public opinion on Twitter along three dimensions: sentiment, topics and summary.

In the first line of our work, we study how to classify public sentiment on Twitter. We focus on the task of multi-target-specific sentiment recognition on Twitter, and propose an approach which utilises the syntactic information from parse-tree in conjunction with the left-right context of the target. We show the state-of-the-art performance on two datasets including a multi-target Twitter corpus on UK elections which we make public available for the research community. Additionally we also conduct two preliminary studies including cross-domain emotion classification on discourse around arts and cultural experiences, and social spam detection to improve the signal-to-noise ratio of our sentiment corpus.

Our second line of work focuses on automatic topical clustering of tweets. Our aim is to group tweets into a number of clusters, with each cluster representing a meaningful topic, story, event or a reason behind a particular choice of sentiment. We explore various ways of tackling this challenge and propose a two-stage hierarchical topic modelling system that is efficient and effective in achieving our goal.

Lastly, for our third line of work, we study the task of summarising tweets on common topics, with the goal to provide informative summaries for real-world events/stories or explanation underlying the sentiment expressed towards an issue/entity. As most existing tweet summarisation approaches rely on extractive methods, we propose to apply state-of-the-art neural abstractive summarisation model for tweets. We also tackle the challenge of cross-medium supervised summarisation with no target-medium training resources. To the best of our knowledge, there is no existing work on studying neural abstractive summarisation on tweets. In addition, we present a system for providing interactive visualisation of topic-entity sentiments and the corresponding summaries in chronological order.

Throughout our work presented in this thesis, we conduct experiments to evaluate and verify the effectiveness of our proposed models, comparing to relevant baseline methods. Most of our evaluations are quantitative, however, we do perform qualitative analyses where it is appropriate. This thesis provides insights and findings that can be used for better understanding public opinion in social media.

Contents

Abbreviations

AP Affinity Propagation

AdaRNN Adaptive Recurrent Neural Network

CNN Convolutional Neural Network

DMM Dirichlet Multinomial Mixture

ED Event Detection

FSD First Story Detection

G3 Three-way Gate

GRNN Gated Recurrent Neural Network

GSDMM Gibbs Sampling algorithm for the Dirichlet Multinomial Mixture

HC Hierarchical Clustering

ILP Integer Linear Programming

KLD Kullback-Leibler Divergence

LCTM Latent Concept Topic Model

LFTM Latent Feature Topic Models

LM Language Model

LSTM Long Short Term Memory

LDA Latent Dirichlet Allocation

NPMI Normalised Pointwise Mutual Information

OLDA Online Latent Dirichlet Allocation

OOV Out of Vocabulary

PMI Pointwise Mutual Information

RNN Recurrent Neural Network

SMS Short Message Service

SSWE Sentiment-Specific Word Embedding

SVM Support Vector Machine

TOLDA Online LDA for Twitter

Tf-idf Term frequency-inverse document frequency

TCLSTM Target Connection Long Short Term Memory

TDLSTM Target Dependent Long Short Term Memory

WE Word Embedding

WMD Word Mover's Distance

List of Tables

List of Figures

CHAPTER 1

Introduction

In April 2013, a global information services company, Experian, reported that of every hour the British spend online, 13 minutes are on social media - more than entertainment, shopping, checking the news, email or anything else[1]. Social media has gained prominence as a means for the general public and high-profile governmental figures such as the president of the US, to express opinions towards a broad range of topics, including social and political issues. A 2013 study [3] reported the two major UK political parties had more Twitter followers than their formal party members. For those who are not considered as mainstream parties social media has provided a great and cost-less arena for their voices. Beppe Grillo, a Italian comedian with no history of politics, utilised social media for his "Five Stars Movement". In 2013 his party won 25.55% of the vote for the Chamber of Deputies and thus "Five Stars Movement" became the largest party in the Chamber of Deputies. Therefore we are witnessing social media as a platform that is fast changing the public discourse in society and setting trends in topics that range from urban environment and traffic to politics and entertainment. With the enormity of social media data and its constantly evolving nature, it also provides us the form of collective wisdom that can be utilised to analyse collective behaviors, understand emerging social, economic and

[1] `https://goo.gl/nZQp9e`

political phenomena.

With the explosive growth of user-generated content from social media sites such as Twitter, in recent years we have seen a rapidly increasing research interest in using this new type of data to understand, analyse, represent and extract a range of actionable patterns [4]. This includes discovering bursty topics [5, 6], constructing user profiles [7, 8, 9], recognising sentiments or emotions expressed in social media posts [10, 11, 12] and even predicting real world outcomes [13, 14, 15]. Although social media mining has become a popular research area, understanding public opinion towards social-political topics such as election remains a challenge. Social media posts tend to be short, e.g. tweets have a 140-character limit. They are also noisy and often contain SMS lingo, misspelling and broken sentences, resulting in data sparsity issue. Social spam and posts that spreading fake news or misinformation also contribute greatly to the noisy nature of social media data. With regard to public opinion on social-political issues, it is also difficult to parse the author's attitude towards the object and the real intent behind such sentiment (e.g. Poe's law). In addition, the dynamic nature of social media data streams leads to *topic or concept drift*, which in turn requires efficient and time-sensitive systems. Therefore, conventional text mining methods cannot be directly applied to understand social media data.

Sentiment is a very important element and a key factor in understanding public opinion, as in the case of political events, where opinions are often expressed through positive or negative sentiments. In recent years, sentiment analysis has been applied to detect and track public sentiment from social media [10, 13, 16, 17]. Delving into the field of sentiment analysis, finer-grained sentiment has also been studied to provide granularity from different angles such as emotion analysis [18, 19] and target-specific sentiment classification [11, 1, 20]. Clustering of social media posts is another important research area [21, 22, 23]. By grouping user-generated posts into thematic topics, not only is it easier for users to digest large volumes of

data, but a range of professionals are also able to rely on social media for analysing public opinion, particularly focusing on specific topics and for instance connecting sentiment expressed towards a topic to the real-world event. Finally to fully digest public opinions, text summarisation has also been applied to social media, where a condensed summary is generated in capturing the narrative surrounding a topic or event [24, 25, 26]. In addition to the aforementioned research areas, other challenges in social media mining that are gaining research interest include social spam filtering [27, 28, 29], domain adaptation [30, 31], sarcasm recognition [32] and rumour detection [33, 34], all of which can also help analyse and quantitatively assess public opinion through social media.

In this thesis, we continue previous research on social media mining and contribute our work in understanding public opinion on Twitter from three different yet interconnected directions: *sentiment*, *topics* and *summary*. Our first line of work is the sentiment classification of tweets. We focus on the problem of target-specific sentiment recognition and introduce a challenging task of identifying sentiment towards multiple targets in a tweet. Additionally, considering that the language use varies across domains, we also explore ways to alleviate such domain issue by exploring domain adaptation for Twitter emotion classification. Our second line of work focuses on the topical clustering of tweets. Our aim is to assign every tweet of a large Twitter corpus to the corresponding cluster, with each cluster representing a thematic topic, story, event or a reason underlying a particular choice of sentiment. To achieve this, we propose a two-stage hierarchical topic modelling system. Lastly, our third line of work applies abstractive multi-document summarisation for explaining the reasons behind the sentiment towards particular entities. Additionally, we present an interactive web interface which provides the visualisation of sentiments and corresponding extractive summaries in chronological order.

1.1 Research Outline and Questions

The main question that motivates the research underlying this thesis is: *How can we better understand public opinion on social media?* We propose to analyse public opinion on Twitter, with respect to specific socio-political issues from both macro and micro perspectives. By recognising target-specific sentiment and providing public sentiment evolution for a socio-political event such as elections, researchers can analyse such opinion towards different issues and entities and understand how it develops over time, *on the macro level*. With topical clustering of tweets and opinion summarisation, we provide a system for adding the explanation and justification behind why such sentiment is commonly expressed towards a particular entity on a particular day observed on the macro scale, and thus offers *a micro perspective*. Our proposed approach can be used by policy makers and organisations to better understand public opinion on social media.

This thesis aims to advance the state-of-the-art on all of the aforementioned research areas. In Chapter 2 we provide the literature review for these areas. In Chapter 3 we present two preliminary studies including social spam detection for improving the signal-to-noise ratio of our sentiment corpus, and Twitter cross-domain emotion analysis using domain adaptation. Starting from Chapter 4 we describe our work on three research questions (**RQ1** – **RQ3**) that we believe are important for understanding public opinion on social media, and contribute new solutions to each research challenge.

An important challenge of Twitter sentiment analysis is to distinguish and detect sentiment expressed towards different targets appearing in the same tweet. Jiang et al. [11] showed that 40% of classification errors are caused by only considering the overall sentiment expressed in an entire tweet and ignoring the fact that often a tweet contains different types of sentiment expressed towards different targets. To understand public opinion on social media, it is essential to not only

analyse the overall public mood, but also to identify sentiment towards different key issues and entities. In Chapter 4 we address the following question:

RQ1: *How can we infer the sentiment towards a specific target as opposed to tweet-level sentiment? Can we find an effective approach for identifying sentiment towards multiple targets within a tweet?*

In answering this question, we move away from the assumption that each tweet mentions a single target and introduce a more realistic task of identifying sentiment towards multiple targets within a tweet. To tackle this challenge, we build a multi-target corpus that is far more challenging and contains more diverse opinions towards different socio-political issues. We investigate different approaches of utilising syntactic dependencies of the targets, and propose a method that combines such syntactic information for each target with its left-right context, showing competitive performance.

While social media is a rich resource to shed light on public sentiment and to track real-world stories, it is often difficult for humans to digest and keep track of all the relevant information provided in the large volumes of data. Automatic topical clustering of tweets can help to produce a manageable list of topics that is much easier for users to digest, enabling for instance identification of real-world events among those topics. In contrast to topic detection from newswire articles or scientific journal documents clustering, to cluster social media posts such as tweets topically is more difficult. Such user-generated content usually lack context due to their brevity (e.g. 140 characters for tweets) and are noisy in nature. As a consequence, traditional document clustering approaches and conventional topic models fall short of delivering good performance. This motivates us to ask **RQ2**, and in Chapter 5 we aim to address such problem by studying and proposing a state-of-the-art topic modelling system.

RQ2: *Can we develop a system to effectively group tweets to a number of clusters, with each cluster representing a thematic topic?*

In our last step towards understanding how public opinions are shaped on Twitter, we study the task of multi-document summarisation for tweets. Continuing our work in Chapter 4 and Chapter 5, in Chapter 6 we present a system for time-sensitive, topic-based summarisation of sentiment towards different issues and entities, with the goal of providing explanation and justification behind such sentiment. Most existing tweet summarisation approaches rely on extractive methods, which identify such task as a selection or ranking problem. In our work we also set out to find an abstractive summarisation model that can resemble how humans write summaries, which is a more challenging task.

Recently neural sequence-to-sequence learning models (or seq2seq) [35] have shown success in various NLP tasks including machine translation and abstractive summarisation for news articles. While seq2seq presents a promising way forward for abstractive summarisation, extrapolating such approach for social media posts such as tweets, is not trivial. To the best of our knowledge, there is currently no study on applying seq2seq on tweets. One key issue here is the lack of or non-existence for sufficient training data. In Chapter 6, we are motivated to address the challenges in **RQ3**, by applying the state-of-the-art neural abstractive summarisation model with a pretraining step.

RQ3: *How can we generate abstractive summaries for opinions towards common topics expressed on Twitter? Is it possible to generate tweet abstracts from scratch with limited training resources?*

We answer these five research questions in the discussion and conclusion sections of each individual chapter between Chapter 4 and Chapter 6. In Chapter 7 we summarise our findings and suggest future research directions. In the next sections, we list the contributions of this thesis to the research field, and the publications of each line of work.

1.2 Main Contributions

In this section we describe our contributions which can be classified in four categories: introducing new tasks, proposing new models, performing new analyses, and releasing data and code.

Social spam detection Unlike most existing spammer detection studies which rely on extensive and expensive user data, we propose to study the categorisation of a tweet as spam or not from its inherent features that can be obtained in real time. We compare five classification algorithms over two different datasets, thus providing an important evaluation for future studies.

Twitter emotion classification We evaluate the model-based adaptive-SVM approach against a set of domain-dependent and domain-independent strategies, in both classification performance and computation time cost. We also make our annotated emotion corpus and code available to the public.

Target-specific sentiment recognition We introduce the task of multi-target-specific sentiment classification for Twitter data. Annotated corpus is important for the research community to benchmark their systems and further the performance for multi-target sentiment classification. We construct and release to the public a new multi-target sentiment corpus that contains far more target entities (as well as topics) and thus more challenging. We propose a new target-specific sentiment model that combines context around a target and its syntactic dependencies. Comparing with both target-independent and target-dependent approaches in both a single-target and our multi-target datasets, our proposed model shows state-of-the-art performance. We also conduct experiment by dividing data into three subsets based on the number of distinct target sentiment values per tweet. Our analysis provides insight on how target-independent and target-dependent models perform for each of

these scenarios, which can be used as a basis for future improvement. The implementation code for this work is also made available to the public.

Topical clustering of tweets We propose a two-stage hierarchical topic modelling system, in which we leverage a state-of-the-art Twitter topic model, a topic model incorporating word embeddings and a tweet pooling step without the use of any metadata. We conduct extensive experiments to evaluate our system in several metrics on two datasets, showing the best results in both clustering performance and topic coherence. We also provide a qualitative analysis of the effectiveness of our system and thus justify its applications.

Twitter Opinion summarisation We present a system for providing interactive visualisation of topic-entity sentiments in chronological order while providing fine-grained summaries to give insights into the underlying reasons. We are the first to apply state-of-the-art abstractive summarisation model used for traditional news articles, to tweets. We provide insightful evaluation on the feasibility of cross-medium abstractive summarisation with no target-medium training resources. Experiments are conducted for both event summarisation and opinion summarisation, with and without pre-training. We believe our results and analysis are valuable to the summarisation research community.

1.3 Publications

For each research chapter we list on which publication(s) it is based.

Chapter 3: The first half of this chapter is based on Bo Wang, Arkaitz Zubiaga, Maria Liakata and Rob Procter [36], "Making the most of tweet-inherent features for social spam detection on twitter". *5th Workshop on Making Sense of Microposts (#Microposts2015), WWW 2015.* The design of the algorithm, the experiments and paper write-up were mostly contributed by Bo Wang.

The second half of chapter is based on Bo Wang, Maria Liakata, Arkaitz Zubiaga, Rob Procter and Eric Jensen [37], "SMILE: Twitter emotion classification using domain adaptation". *4th Workshop on Sentiment Analysis where AI meets Psychology, IJCAI 2016.* The design of the algorithm, the experiments and paper write-up were mostly contributed by Bo Wang.

Chapter 4: This chapter is based on Bo Wang, Maria Liakata, Arkaitz Zubiaga and Rob Procter [38], "TDParse: Multi-target-specific sentiment recognition on Twitter". *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2017).* The design of the algorithm, the experiments and paper write-up were mostly contributed by Bo Wang.

A small part of this chapter is also based on Richard Townsend, Adam Tsakalidis, Yiwei Zhou, Bo Wang, Maria Liakata, Arkaitz Zubiaga, Alexandra I Cristea and Rob Procter [39], "WarwickDCS: From phrase-based to target-specific sentiment recognition". *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015).* The design of the algorithm and paper write-up were partly contributed by Bo Wang.

Chapter 5: This chapter is based on Bo Wang, Maria Liakata, Arkaitz Zubiaga and Rob Procter [40], "A Hierarchical Topic Modelling Approach for Tweet Clustering". *9th International Conference on Social Informatics (SocInfo 2017).* The design of the algorithm, the experiments and paper write-up were mostly contributed by Bo Wang.

Chapter 6: This chapter is partially based on Bo Wang, Maria Liakata, Adam Tsakalidis, Spiros Georgakopoulos Kolaitis, Symeon Papadopoulos, Lazaros Apostolidis, Arkaitz Zubiaga, Rob Procter and Yiannis Kompatsiaris [41], "TOTEMSS: Topic-based, Temporal Sentiment Summarisation for Twitter". *Proceedings of the 8th International Joint Conference on Natural Language*

Processing (IJCNLP 2017). The design of the algorithm and paper write-up were mostly contributed by Bo Wang.

Work on other publications also contributed to the thesis, albeit indirectly:

- Arkaitz Zubiaga, Alex Voss, Rob Procter, Maria Liakata, Bo Wang and Adam Tsakalidis [42], "Towards real-time, country-level location classification of worldwide tweets". *Published in IEEE Transactions on Knowledge and Data Engineering, 2017.*

- Arkaitz Zubiaga, Bo Wang, Maria Liakata and Rob Procter [43], "Political Homophily in Independence Movements: Analysing and Classifying Social Media Users by National Identity". *Published in IEEE Intelligent Systems, 2018.*

CHAPTER 2

Background

In this chapter, we provide background and literature review for our research work included in this thesis. We start by introducing two approaches of tackling Twitter spam in Section 2.1, namely social spammer detection and spam detection, and the corresponding related work for each approach. In Section 2.2 we detail previous work on sentiment analysis for Chapter 3 and 4. Specifically, Section 2.2.1 surveys existing work for sentiment analysis on social media such as Twitter. Section 2.2.2 gives the background material on cross-domain sentiment classification using domain adaptation. In Section 2.2.3 and 2.2.4 we discuss related work on target-specific sentiment and aspect-level sentiment classification, as well as the similarity and difference between the two. This gives an overview of Twitter sentiment analysis as a whole and different domains within this research area, preparing for Chapter 4.

In order to fully understand the sentiment expressed towards a particular target, we need to group such opinion into different clusters and generate summary for each cluster, as described in Chapter 1. Existing work on topical clustering of tweets is discussed in Section 2.3, where two different approaches are reviewed. Because our proposed system for tweet clustering is based on topic models, we also review different evaluation metrics for topic modelling. Finally, we conduct litera-

ture review for automatic summarisation, which serves as the background material for Chapter 6. We describe the state-of-art methods in two approaches, namely extractive summarisation and abstractive summarisation.

2.1 Spam Detection on Twitter

The detection of spam has now been studied for more than a decade since email spam [44]. In the context of email messages, spam has been widely defined as "unsolicited bulk email" [45]. The term "spam" has then been extended to other contexts, including "social spam" in the context of social media. Similarly, social spam can be defined as the "unwanted content that appears in online social networks". It is, after all, the noise produced by users who express a different behavior from what the system is intended for, and has the goal of grabbing attention by exploiting the social networks' characteristics, including for instance the injection of unrelated tweet content in timely topics, sharing malicious links or fraudulent information. Social spam hence can appear in many different forms, which poses another challenge of having to identify very different types of noise for social spam detection systems.

There are two ways of dealing with Twitter spam, namely spammer detection and directly detecting spam tweets. In the following sections we describe relevant research on both approaches.

2.1.1 Social Spammer Detection

Most of the previous work in the area has focused on the detection of users that produce spam content (i.e., spammers), using historical or network features of the user rather than information inherent to the tweet. Early work by [27], [46] and [28] put together a set of different features that can be obtained by looking at a user's previous behaviour. These include some aggregated statistics from a user's past tweets such as average number of hashtags, average number of URL links and

average number of user mentions that appear in their tweets. They combine these with other non-historical features, such as number of followers, number of followings and age of the account, which can be obtained from a user's basic metadata, also inherent to each tweet they post. Some of these features, such as the number of followers, can be gamed by purchasing additional followers to make the user look like a regular user account.

Lee et al. [47] and Yang et al. [48] employed different techniques for collecting data that includes spam and performed comprehensive studies of the spammers' behaviour. They both relied on the tweets posted in the past by the users and their social networks, such as tweeting rate, following rate, percentage of bidirectional friends and local clustering coefficient of its network graph, aiming to combat spammers' evasion tactics as these features are difficult or costly to simulate. Ferrara et al. [49] used network, user, friends, timing, content and sentiment features for detecting Twitter bots, their performance evaluation is based on the social honeypots dataset from [47]. Miller et al. [50] treats spammer detection as an anomaly detection problem as clustering algorithms are proposed and such clustering model is built on normal Twitter users with outliers being treated as spammers. They also propose using 95 uni-gram counts along with user profile attributes as features. The sets of features utilised in the above works require the collection of historical and network data for each user, which do not meet the requirements of our scenario for spam detection.

2.1.2 Social Spam Detection

Few studies have addressed the problem of spam detection. Santos et al. [51] investigated two different approaches, namely compression-based text classification algorithms (i.e. Dynamic Markov compression and Prediction by partial matching) and using "bag of words" language model (also known as uni-gram language model) for detecting spam tweets. Martinez-Romo and Araujo [29] applied Kullback-Leibler

Divergence (KLD) and examined the difference of language used in a set of tweets related to a trending topic, suspicious tweets (i.e. tweets that link to a web page) and the page linked by the suspicious tweets. These language divergence measures were used as their features for the classification. They used several URL blacklists for identifying spam tweets from their crawled dataset, therefore each one of their labelled spam tweets contains a URL link, and is not able to identify other types of spam tweets. In our studies we have investigated and evaluated the discriminative power of four feature sets on two Twitter datasets (which were previously in [47] and [48]) using five different classifiers. We examine the suitability of each of the features for the spam classification purposes. Comparing to [29] our proposed system described in Chapter 3 is able to detect most known types of spam tweet irrespective of having a link or not. Also our system does not have to analyze a set of tweets relating to each topic (which [29] did to create part of their proposed features) or external web page linked by each suspicious tweet, therefore its computation cost does not increase dramatically when applied for mass spam detection with potentially many different topics in the data stream.

The few works that have dealt with spam detection are mostly limited in terms of the sets of features that they studied, and the experiments have been only conducted in a single dataset (except in the case of [29], where very limited evaluation was conducted on a new and smaller set of tweets), which does not allow for generalisability of the results. In Chapter 3, we evaluate a wide range of tweet-inherent features (namely user, content, n-gram and sentiment features) over two different datasets, obtained from [47] and [48] and with more than 10,000 tweets each, for the task of spam detection. The two datasets were collected using completely different approaches (namely deploying social honeypots for attracting spammers; and checking malicious URL links), which helps us learn more about the nature of social spam and further validate the results of different spam detection systems.

2.2 Sentiment Analysis

In recent years sentiment analysis has become ever more popular, with over 7,000 articles written on the topic [52], applications ranging from box office [53] and election prediction [13], to detecting emotions in suicide notes [54]. Despite the popularity and commercial adoption of sentiment analysis especially for social media, a number of challenges in this field are still yet to be solved. For example as reviewed in [55], sentiment is domain specific and the meaning of words changes depending on the context they are used in. Another important challenging task is to distinguish and detect sentiment expressed towards different target entities appearing in the same text. Here we address the cross-domain challenge in Chapter 3 and target-specific sentiment analysis in Chapter 4.

In this section, we start with a general review on sentiment analysis research on social media such as Twitter. Then we discuss relevant work on domain adaptation for sentiment classification. At last, we zoom in on two related research areas for entity-level sentiment analysis, namely target-specific sentiment recognition on Twitter (described in Section 2.2.3) and aspect-level sentiment classification on reviews (described in Section 2.2.4).

2.2.1 Sentiment Analysis on Social Media

In an earlier study, Wang et al. [16] develop a real-time large scale (collected over 36 million tweets) political sentiment analysis system achieving 59% in accuracy on four-category classification of negative, positive, neutral or unsure. It uses a crowdsourcing platform (Amazon Mechanical Turk) to acquire sentiment annotated training data, and built simple Naive Bayes model on unigram features. Machine learning-based approaches require creating a model by training the classifier with a large set of sentiment annotated training data, which is labor-intensive to acquire. Lexicon-based approaches on the other hand (e.g. [56]), use sentiment dictio-

nary to determine opinion orientations, but because of the noisy nature of tweets, lexicon-based approaches suffer from low recall problem. Zhang et al. [57] propose combining lexicon-based and learning-based methods, using lexicon to label tweets as training data for learning a Support Vector Machine model. Go et al. [10] introduce a distant-supervision approach using emoticons as noisy labels, producing large amount of training data. In SemEval-2013 and 2014 Twitter sentiment analysis competitions the best performing systems [58, 59] both use rich lexical and manually engineered features. However, the development of lexica can be time consuming and is domain specific. An interesting study by Tang et al. [60] propose a joint learning framework for tweet-level sentiment classification. The framework has a tweet segmentation model that is updated at each training iteration by verifying predicted sentiment of each segmentation candidate, top ranked candidates are selected in turn for training the sentiment classifier. However, learning such sentiment-specific segmentation model is a difficult task if only tweet-level sentiment information is used as the training signal[1]. Several other studies focus on incorporating additional information to further improve performance such as user background [61], topic information [62], user bias towards a topic [63], or social relations [64, 17].

Deep learning has also been applied in the field of Twitter sentiment analysis. Severyn and Moschitti [65] use a convolutional neural network (CNN) for predicting polarities at both tweet and phrase levels, using distant-supervision data for network initialisation. Ren et al. [66] propose a context-based neural network model incorporating contextualized features from relevant tweets (to the target tweet) in the form of word embedding vectors. A recent work by Yang et al. [67] proposes an attention-based neural architecture, incorporating the author's position in the social network, which makes it possible to induce personalized classifiers. These supervised deep learning approaches usually require large amount of training data, which is not always available. In two studies by Tang et al. [68, 69], sentiment-specific word em-

[1]This is observed by evaluating its system on various data sets.

beddings are learnt and used as features for identification of tweet-level sentiment achieving good performance. Similar to [39], we adopt a hybrid approach which incorporates rich and diverse set of features including lexica, n-gram, cluster, and word embeddings (including the one proposed by [69]), to train a SVM classifier as a target-independent baseline model to be evaluated and compared in Section 4.4.1. One future direction of this area is to have a more explicit model of morphology than just character/sub-word/word composition, which will give us the morphologically-aware word representations that can be used for modelling sentiment in a sentence.

Support Vector Machine

Support Vector Machines (SVM) [70] are a supervised machine learning algorithm used for classification, regression and other learning tasks. It maps data points in d-dimensional space, and tries to find a $(d-1)$-dimensional hyperplane (or a set of hyperplanes) that separates the data points into two classes and the distance from the hyperplane to the nearest training data point on each side is maximised. If the training data is not linearly separable, SVM can be extended by using the hinge loss function: $\max(0, 1 - y_i(\mathbf{w} \cdot \mathbf{x}_i - b))$. If the data is on the wrong side of the margin, the function's value is proportional to the distance from the margin. Therefore computing the soft-margin SVM amounts to minimising:

$$\left[\frac{1}{n}\sum_{i=1}^{n}\max(0, 1 - y_i(\mathbf{w} \cdot \mathbf{x}_i - b))\right] + \lambda\|\mathbf{w}\|^2 \tag{2.1}$$

where λ determines the tradeoff between the margin-size and making sure $\mathbf{x}_i$ lie on the correct side of the margin. Equation .2.1 can be rewritten as the primal

optimisation problem as below:

$$\begin{aligned} \min_{textbfw,b,\xi} \quad & \frac{1}{2}(\mathbf{w})^T\mathbf{w} + C\sum_{i=1}^{N}\xi_i \\ \text{s.t.} \quad & y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0, \text{for all } i \end{aligned} \tag{2.2}$$

where $\phi(\mathbf{x}_i)$ maps $\mathbf{x}_i$ into a higher-dimensional space, and C is the regularisation parameter. Due to the possible high dimensionality of $\mathbf{w}$, the Lagrangian dual of the above problem is usually being solved:

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}(\boldsymbol{\alpha})^T Q\boldsymbol{\alpha} - \mathbf{e}^T\boldsymbol{\alpha} \\ \text{s.t.} \quad & \mathbf{y}^T\boldsymbol{\alpha} = 0, \\ & 0 \leq \alpha_i \leq C, \text{for all } i \end{aligned} \tag{2.3}$$

where $\mathbf{e} = [1, ..., 1]^T$, Q is the positive semidefinite matrix, $Q_{i,j} \equiv y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$, and $K(\mathbf{x}_i, \mathbf{x}_j) \equiv \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$ is the kernel function[2]. The optimal $\mathbf{w}$ satisfies:

$$\mathbf{w} = \sum_{i=1}^{n} y_i\alpha_i\phi(\mathbf{x}_i)$$

and the decision function becomes:

$$\text{sgn}(\mathbf{w}^T\phi(\mathbf{x}) + b) = \text{sgn}(\sum_{i=1}^{n} y_i\alpha_i K(\mathbf{x}_i, \mathbf{x}) + b)$$

All the weights, support vectors and other information such as kernel parameters are stored in the model for prediction.

[2]This allows the algorithm to fit the maximum-margin hyperplane in the transformed feature space.

Neural Networks

Neural networks provide powerful tools for modeling language, and have become the state-of-the-art models for many NLP tasks in the recent years. There are two main deep neural network architectures: **convolutional neural network (CNN)** [71] and **recurrent neural network (RNN)** [72]. **CNNs** generally consist of an input layer, one or more convolution and max pooling layers, the fully connected layer, and loss layer. Consider a sequence of words $\mathbf{x} = x_1, x_2, ..., x_n$, each with its corresponding embedding representation $v(x_i)$. A one-dimensional convolution layer of width k works by moving a sliding window of the same size over the sentence, and applying the filter to the sequence in the window. The filter function is usually a linear transformation followed by a non-linear activation function. Let $\mathbf{c}_i \in \mathbb{R}^{kd}$ be the concatenated vector of the sliding window containing k inputs $x_i, x_{i+1}, ..., x_{i+k-1}$ and m is the total number of these windows depending on whether narrow or wide convolution is used. When $i < 1$ or $i > n$, the embedding representations for x_i are zero padded. The result of the convolution layer is m vectors $\mathbf{p}_1, ..., \mathbf{p}_m$, $\mathbf{p}_i \in \mathbb{R}^d$:

$$\mathbf{p}_i = g\left(\mathbf{W} \cdot \mathbf{c}_i + \mathbf{b}\right)$$

where $\mathbf{W} \in \mathbb{R}^{d \times wd}$ is the convolution weights and $\mathbf{b} \in \mathbb{R}^d$ is the bias. g is an activation function to increase non-linearity, its common choices are hyperbolic tangent function `tanh`, sigmoid function `sigmoid` and rectified linear unit `ReLU`. $\mathbf{p}_1, ..., \mathbf{p}_m$ are then combined using a pooling layer such as a max-pooling operation to extract the most salient information across window positions. The resulting vector from the pooling layer is then fed into the downstream network layers including a loss layer for calculating the loss with respect to the downstream task.

The word order sensitivity captured in convolutional networks is restricted to mostly local patterns. **RNNs** recursively take a state vector $\mathbf{s}_i$ and an input vector $\mathbf{x}_{i+1}$, and result in a new state vector $\mathbf{s}_{i+1}$. It provides a framework for modeling

sequence based on the entire history of states without resorting to the Markov assumption, which is traditionally used in language models. Gating mechanisms have been developed and widely used to alleviate the vanishing gradient problem that exists in standard RNNs. **Gated recurrent unit (GRU)** [73] and **long short-term memory (LSTM)** [74] are two types of RNNs using different gating mechanism. **GRU** has a update gate and a reset gate, which control what should be passed to the output. It models input $\mathbf{x}_t$ as follows:

$$\mathbf{z}_t = \sigma_g(\mathbf{W}_z\mathbf{x}_t + \mathbf{U}_z\mathbf{h}_{t-1} + \mathbf{b}_z) \tag{2.4}$$

$$\mathbf{r}_t = \sigma_g(\mathbf{W}_r\mathbf{x}_t + \mathbf{U}_r\mathbf{h}_{t-1} + \mathbf{b}_r) \tag{2.5}$$

$$\mathbf{h}_t = (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \sigma_h(\mathbf{W}_h\mathbf{x}_t + \mathbf{U}_h(\mathbf{r}_t \odot \mathbf{h}_{t-1}) + \mathbf{b}_h) \tag{2.6}$$

where $\mathbf{x}_t \in \mathbb{R}^d$ is the input at time step t, $\mathbf{h}_t \in \mathbb{R}^h$ is the hidden state encoding all the inputs preceding t, $\mathbf{z}_t$ is the update gate, $\mathbf{r}_t$ is the reset gate, σ_g is a `sigmoid` function, σ_h is a `tanh` function, and $\mathbf{U}$, $\mathbf{W}$ and $\mathbf{b}$ are the parameters.

In a standard **LSTM** network, each of its units is composed of a cell, an input gate, an output gate and a forget gate. LSTM models $\mathbf{x}_t$ as follows:

$$\mathbf{f}_t = \sigma_g(\mathbf{W}_f\mathbf{x}_t + \mathbf{U}_f\mathbf{h}_{t-1} + \mathbf{b}_f) \tag{2.7}$$

$$\mathbf{i}_t = \sigma_g(\mathbf{W}_i\mathbf{x}_t + \mathbf{U}_i\mathbf{h}_{t-1} + \mathbf{b}_i) \tag{2.8}$$

$$\mathbf{o}_t = \sigma_g(\mathbf{W}_o\mathbf{x}_t + \mathbf{U}_o\mathbf{h}_{t-1} + \mathbf{b}_o) \tag{2.9}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \sigma_c(\mathbf{W}_c\mathbf{x}_t + \mathbf{U}_c\mathbf{h}_{t-1} + \mathbf{b}_c) \tag{2.10}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \sigma_h(\mathbf{c}_t) \tag{2.11}$$

where the input gate $\mathbf{i}_t \in \mathbb{R}^h$, forget gate $\mathbf{f}_t \in \mathbb{R}^h$ and output gate $\mathbf{o}_t \in \mathbb{R}^h$ are generated by applying sigmoid function over the input vector $\mathbf{x}_t \in \mathbb{R}^d$ and preceding hidden state vector $\mathbf{h}_{t-1} \in \mathbb{R}^h$. In order to generate the hidden state at current time step t, it first applies σ_c (i.e. a `tanh` function) over $\mathbf{x}_t$ and $\mathbf{h}_{t-1}$, then combines it

with $\mathbf{c}_{t-1}$ using input gate $\mathbf{i}_t$ and forget gate $\mathbf{f}_t$ to get an updated cell state $\mathbf{c}_t \in \mathbb{R}^h$. The final hidden state $\mathbf{h}_t$ is generated by multiplying the output gate vector $\mathbf{o}_t$ with $\sigma_h(\mathbf{c}_t)$, where σ_h is another `tanh` function.

Word Embedding

Word embeddings, or distributional semantic models, are based on the idea that contextual information constitutes a viable representation of linguistic items such as words. While topic models (described later in this chapter) use documents as contexts, neural language models and distributional semantic models instead use words as contexts. Collobert and Weston [75] showed word embeddings trained on a sufficiently large dataset to be useful for downstream tasks, and since then it has been used in various NLP applications such as sentiment analysis, named entity recognition, parsing, tagging and machine translation.

The main differences among the word embedding models are computational complexity and training objective. The two most popular word embedding models, `word2vec` and `GloVe`, both encode general semantic relationships. Mikolov et al. [76] proposed `word2vec` with two architectures: Continuous bag-of-words (CBOW) and Skip-gram. These models are shallow, two-layer neural networks that are trained to reconstruct linguistic contexts of words. While a classic language model aims to predict each word based on its previous words, CBOW uses both n words before and after the target word w_t for prediction. The objective function of CBOW is shown below:

$$J_\theta = \frac{1}{T}\sum_{t=1}^{T} \log p(w_t \mid w_{t-n}, \cdots, w_{t-1}, w_{t+1}, \cdots, w_{t+n}) \tag{2.12}$$

Instead of predicting the target word based on context, skip-gram uses the target word as an input to a log-linear classifier with continuous projection layer, and

predict its surrounding words. It has the following objective function:

$$J_\theta = \frac{1}{T}\sum_{t=1}^{T} \sum_{-n\leq j\leq n, \neq 0} \log p(w_{t+j} \mid w_t) \tag{2.13}$$

To mitigate the cost of computing the final softmax layer, Mikolov et al. [77] introduced negative sampling algorithm and the subsampling of frequent words, showing much more computationally efficient model architecture. They also proposed an alternative to the sampling approach, which uses a binary Huffman tree for their hierarchical softmax (an approximation to full softmax).

`GloVe` [78] is a count-based model that learns word vectors by essentially performing dimensionality reduction on the word co-occurrence counts matrix X. This large matrix is factorised to yield a lower-dimensional matrix, where each row is a vector representation for a word. Pennington et al. [78] encode the information present in the ratios of word co-occurrence probabilities instead of the probabilities themselves. To achieve this, they proposed a weighted least squares objective J:

$$J = \sum_{i,j=1}^{V} f\left(X_{ij}\right)\left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij}\right)^2 \tag{2.14}$$

where w_i and b_i are word vector and bias respectively of word i, while $\tilde{w}_j$ and $\tilde{b}_j$ are context word vector and bias for word j. X_{ij} is the number of times word i occurs in the context of word j. The weighting function f is used to prevent learning only from extremely common word pairs, and it is defined in `GloVe` as the following:

$$f(x) \begin{cases} \left(\frac{x}{x_{max}}\right)^\alpha & \text{if } x < x_{max}, \\ 1 \quad \text{otherwise}\,. \end{cases} \tag{2.15}$$

2.2.2 Cross-domain Sentiment Classification

Most existing domain adaptation approaches can be classified into two categories: feature-based adaptation and instance-based adaptation. The former seeks to construct new adaptive feature representations that reduce the difference between domains, while the latter aims to sample and re-weight source domain training data for use in classification within the target domain.

With respect to feature domain adaptation, [79] applied structural correspondence learning (SCL) algorithm for cross-domain sentiment classification. SCL chooses a set of *pivot features* that frequently occur in both domains and have highest mutual information to the domain labels, and uses these pivot features to align other features by training N linear predictors. Finally it computes singular value decomposition (SVD) to construct low-dimensional features to improve its classification performance. A small amount of target domain labelled data is used to learn to deal with misaligned features from SCL. [80] found that SCL did not work well for cross-domain adaptation of sentiment on Twitter due to the lack of mutual information across the Twitter domains and uses subjective proportions as a backoff adaptation approach. [81] proposed to construct a bipartite graph from a co-occurrence matrix between domain-independent and domain specific features to reduce the gap between different domains and use spectral clustering for feature alignment. The resulting clusters are used to represent data examples and train sentiment classifiers. They used mutual information between features and domains to classify domain-independent and domain specific features, but in practice this also introduces mis-classification errors. [82] describes a cross-domain sentiment classification approach using an automatically created sentiment sensitive thesaurus. Such a thesaurus is constructed by computing the point-wise mutual information between a lexical element u and a feature that can be either a sentiment feature or another lexical element that co-occurs with u in the training data, as well as relatedness be-

tween two lexical elements. Therefore common domain-independent words are used as pivots that transfer information from one domain to another. The problem with these feature adaptation approaches is that they try to connect domain-dependent features to known or common features under the assumption that parallel sentiment words exist in different domains, which is not necessarily applicable to various topics in tweets [83].

When it comes to instance adaptation, [84] proposes an instance weighting framework that prunes "misleading" instances and approximates the distribution of instances in the target domain. Their experiments show that by adding some labelled target domain instances and assigning higher weights to them performs better than either removing "misleading" source domain instances using a small number of labelled target domain data or bootstrapping unlabelled target instances. [85] adapts the source domain training data to the target domain based on a logistic approximation. [31] learns different classifiers on different sets of features and combines them in an ensemble model. Such an ensemble model is then applied to part of the target domain test data to create new training data (i.e. documents for which different classifiers had the same predictions). We include this ensemble method as one of our baseline approaches for evaluation and comparison.

Except for [31] and [80], none of the above studies carry out cross-domain sentiment classification for Twitter data, which has been proven more challenging. [30] and [86] studied cross-medium sentiment classification, which transfers sentiment classifier trained on blogs or reviews to tweets. [87] examined whether the observation about domain-dependent models improving sentiment classification of reviews also applies to tweets. They found such models to achieve significantly better performance than domain-independent models for *some* topics. [83] implements a multi-class semi-supervised Support Vector Machines (S3VMs) model that performs co-training on both textual and non-textual features (e.g. temporal features) for sentiment classification on tweets. In order to make their model adaptive to differ-

ent topics, confident unlabelled target-domain data are selected and topic-adaptive sentiment words are used as additional lexicon features. Ruder et al. [88] review different strategies to select training data from multiple sources for domain adaptation for sentiment analysis, based on feature representation, similarity metrics, and the level of the selection. They find both selecting the most similar domain and subsets outperform instance-level selection. A Bayesian Optimisation based data selection approach is also proposed by the same author [89].

More recently, several studies have developed deep learning models for domain adaptation. [90] is the first to propose learning a unified feature representation for different domains, under the intuition that deep learning algorithms learn intermediate concepts (between raw input and target) and these intermediate concepts could yield better transfer across domains. [91] use two parameter-sharing memory networks with attention for automatically capturing important sentiment words that are shared in both domains (i.e. pivots), where one network is for sentiment classification and the other is for domain classification. The two networks are trained jointly. By augmenting the skip-gram objective with a regularisation term, [92] learns cross domain word embeddings that is shown to achieve good performance in cross-domain sentiment classification. However, both source and target domains are reviews from different sites. [93] uses emoji tweets for pretraining a model that can be used in a new task with fine-tuning. Their proposed transfer learning approach sequentially unfreezes and fine-tines each layer, then lastly the entire model is trained with all layers. The authors evaluated on 3 tasks including emotion analysis, however, only 'Fear', 'Joy' and 'Sadness' are evaluated as the remaining emotions rarely occurred in the observations.

In contrast with most cross-domain sentiment classification works, we use a SVM-based approach proposed in [94], which directly adapts existing classifiers trained on general-domain corpora. We believe this is more efficient and flexible [95] for our task. We evaluate on a set of manually annotated tweets about cul-

tural experiences in museums and conduct a finer-grained classification of emotions conveyed (i.e. *anger*, *disgust*, *happiness*, *surprise* and *sadness*).

2.2.3 Target-dependent Sentiment Recognition

The 2015 Semeval challenge introduced a task on target-dependent Twitter sentiment [96] which most systems [97, 98] treated in the same way as tweet level sentiment. The best performing system in the 2016 Semeval Twitter challenge substask B [99], named Tweester, also performs on tweet level sentiment classification. This is unsurprising since tweets in both tasks only contain a single predefined target entity and as a result often a tweet-level approach is sufficient. An exception to tweet level approaches for this task [39], trained a SVM classifier for tweet segmentation, then used a phrase-based sentiment classifier for assigning sentiment around the target and returning the majority sentiment. The Semeval aspect-based sentiment analysis task [100, 101] aims to identify sentiment towards entity-attribute pairs in customer reviews. This differs from the target-dependent task in the following way: both the entities and attributes are limited to a predefined inventory of limited size; they are aspect categories reflected in the reviews rather than specific targets, while each review only has one target entity, e.g. a laptop or a restaurant. Also sentiment classification in formal text such as product reviews is very different from that in tweets. Recently Vargas et al. [20] analysed the differences between the overall and target-dependent sentiment of tweets for three events containing 30 targets, showing many significant differences between the corresponding overall and target-dependent sentiment labels, thus confirming that these are distinct tasks.

Early work tackling target-dependent sentiment in tweets [11] designed target dependent and independent features manually, relying on the syntactic parse tree and a set of grammar-based rules, and incorporating the sentiment labels of related tweets (i.e. retweets, replies and other tweets by the same users) to improve the classification performance. Recent work [1] used recursive neural networks [102] and

adaptively chose composition functions to combine child feature vectors according to their dependency type, to reflect sentiment signal propagation to the target. Their data-driven composition selection approach relies on the dependency types (generated from Stanford parser[3]) as features and a small set of rules for constructing target-dependent trees. Their manually annotated dataset contains only one target per tweet and has since been used for benchmarking by several subsequent studies [103, 104, 105]. Vo and Zhang [103] exploit the left and right context around a target in a tweet and combine low-dimensional embedding features from both contexts and the full tweet using a number of different pooling functions. Despite not fully capturing semantic and syntactic information given the target entity, they show a much better performance than Dong et al. [1], indicating useful signals in relation to the target can be drawn from such context representation. Both Tang et al. [104] and Zhang et al. [105] adopt and integrate left-right target-dependent context into their recurrent neural network (RNN) respectively. While Tang et al [104] propose two long short-term memory (LSTM) models showing competitive performance to Vo and Zhang [103], Zhang et al [105] design a gated neural network layer between the left and right context in a deep neural network structure but require a combination of three corpora for training and evaluation. Results show that conventional neural network models like LSTM are incapable of explicitly capturing important context information of a target [106]. Tang et al. [104] also experiment with adding attention mechanism for LSTM but fail to achieve competitive results possibly due to the small training corpus.

Going beyond the existing work, in Chapter 4 we introduce the more challenging task of classifying sentiment towards multiple target entities within a tweet. We show the tweet level approach that many sentiment systems adopted in both Semeval challenges, fail to capture all target-sentiments in a multi-target scenario (Section 4.5.1).

[3]https://nlp.stanford.edu/software/lex-parser.shtml

2.2.4 Aspect-level Sentiment Classification

The task of classifying target-specific sentiment is related to aspect-level sentiment analysis, which is mostly analysed on product reviews. Its goal is to identify sentiment polarity expressed towards aspect categories [100, 101]. To capture such aspect-level sentiment on reviews, Lakkaraju et al. [107] use recursive neural tensor network (RNTN) proposed by [108] to learn representations of words and parses of phrases and sentences containing the words. The features contribute to an objective function relating features of the words and phrase constituents to sentiment labels which the system seeks to optimise. Nguyen et al. [109] transform the dependency parsing trees into target-dependent binary phrase dependency trees in order to learn to classify aspect-level sentiment in the restaurant reviews. One potential problem of recursive neural networks is having to binarise syntactic trees and resulting in long propagation paths. This may lead to information loss or commonly known as vanishing gradient problem [110]. Identifying sentiment for product reviews is different from that of tweets, as not only in reviews if any sentiment is expressed in a sentence containing a target it is highly likely the sentiment is towards such target as argued in [11], but also such compositionality from [107] is more difficult to achieve and requires a dependency parser trained specifically for tweets (such as [111], which does not provide sufficient dependency type information). One way to potentially alleviate the latter problem is to have many different parses and learn to choose or combine them, as suggested by Le et al. [112].

In [109] the authors achieve good performance for review sentences containing one or two aspects with all aspects in the sentence having the same sentiment type. They show sentences mentioning three aspects with different sentiment types to be the most difficult case with the best 48.13 in F_1 score, comparing to 62.21 for all sentences. In Chapter 4 we show our new multi-target corpus has 1649 out of 4077 tweets (40%) having three or more targets with different sentiment categories thus

posing a challenging task.

2.3 Tweet Clustering

A topical clustering system aims to group a set of tweets, usually posted in the same period of time, to a number of clusters, with each cluster representing a meaningful topic. This is also tightly related to topic detection described by the task of Topic Detection and Tracking (TDT) [113] as extracting event-based[4] topics from a corpus of textual data. According to Aiello et al. [22], methodologically, existing general-purpose topic detection fit into two main categories: 1), *document-pivot* approaches where topics are represented by such document clusters; 2), *feature or term-pivot* methods where the most important terms are clustered and a topic is represented by a cluster of terms instead. In this section we review the recent developments on tweet clustering and Twitter topic detection from the aforementioned two perspectives.

2.3.1 Document-Pivot Methods

Document-pivot approaches usually involve encoding documents in some vector representations that can be either sparse one-hot vectors or dense embedding matrices. Then similarity metrics are used to measure and group similar documents together as clusters. An early work on breaking news detection in Twitter [114] uses bag-of-words for tweet representation and textual similarity between tweets is compared using boosted tf-idf[5]. Rosa et al. [21] find traditional unsupervised methods to produce incoherent topical clusters and suggest the superiority of supervised models using hashtags as training labels. Similar approaches can be found in many of the Twitter event detection literature where online clustering is adopted for incremental and efficient tweet clustering. To alleviate the cluster over-fragmentation issue that exists in the online clustering approach, these studies usually perform a second

[4]Here an "event" is defined as some unique thing that happens at some point in time.

[5]The similarity score is based on tf-idf but boosted by proper noun terms. The Stanford Named Entity Recogniser is used for the classification of proper noun terms.

stage of offline clustering [115] [116] or classification [117]. The winner system [118] of the 2014 SNOW Data Challenge[6], uses a method based on aggressive tweet/term filtering combined with two-stage hierarchical clustering and ranking. In terms of clustering algorithm Rangrej et al. [119] compare K-means, a Singular Value Decomposition (SVD) based method and Affinity Propagation, they find the graph-based Affinity Propagation method to be the most effective in clustering tweets. Tweet clustering is also studied in the First Story Detection research area, where besides the use of tf-idf term representation and cosine similarity, Locality Sensitive Hashing (LSH) is adopted to approximate and speed up the nearest neighbor search process [120].

Many of the aforementioned studies focus on the problem of online clustering of a stream of tweets. They use an incremental clustering framework, which assigns newly arriving tweets to the existing clusters. In Chapter 5, we primarily focus on how to best cluster a static collection of tweets, which are set to be performed efficiently offline, possibly at the end of each day.

Using tf-idf feature vectors as tweet representation has the issue of sparsity. Noticeably Tsur et al. [121] concatenate tweets mentioning the same hashtags into virtual documents, and perform clustering on the virtual documents instead. This way it alleviates the sparsity problem of tweets. Another challenge of tweet clustering is how to go beyond the limitation of bag-of-words representation and encode tweets in some vector embeddings that enables the semantic similarity matching in tweet content. This aims to avoid clustering tweets based on language similarity rather than topical coherence, as mentioned in [21]. Ganesh et al. [122] compare various tweet representations generated by supervised and unsupervised learning methods, over a set of tweet-specific elementary property classification tasks such as predicting slang words or reply time, in trying to show the basic characteristics of different tweet representations. Their results show Skip-Thought Vectors [123] to be

[6] `http://www.snow-workshop.org/2017/challenge/`

good for most of the social tasks including in predicting whether a tweet is a reply, due to its inter-sentential features learnt from predicting surrounding sentences as well as the recurrent structure in both the encoder and decoder. Vakulenko at al. [124] employ a character-based tweet embedding method, named Tweet2Vec [125], along with hierarchical clustering for the task of clustering tweets. They demonstrate to outperform [118] for the 2014 SNOW breaking news detection corpus. Interestingly Arora et al. [126] propose a simple and unsupervised approach to sentence embedding based on the weighted average of word vectors in the sentence and "*common component removal*", reporting surprisingly good performances on 22 textual similarity data sets, including a Twitter corpus.

2.3.2 Term-Pivot Methods

Feature-pivot methods are commonly based on the analysis of associations between terms, and are closely related to topic modelling. Conventional topic models such as Latent Dirichlet Allocation (LDA) [127] have shown great success in various Natural Language Processing (NLP) tasks for discovering the latent topics that occur in long and structured text documents. Due to the limited word co-occurrence information in short texts, conventional topic models perform much worse for social media microposts such as tweets as demonstrated by Rosa et al. [21]. Here we review the recent developments on Twitter topic modelling and how to tackle the sparse and noisy nature of tweets.

Earlier studies try to utilise external knowledge such as Wikipedia [128] to improve topic modelling on short texts. This requires a large text corpus which may have a domain issue for the task at hand. Since then four approaches have been studied in the literature to adapt conventional topic models for short texts such as tweets:

1) Directly model the generation of word co-occurrence pattern in the whole corpus (rather than at document-level) based on biterms, where a biterm denotes an

unordered word-pair co-occurring in a tweet, as demonstrated by Yan et al. [129]. Since it does not model the document generation process, the topic distribution of each document cannot be directly obtained and instead it is derived based on the topic proportions of biterms of the document.

2) Apply a document pooling strategy, to aggregate tweets to a number of virtual documents, based on authors [130], hashtags [131], conversation [132] or other metadata [133] such as timestamps and named entities. This strategy helps to overcome the limited context information in tweets, but pooling by such metadata can potentially have adverse effect on the subsequent topic modelling.

3) [134] proposed a simple topic model, named Dirichlet Multinomial Mixture (DMM) model. The DMM model has since then been used in many Twitter topic modelling studies for alleviating the data sparsity problem and reported to give more coherent topics [135, 23, 136, 137], given that its underlying assumptions are reasonable for short texts.

4) Complement topic models which use the global word collocation patterns in the same document/tweet, with word embeddings that exploit the local word collocation patterns within a context window. [138] extend LDA and DMM to incorporate word embeddings as latent features. Such latent feature component is integrated with its original topic-word Dirichlet multinomial component. [137] propose to incorporate word embeddings through the generalised *Pólya urn* model in topic inference. [139] propose to infer topics via document-level co-occurrence patterns of latent concepts instead of words themselves. All of these approaches aim to improve topic coherence by connecting semantically related words to overcome the short length of tweets.

Besides the topic-model-based approaches, [22] proposed a term clustering method, named BNgram, where the distance between terms is defined by the proportion of tweets in which they co-occur. They found that although this method achieves good topic recall, it is the most effective only when the fixed number of

topics is set to be very small.

In Chapter 5, we present a comparative study on both topic modelling and document clustering approaches over two datasets, namely a first story detection corpus [2] and a large-scale event detection corpus covering over 500 events [140]. Our proposed two-stage topic modelling system adopts three of the four strategies mentioned above, achieving not only the best performance measured in document clustering metrics but also topic coherence for its generated topics.

Latent Dirichlet Allocation

Probabilistic topic modeling is a suite of data-driven statistical algorithms that aim to discover the main themes (i.e. topics) that pervade a large collection of documents. Since Latent Dirichlet Allocation or LDA was introduced by Blei et al. [127] in 2003, it has become the most commonly used topic model.

LDA represents each document d as a distribution θ_d over topics, where each topic t is a probability distribution ϕ_t over words W. The topic assignment for the dth document are z_d, where $z_{d,n}$ is the topic assignment for the nth word in document d. Both per-document topic distribution and per-topic word distribution have the Dirichlet prior, where α and β are parameters of the priors as presented in Figure 2.1. LDA describes the probabilistic process for generating each document as follows. For each document, it generates words by firstly randomly choosing a distribution over topics. Then for each word, it randomly choose a topic assignment and a word from the corresponding topic which is defined by distribution over the vocabulary. This generative process defines a joint probability distribution over both the observed (i.e. words in the documents) and latent variables (i.e. topics):

$$\begin{aligned} & p(\phi_{1:T}, \theta_{1:D}, z_{1:D}, w_{1:D}) \\ & = \prod_{i=1}^{T} p(\phi_i) \prod_{d=1}^{D} p(\theta_d) \left(\prod_{n=1}^{N} p(z_{d,n}|\theta_d) p(w_{d,n}|\phi_{1:T}, z_{d,n}) \right) \end{aligned} \tag{2.16}$$

where this joint distribution specifies a number of dependencies such as the topic assignment $z_{d,n}$ depends on the per-document topic proportions θ_d. These dependencies are presented in graphical model for LDA as seen in Figure 2.1. During

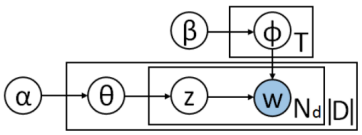


Figure 2.1: Graphical Model of Latent Dirichlet Allocation (LDA)

inference, we use the joint distribution to compute the conditional distribution (or the posterior distribution) of the latent variables (i.e. the topic structure) given the documents:

$$p(\phi_{1:T}, \theta_{1:D}, z_{1:D}|w_{1:D}) = \frac{p(\phi_{1:T}, \theta_{1:D}, z_{1:D}, w_{1:D})}{p(w_{1:D})} \tag{2.17}$$

The marginal probability of the observations $p(w_{1:D})$ is intractable to compute. Therefore Equation (2.17) is approximated by adapting a distribution close to the true posterior. This is generally achieved by either using sampling-based or variational algorithms. Blei et al. [127] use the latter approach which approximates this intractable posterior distribution over hidden variables, with a simpler distribution containing free variational parameters λ, γ, η (Equation (2.18)). The hidden variables of this variational distribution are independent of each other.

$$q(\phi_{1:T}, \theta_{1:D}, z_{1:D}|\lambda, \gamma, \eta) = \prod_{i=1}^{T} \mathrm{Dir}(\phi_i|\lambda_i) \prod_{d=1}^{D} q_d(\theta_d, z_d|\gamma_d, \eta_d) \tag{2.18}$$

The optimising values of the variational parameters are found by minimising the Kullback-Leibler (KL) divergence between the variational distribution $q(\phi, \theta, z)$ and the true posterior:

$$argmin_{q \in Q} KL(q(\phi, \theta, z|\lambda, \gamma, \eta) \,||\, p(\phi, \theta, z|w, \alpha, \beta)) \tag{2.19}$$

The approximate empirical Bayes estimates for the LDA model can be found via an alternating variational expectation-maximization (EM) procedure that maximises a lower bound with respect to the variational parameters λ, γ, η, which yields an approximate posterior distribution on ϕ, θ, z.

Dirichlet Multinomial Mixture

The Dirichlet Multinomial Mixture (DMM) model [134] is a probabilistic generative model for documents. It has two assumptions about its generative process: 1) the documents are generated by a mixture model [141]; 2) there is a one-to-one correspondence between mixture components and clusters, resulting each document is sampled from one single latent topic.

$$\Theta \mid \alpha \sim Dir\,(\alpha) \tag{2.20}$$

$$z_d \mid \Theta \sim Mult(\Theta) \qquad d = 1, ..., D \tag{2.21}$$

$$\Phi_k \mid \beta \sim Dir(\beta) \qquad k = 1, ..., K \tag{2.22}$$

$$d \mid z_d, \{\Phi_k\}_{k=1}^{K} \sim p(d \mid \Phi_{z_d}) \tag{2.23}$$

The graphical representation of DMM is shown in Figure 2.2. To generate document d, DMM first selects a mixture component z_d for document d according to the mixture weights Θ (Equation 2.21) which is generated by a Dirichlet distribution with a hyper-parameter α (Equation 2.20). Then document d is generated from distribution $p(d \mid \Phi_{z_d})$, shown in Equation 2.23, where the cluster parameter Φ_z is also generated by a Dirichlet distribution with a hyper-parameter β (Equation 2.22). The likelihood of document d is characterised by the sum of the total probability over all mixture components:

$$p(d) = \sum_{k=1}^{K} p(d \mid z = k) p(z = k) \tag{2.24}$$

where K is the number of mixture components (i.e. topics). By making the Naive Bayes assumption that all words in document d are generated independently, the probability of d generated by topic k can be derived as:

$$p(d \mid z = k) = \prod_{w \in d} p(w \mid z = k) \tag{2.25}$$

where $p(w \mid z = k) = p(w \mid z = k, \Phi) = \phi_{k,w}$ with $\sum_w \phi_{k,w} = 1$.

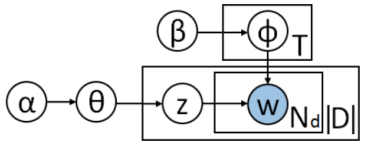


Figure 2.2: Graphical Model of Dirichlet Multinomial Mixture (DMM)

Yin et al. [23] proposed a collapsed Gibbs sampling algorithm for the DMM model. It samples topic z_d for document d using conditional probability $p\left(z_d = z \mid \vec{z}_{\neg d}, \vec{d}\right)$, where $\vec{z}_{\neg d}$ is the topic assignments of documents other than document d.

$$\begin{aligned} & p(z_d = z | \vec{z}_{\neg d}, \vec{d}, \alpha, \beta) \\ & \propto (m_{z,\neg d} + a) \frac{\Gamma(n_{z,\neg d} + V\beta)}{\Gamma(n_{z,\neg d} + n_d + V\beta)} \prod_{w \in W} \frac{\Gamma(n^w_{z,\neg d} + n^w_d + \beta)}{\Gamma(n^w_{z,\neg d} + \beta)} \end{aligned} \tag{2.26}$$

where m_z is the number of documents in topic z, $m_{z,\neg d}$ is the number of documents assigned to topic z excluding the document d, n_z is the number of words in topic z, $n^w_{z,\neg d}$ is the number of occurrences of word w in topic z excluding the document d, Γ is the Gamma function. The sampling process is also described in Section 5.2.

Latent Feature LDA

The latent feature LDA or LFLDA [138] has a mixture of a latent feature component and the topic-word Dirichlet multinomial component of LDA, instead of the topic-

word component alone. As shown in Figure 2.3, τ_t and ω_w are latent feature weights associated with topic t and word w respectively, where w is fixed for pre-trained word embeddings. The generative process of LFLDA starts with randomly choosing

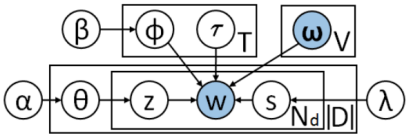


Figure 2.3: Graphical Model of LFLDA

a topic distribution θ_d for document d. Then the model randomly chooses a topic assignment, and it has a binary indicator $s_{d,n}$ sampled from a Bernoulli distribution to choose whether the word $w_{d,n}$ should be generated by the Dirichlet multinomial or latent feature component.

$$\theta_d \sim \mathrm{Dir}(\alpha) \qquad z_{d,n} \sim \mathrm{Cat}(\theta_d) \tag{2.27}$$

$$\phi_z \sim \mathrm{Dir}(\beta) \qquad s_{d,n} \sim \mathrm{Ber}(\lambda) \tag{2.28}$$

$$w_{d,n} \sim (1 - s_{d,n})\mathrm{Cat}(\phi_{z_{d,n}}) + s_{d,n}\mathrm{CatE}(\tau_{z_{d,n}}\omega^{\mathrm{T}}) \tag{2.29}$$

CatE is a categorical distribution with log-space parameters. Sampling-based algorithms in statistical inference attempt to collect samples from the posterior to approximate it with an empirical distribution. LFLDA [138] adopts the Gibbs sampling algorithm for approximating the true posterior. The outline of its algorithm is shown below:

Algorithm 1 An approximate Gibbs sampling algorithm for LFLDA

Initialise the topic-word variables z_{d_i} using the LDA sampling algorithm
for *iteration iter = 1, 2, ...* **do**
    for *topic t = 1, 2, ...* **do**
        $\tau_t = \mathrm{argmax}_{\tau_t} \mathrm{P}(\tau_t | Z, S)$
    for *document d = 1, 2, ..., |D|* **do**
        for *word index i = 1, 2, ..., N_d* **do**
            sample z_{d_i} and s_{d_i} from $\mathrm{P}\left(z_{d_i} = t, s_{d_i} | Z_{\neg d_i}, S_{\neg d_i}, \tau, \omega\right)$

S denotes the distribution indicator variables for the whole corpus D. The algorithm integrates out s_{d_i} to sample z_{d_i}:

$$\begin{aligned}
&\mathrm{P}\left(z_{d_i} = t \mid Z_{\neg d_i}, \tau, \omega\right) \\
&\propto \left(N^t_{d_{\neg i}} + K^t_{d_{\neg i}} + \alpha\right) \left((1-\lambda) \frac{N^{t,w_{d_i}}_{\neg d_i} + \beta}{N^t_{\neg d_i} + V\beta} + \lambda \mathrm{CatE}(\omega_{d_i} \mid \tau_t\, \omega^{\mathrm{T}}) \right)
\end{aligned} \tag{2.30}$$

Then s_{d_i} is sampled given $z_{d_i} = t$:

$$\mathrm{P}(s_{d_i} = s \mid z_{d_i} = t) \propto \begin{cases} (1-\lambda) \frac{N^{t,w_{d_i}}_{\neg d_i} + \beta}{N^t_{\neg d_i} + V\beta} & \text{for } s = 0, \\ \lambda \mathrm{CatE}(\omega_{d_i} \mid \tau_t\, \omega^{\mathrm{T}}) & \text{for } s = 1 \end{cases} \tag{2.31}$$

Here, $N^{t,w}_d$ is the number of times a word w in document d is generated from topic t by the Dirichlet multinomial component of LFLDA, while $K^{t,w}_d$ is the number of times w is generated by the latent feature component. $N^w_d + K^w_d$ is the total number of times the word w appears in the document d.

2.3.3 Evaluation of Topic Models

An earlier work [142] on intrinsically evaluating learnt topics, provided a summary of evaluation techniques using held-out likelihood. Many of these are predictive metrics based on model perplexity, which means they only measure the probability of observations and ignore the internal representation of the models. Chang et al. [143] showed in contrary to expectations the extrinsically measured topic coherence correlates negatively with the model perplexity, which shows the need for a better way of evaluating topic models. Since then various methodologies have been proposed for measuring the intrinsic semantic interpretability of topics, below we describe two most widely used approaches for such task:

a) Chang et al. [143] designed a word intrusion task for indirectly evaluating topic interpretability, where a randomly selected "intruder word" is injected into the

top-N words of a given topic and humans are asked to identify the intruder word that does not belong to the topic. To measure topic interpretability, they defined "model precision" as the relative success of human annotators at identifying the intruder word:

$$\mathrm{MP}_k^m = \sum_s \mathbb{1}(i_{k,s}^m = w_k^m)/S \tag{2.32}$$

where $i_{k,s}^m$ is the index of the intruding word from the $k_t h$ topic inferred by model m, w_k^m is is intruder selected by subject s, and S is the number of subjects. Such word intrusion task is automated in [144], where it is treated as a learning-to-rank problem with the objective of detecting the least representation word (i.e. the intruder word). The pairwise approach is used for identifying intruder word (which has a different target value than the normal topic words) in any given pair of words. For each of the top-N topic words (including intruder word), the authors compute its conditional probabilities, Pointwise Mutual Information (PMI) or Normalised PMI (NPMI) [145] with all other top topic words as word association features. These features are combined along with the target values that define the order of a given word pair, in a ranking support vector regression model (SVM^{rank} [146]) to learn the intruder words. It is shown to achieve near-human levels of accuracy.

b) Newman et al. [147] introduced a more direct approach by calculating the semantic similarity of the top-N words of each topic using external resources such as WordNet and Wikipedia. They found the method based on PMI term co-occurrence using Wikipedia achieving the closest performance to human judgments. [148] found such performance can be substantially improved if the system scores and human ratings are aggregated over different numbers of topic words (i.e. N) before computing the correlation. Other work on directly measuring topic coherence include replacing PMI with conditional probability based on co-document frequency proposed in [149], and using classical distributional semantic similarity methods for

computing the pairwise association of the topic words [150].

Feng at al. [151] evaluated ten automatic topic coherence metrics for Twitter data, and showed a PMI-based metric using Twitter corpus as background data achieving the highest levels of agreement with the human assessments of topic coherence. More recently, [152] showed a new word embedding-based topic coherence metric effectively capturing the coherence of topics from tweets. It is also more robust and efficient than the PMI-based metrics. In our work, we adopt this word embedding-based metric as well as the word intrusion task for the evaluation of our proposed models.

2.4 Opinion Summarisation

With the growth of the web especially social media over the last decade, we now have overwhelming amount of opinions about a broad range of topics all over the Internet. Automatic opinion summarisation system takes these opinionated documents as input and attempts to generate a concise and coherent summary while preserving the most important information and the overall meaning in the input documents [153]. The simplest form of an opinion summary is by aggregating the sentiment scores as proposed in many aspect-based opinion summarisation methods on product reviews [154, 155, 156], or by visualising how sentiments towards different target entities develop in a time series graph as we discuss in Section 6.1. Topic modelling is also be used as a summarisation tool by obtaining representative terms for each topic [157].

Another direction of opinion summarisation research focuses on automatic text summarisation, which was proposed by H. P. Luhn [158] in 1958 with a term frequency based approach. Different to the classic text summarisation problem, the sentiment in the input document is not to be neglected for opinion summarisation. However, text summarisation can be used in the final summary selection or gener-

ation step. Early work in text summarisation mostly focuses on single-document summarisation, where the goal was to construct a summary for one single input document such as a news article or an academic paper. The surge of online text data has led to an increasing research interest in multi-document summarisation where its input consists of multiple different documents. One big challenge for summarising multiple documents is to reduce redundancy and produce concise but informative summaries. In this thesis, we are motivated to summarise tweets mentioning the same topic, and thus our problem formulation falls into the domain of multi-document summarisation.

Methodologically, text summarisation can be classified to two main approaches: extractive summarisation and abstractive summarisation. Methods for extractive summarisation select relevant sentences or parts of sentences from the original document(s) to form the summary, whereas abstractive summarisation produces an abstract summary applying natural language generation which is more challenging. Most existing tweet summarisation approaches rely on extractive methods, which rank and select tweets according to various relevance criteria for a summary. This approach unavoidably ends up including secondary, incomplete or redundant information. These summaries also typically lack coherence and cohesion. On the contrary, abstractive approaches aim to compose the summary from scratch that draws information from different sources, potentially using vocabulary unseen in the original document. Such abstractive summaries can be less verbose, more informative and are more likely to resemble high-quality, human written pieces representing the collective opinion of tweets on a given topic or entity. In the following sections, we review relevant work on both extractive and abstractive summarisation including the recent development on text summarisation using neural models, as well as summarisation on tweets.

2.4.1 Extractive Summarisation

For traditional documents such as news articles, a large number of extractive summarisation techniques have been developed over the past decade, largely contributed by conferences like the Text Analysis Conferences[7] (previously known as Document Understanding Conferences[8]) and Text Retrieval Conferences[9]. The extractive approach formulates the summarisation problem as a sentence selection task, thus the summary becomes easier to construct and does not suffer from the grammatical issue. As a result, measuring sentence importance and extracting the top N most important sentences, are the essential parts of the task. The common multi-document extractive summarisation approach includes the centroid-based [159], graph-based [160], sentence-based topic model [161], (and for selecting sentences) greedy search [162, 163], integer linear programming (ILP) [164], submodular function maximisation [165, 166], and supervised learning to rank based methods [167]. Recently, deep neural network has been applied in the extractive summarisation research although mostly for single-document [168, 169, 170], while the few for multi-document includes a joint learning framework of summarisation and text classification [171].

Though a lot of progress has been made in extractive summarisation, the extractive approach has the limitation of unavoidably including redundant information and its summaries typically lack cohesion. In [172] the authors suggest that advances in extractive text summarisation have slowed down in the past few years. More importantly, extractive summarisation is fundamentally different to how humans write summaries. As reviewed in [173], there are two possible directions of further research in summarisation. One option is to make an ensemble of multiple extractive models. The other approach is to move towards the area of abstractive summarisation, which we review in the following section.

[7] `http://tac.nist.gov/`
[8] `http://duc.nist.gov/`
[9] `http://trec.nist.gov/`

2.4.2 Abstractive Summarisation

Rather than simply extract important sentences, abstractive summarisation covers techniques that designed to resemble the way humans construct summaries. In [174], the author compares the human generated summaries to the original input documents, and observes that humans tend to use and modify the input content in four ways: sentence compression, information fusion, paraphrasing, and generation.

Extractive summaries have inherent limitations primarily because only a part of the extracted sentences is informative and the other part is redundant. **Sentence compression** methods aim to create a compact and grammatical sentence as summary while keeping salient information. Much work on this approach has looked at deletion-based sentence compression techniques [175, 176]. **Information fusion** is another approach for generating non-extractive summaries, which aims to fuse multiple sentences by removing redundant content while preserving important information. Among the information fusion methods, the graph-based techniques have attracted much research interest [177, 178]. These techniques generally construct a word graph from topically related sentences and select the best suited path as the final summary. When choosing a path, several different factors can be considered such as redundancy, informativeness and readability. [178] identifies such summary path using an integer linear programming (ILP) model.

Generating summary from scratch is far more challenging than compressing or fusing sentences, since both language understanding and generation are required. The **abstract generation** approach extracts concepts about the input documents rather than sentences or phrases, and the relationships among these concepts. In addition to conciseness and informativeness, it is also difficult to generate a summary that is grammatical, coherent and semantically correct. Early work in abstractive summary generation rely on manually crafted templates or rules for generating grammatically correct sentences [179, 180]. In recent years, there has been a surge of

interest in using sequence transduction neural network architectures for NLP tasks such as machine translation, question answering, dialogue generation, and abstractive summarisation.

Central to these approaches is a sequence-to-sequence (or **seq2seq**) model, as introduced in [73], consists of two recurrent neural networks (RNNs): an *encoder* that reads the input sequence and encodes into a fixed-size state vector, which is passed to a *decoder* that generates the output sequence. Another prominent work [35] uses two multi-layered Long Short-Term Memory networks (LSTMs) for the encoding and decoding. They show even with a limited vocabulary, the seq2seq model can do well on sequence learning problems such as a large scale machine translation task. To locate the region of focus during decoding, an attention mechanism was introduced in [181]. This makes the seq2seq model more reliable with long sentences.

Inspired by the development of neural machine translation, Rush et al. [182] were the first to apply the encoder-decoder architecture to **neural abstractive summarisation**. They use a convolutional model for encoding, and an attentional feed-forward network along with beam search for generating the summary. As an extension to this work, [183] replace the decoder with an RNN, achieving improved performance. Both studies evaluate on two sentence-level news article summarisation datasets, namely Gigaword[10] and DUC-2004[11]. The headline of each article and its first sentence are paired to create input-summary corpus. Nallapati et al. [184] present a new corpus that comprises multi-sentence summaries, by modifying a question answering dataset for summarisation, resulting in the *CNN/Daily Mail* dataset. For handling out-of-vocabulary (OOV) words (with respect to training data), instead of emitting the 'UNK' token as placeholder, the authors train a decoder/pointer switch that either generates a word from the vocabulary or copies a word from the source text. To improve the handling of rare and OOV words, [185] use a hybrid pointer-generator network, which learns when to use the pointer by mix-

[10] `https://catalog.ldc.upenn.edu/LDC2012T21`
[11] `http://duc.nist.gov/data.html`

ing the probabilities from copy distribution and the vocabulary distribution. They show this mixture approach can accurately reproduce rare but in-vocabulary words. Such pointer-generator model has the tendency to repeat itself when producing summary. To discourage repetition, the authors propose a coverage mechanism to keep track of what has been summarised. [186] introduce a mixed objective learning function for abstractive summarisation, which combines the maximum-likelihood cross-entropy loss used in word prediction with rewards from policy gradient reinforcement learning (RL). To summarise a set of multiple text units like movie reviews, [187] design an importance-based sampling method using manually engineered features for generating input for the encoder. To have better summaries, the authors also perform post-processing re-ranking based on cosine similarity.

The one key constraint of the seq2seq models is that they require a large amount of labelled training data, which is expensive to obtain, such as the Gigaword corpus used in many of the aforementioned work and The New York Times Annotated Corpus [188]. A number of studies have explored using unlabelled data for learning a language model or sequence autoencoder as a pretraining step, to initialise the network in another supervised model for text classification [189, 190], machine translation [191] or abstractive summarisation [191, 192], and showing improved performance. A similar approach has been used in the machine translation research to transfer learnt parameters trained from high-resource data to the low-resource scenario [193, 194]. [195] investigated the feasibility of cross-domain (news stories to opinion articles) abstractive summarisation. They found a model trained on out-of-domain data can learn to detect summary-worthy content, but may not match the generation style in the target domain. To the best of our knowledge, there is currently no study applying seq2seq abstractive summarisation on tweets, possibility due to the insufficient training resource.

Sequence-to-sequence Learning

Sequence-to-sequence (seq2seq) learning refers to a set of sequential learning problems that aims in mapping an variable length sequence as input to another variable length sequence as output. For example, speech recognition, machine translation and text summarisation are such problems. Such a seq2seq model is a general method to learn the conditional distribution over an output sequence conditioned on the input sequence, $p\left(y_1, ..., y_{N'} \mid x_1, ..., x_N\right)$, where the the input and output sequence lengths N and N' are unknown and may differ.

A seq2seq has two neural networks, which the first neural network maps the input sequence to a fixed-sized vector representation (i.e. encoding), and the second neural network maps the vector representation to the target sequence (i.e. decoding). In [73] the encoder is an RNN that reads each symbol of an input sequence $\mathbf{x}$ sequentially, and the aforementioned conditional probability is computed by obtaining the representation v of the input sequence $(x_1, ..., x_N)$ given by the last hidden state of the RNN. The decoder is another RNN that computes the probability of $(y_1, ..., y_{N'})$ with a standard RNN language model formulation:

$$p\left(y_1, ..., y_{N'} \mid x_1, ..., x_N\right) = \Pi_{n=1}^{N'} p\left(y_n \mid v, y_1, ..., y_{n-1}\right) \tag{2.33}$$

where the initial hidden state is set to be the representation v of $x_1, ..., x_N$, and finally each $p\left(y_n \mid v, y_1, ..., y_{n-1}\right)$ distribution is represented with a softmax over all the words in the fixed vocabulary. Sutskever et al. [35] use two LSTMs for encoding and decoding, as it is better at learning long range temporal dependencies. The encoder and encoder of seq2seq are jointly trained to maximise the conditional log-likelihood:

$$\max_\theta \frac{1}{N} \sum_{n=1}^{N} \log p_\theta(\mathbf{y}_n \mid \mathbf{x}_n) \tag{2.34}$$

where θ is the set of model parameters. Once the model is trained, a beam search

(i.e. N-best search) is commonly used to find a target sequence (e.g. translation or summarisation) that approximately maximises the conditional probability. This means during testing at each timestep, instead of considering every possible hypothesis of the output sequence or the one best hypothesis (i.e. Greedy search), it only consider the b most likely hypotheses according to the model's log probability where b is the "width" of the beam.

2.4.3 Tweets Summarisation

As we have discussed in Chapter 1, social media has become a rich resource for policy makers and organisations to understand public opinion. However, understanding the sentiment towards different issues and entities as manifested in the large volume of tweets is still a difficult task. The traditional way of collecting such public opinions is by the use of opinion polls, which is costly and the polls themselves carry bias. In recent years we have seen a number of studies linking opinions expressed on Twitter and real world events and stories. For example, an early paper by O'Connor et al [196] found both consumer confidence and presidential approval polls exhibited correlation with Twitter sentiment.

The task of summarising large amount of opinions expressed on Twitter is related to aspect-based summarisation [154, 197, 155], which is concerned with aspects of the target and the sentiment towards each aspect. These methods aim to identify the important features for each aspect and attach relevant reviews or other opinionated sentences to the corresponding feature, providing aspect-based summary in a structured way. The diverse, noisy and unstructured nature of tweets makes its summarisation a more challenging task than summarising product reviews. Louis and Newman [198] presented a concept-based approach that maps business-related tweets into the corresponding concepts learnt using external resources, and selects tweets with the highest average probability of words incorporating sentiment information for each top-ranked cluster. In this thesis, our goal is to construct a

fluent text-based summary for tweets mentioning the same target carrying the same sentiment, and thus different to the structured summary provided by the aspect-based opinion summarisation.

Most work in the literature on tweets summarisation focus on generating summary for real-world events such as natural disasters [199, 200] and sport games [201, 202] or trending topics [203, 24, 204], with the aim to reduce information overload and provide key update for the corresponding story. It has become a popular research task demonstrated by the Microblog [205], Temporal Summarisation [206] and Real-Time Summarisation [207] tracks at Text Retrieval Conferences (TRECs) as well as the more recent Exploitation of Social Media for Emergency Relief and Preparedness (SMERP) track [200] at European Conference on Information Retrieval (ECIR). Among these works, a majority of early studies pursue either graph-based [208, 203, 209] or term-frequency based [210, 201] approach for extractive summarisation of tweets. A study by Inouye and Kalita [24] compares eight algorithms and reports the simple term-frequency with redundancy reduction based methods, namely multi-post Hybrid tf-idf and SumBasic [211], achieving the closest performance to human evaluation scores, possibly due to the short, unstructured and unconnected nature of tweets. [212] apply summarisation for tackling the topic labelling problem. They also found the frequency based methods outperforming the other approaches. [209] present a Pagerank-like algorithm for generating summaries of variable lengths. Time-aware summarisation or timeline generation has also attracted research interest for generating event summary in the form of timeline [202, 213, 214]. Both [202] and [213] rely on tweet burstiness for identifying important moments or sub-events of a sports event. [215] propose a time-aware user behavior model to select representative tweets as summary, based on the user's history and collaborative social influences from its social circles.

To determine the salience of the tweets, many studies have also focused on incorporating the social influence of users and their social network (e.g. follower-

followee relationship) structure [216, 217, 215, 218]. Finding insightful and informative tweets is challenging, a related work by Swapna and Jiang [219] tackles the task of detecting thoughtful online comments as a classification problem by studying various linguistic features and training a logistic regression model. Some other works use related web contents to provide additional useful topic information to improve summarisation [204, 220]. The motivation of our work in this thesis is related to [25], which also proposes a topic-oriented opinion summarisation framework. However, they use a template-matching method for identifying insightful tweets and the final representative summary tweets are selected through a optimisation procedure, which is different to our approach described in Chapter 6.

While majority of the summarisation research on tweets including all the aforementioned studies choose to adopt the extractive approach, abstractive summaries are potentially more cohesive and less redundant. However, there has been few work exploring abstractive summarisation of tweets as it is easily affected by noise or the diversity of tweets. Ganesan et al. [177] introduce a graph-based algorithm for merging opinions that share similar textual content and thus reducing redundancy. Because it generates word-graph and explores various sub-paths to construct the final summary, it can still be regarded as a word-level extractive summarisation. This method can be used on highly redundant text such as tweets. [221] propose to update the word-graph constantly with tweets which enables for online abstractive summarisation. A more recent work [199] propose a two-stage summarisation framework, which first identifies a set of important tweets using a content-word based extractive approach [222] and then constructs bigram word-graph followed by integer linear programming based optimisation.

In this thesis we investigate and study the feasibility of applying state-of-the-art neural abstractive summarisation for events and opinions expressed on Twitter, with limited training resources. Additionally, we present a visualisation system for displaying opinion summary towards different topics on each day, using the

techniques described in Chapter 4 and Chapter 5.

CHAPTER 3

Preliminary studies

Twitter social spam detection and cross-domain emotion analysis

In the previous chapter we have introduced the background material for this thesis. Starting with this chapter, we begin presenting our research and answering the research questions listed in Chapter 1. In this chapter we present our preliminary studies for preparing and building up our main research work in its following chapters. These preliminary studies are set to address two questions:

- *How can we develop an efficient and effective way to filter out spam tweets in a data pipeline?*
- *How can we improve emotion classification performance on Twitter when training and testing data are not in the same domain, by using domain adaptation?*

3.1 Social Spam Detection

3.1.1 Introduction

Social networking spam, or social spam, is increasingly affecting social networking websites, such as Facebook, Pinterest and Twitter. According to a study by the social media security firm Nexgate [223], social media platforms experienced a 355% growth of social spam during the first half of 2013. Social spam can reach a surprisingly high visibility even with a simple bot [224], which detracts from a company's social media presence and damages their social marketing ROI (Return On Investment). Moreover, social spam exacerbates the amount of unwanted information that average social media users receive in their timeline, and can occasionally even affect the physical condition of vulnerable users through the so-called "*Twitter psychosis*" [225].

Social spam has different effects and therefore its definition varies across major social networking websites. One of the most popular social networking services, Twitter, has published their definition of spamming as part of their "The Twitter Rules"[1] and provided several methods for users to report spam such as tweeting "*@spam @username*" where *@username* will be reported as a spammer. While as a business, Twitter is also generous with mainline bot-level access[2] and allows some level of advertisements as long as they do not violate "The Twitter Rules". In recent years we have seen Twitter being used as a prominent knowledge base for discovering hidden insights and predicting trends from finance to public sector, both in industry and academia. The ability to sort out the signal (or the information) from Twitter noise is crucial, and one of the biggest effects of Twitter spam is that it significantly reduces the signal-to-noise ratio. Our work on social spam is motivated by the initial attempts at harvesting a Twitter corpus around a specific topic with

[1] https://support.twitter.com/articles/18311-the-twitter-rules
[2] http://www.newyorker.com/tech/elements/the-rise-of-twitter-bots

a set of predefined keywords [33]. This led to the identification of a large amount of spam within those datasets. The fact that certain topics are trending and therefore many are tracking its contents encourages spammers to inject their spam tweets using the keywords associated with these topics to maximise the visibility of their tweets.

As mentioned in Chapter 2, the definition of social spam is context dependant. Here we define social spam as tweets posted by content polluters (e.g. malicious promoters and friend infiltrators [47]) who aim to inject unrelated tweets in timely topics, share malicious links or fraudulent information. As a result social spam usually has different features to normal tweets (e.g. contains many hashtags to increase its visibility), and produces a significant amount of noise both to end users who follow the topic as well as to tools that mine Twitter data.

As described in Section 2.1, the automatic detection of Twitter spam has been addressed in two different ways. The first way is to tackle the task as a user classification problem, which makes use of numerous features that need to gather historical details about a user, such as tweets that a user posted in the past to explore what they usually tweet about, or how the number of followers and followings of a user has evolved in recent weeks to discover unusual behaviour. While this is ideal as the classifier can make use of extensive user data, it is often unfeasible due to restrictions of the Twitter API. The second, alternative way, is to define the task as a tweet classification problem, where a tweet can be deemed spam or non-spam. In this case, the classification task needs to assume that only the information provided within a tweet is available to determine if it has to be categorised as spam. Here, we follow this approach to Twitter spam classification, and propose to classify if a tweet is spam or not by using its inherent features. While this is more realistic for our scenario, it presents the extra challenge that the available features are rather limited, which we study here.

Here we present a comparative study of Twitter spam detection systems. We

investigate the use of different features inherent to a tweet so as to identify the sets of features that do best in categorising tweets as spam or not. Our study compares five different classification algorithms over two different datasets. The fact that we test our classifiers on two different datasets, collected in different ways, enables us to validate the results and claim repeatability. Our results suggest a competitive performance can be obtained using tree-based classifiers for spam detection even with only tweet-inherent features, as comparing to the existing spammer detection studies.

3.1.2 Datasets

A labelled collection of tweets is crucial in a machine learning task such as spam detection. We found no spam dataset which is publicly available and specifically fulfils the requirements of our task. Instead, the datasets we obtained include Twitter users labelled as spammers or not. For our work, we used the latter, which we adapted to our purposes by taking out the features that would not be available in our scenario of spam detection from tweet-inherent features. We used two spammer datasets in this work, which have been created using different data collection techniques and therefore is suitable to our purposes of testing the spam classifier in different settings. To accommodate the datasets to our needs, we sample one tweet for each user in the dataset, so that we can only access one tweet per user and cannot aggregate several tweets from the same user or use social network features. In what follows we describe the two datasets we use.

Social Honeypot Dataset: Lee et al. [47] created and manipulated (by posting random messages and engaging in none of the activities of legitimate users) 60 social honeypot accounts on Twitter to attract spammers. Their dataset consists of 22,223 spammers and 19,276 legitimate users along with their most recent tweets. They used Expectation-Maximization (EM) clustering algorithm and then manually grouped their harvested users into 4 categories: duplicate spammers, duplicate @

spammers, malicious promoters and friend infiltrators. **1KS-10KN Dataset:** Yang et al. [48] defines a tweet that contains at least one malicious or phishing URL as a spam tweet, and a user whose spam ratio is higher than 10% as a spammer. Therefore their dataset which contains 1,000 spammers and 10,000 legitimate users, represents only one major type of spammers (as discussed in their paper).

We used *spammer vs. legitimate user* datasets from [47] and [48]. After removing duplicated users and the ones that do not have any tweets in the dataset we randomly selected one tweet from each spammer or legitimate user to create our labelled collection of *spam vs. legitimate tweets*, in order to avoid overfitting and reduce our sampling bias. The resulting datasets contain 20,707 spam tweets and 19,249 normal tweets (named Social Honeypot dataset, as from [47]), and 1,000 spam tweets and 9,828 normal tweets (named 1KS-10KN dataset, as from [48]) respectively. The example spam tweets are shown in Table 3.1:

Dataset	Sample tweet
Social Honeypot	www.ppnchat.com has got ot be the best chat site on the net, it's free and fun. Real people,real talk!(9:33)
Social Honeypot	Free trial this miracle fruit from the amazon
Social Honeypot	#par #nzl #svk #bra #prk #civ #por #esp #sui #hon #chi #worldcup ;D
Social Honeypot	#LOWEST #Single #Unique #Bid #Win a #Lenovo IdeaPad U450p #Laptop #Value $576.99 #Auction ends:1/28/10@08:00 www.us-DubLi.com #Shopping FUN
1KS-10KN	get 88 followers per day using http://xrl.us/bgingb , fast!
1KS-10KN	adults looking for fun Must see http://twurl.nl/hsudj0 :)getting sleepy
1KS-10KN	hey cuties, im single again.. message me at http://wowurl.com/16r

Table 3.1: Examples of spam tweets

3.1.3 Features

As spammers and legitimate users have different goals in posting tweets or interacting with other users on Twitter, we can expect that the characteristics of spam tweets are quite different to the normal tweets. The features inherent to a tweet include, besides the tweet content itself, a set of metadata including information

about the user who posted the tweet, which is also readily available in the stream of tweets we have access to in our scenario. We analyse a wide range of features that reflect user behaviour, which can be computed straightforwardly and do not require high computational cost, and also describe the linguistic properties that are shown in the tweet content. We considered four feature sets: (i) user features, (ii) content features, (iii) n-grams, and (iv) sentiment features.

User features	**Content features**
Length of profile name	Number of words
Length of profile description	Number of characters
Number of followings (FI)	Number of white spaces
Number of followers (FE)	Number of capitalization words
Number of tweets posted	Number of capitalization words per word
Age of the user account, in hours (AU)	Maximum word length
Ratio of number of followings and followers (FE/FI)	Mean word length
Reputation of the user (FE/(FI + FE))	Number of exclamation marks
Following rate (FI/AU)	Number of question marks
Number of tweets posted per day	Number of URL links
Number of tweets posted per week	Number of URL links per word
N-grams	Number of hashtags
Uni + bi-gram or bi + tri-gram	Number of hashtags per word
	Number of mentions
Sentiment features	Number of mentions per word
Automatically created sentiment lexicons	Number of spam words
Manually created sentiment lexicons	Number of spam words per word
	Part of speech tags of every tweet

Table 3.2: List of features used for spam detection

User features include a list of 11 attributes about the author of the tweet (as seen in Table 3.2) that is generated from each tweet's metadata, such as reputation of the user [27], which is defined as the ratio between the number of followers and the total number of followers and followings and it had been used to measure user influence. Other candidate features, such as the number of retweets and favourites garnered by a tweet, were not used given that it is not readily available at the time

of posting the tweet, where a tweet has no retweets or favourites yet.

Content features capture the linguistic properties from the text of each tweet (Table 3.2) including a list of content attributes and part-of-speech tags. Among the 17 content attributes, number of spam words and number of spam words per word are generated by matching a popular list of spam words[3]. Part-of-speech (or POS) tagging provides syntactic (or grammatical) information of a sentence and has been used in the natural language processing community for measuring text informativeness (e.g. Tan et al. [226] used POS counts as a informativeness measure for tweets). We have used a Twitter-specific tagger [227], and in the end our POS feature consists of uni-gram and 2-skip-bi-gram representations of POS tagging for each tweet in order to capture the structure and therefore informativeness of the text.

N-gram models have long been used in natural language processing for various tasks including text classification. Although it is often criticised for its lack of any explicit representation of long range or semantic dependency, it is surprisingly powerful for simple text classification with reasonable amount of training data.

Sentiment features: Ferrara et al. [49] used tweet-level sentiment as part of their feature set for the purpose of detecting Twitter bots. We have used the same list of lexicons from [58] (which has been proved of achieving top performance in the Semeval-2014 Task 9 Twitter sentiment analysis competition) for generating our sentiment features, including manually generated sentiment lexicons: AFINN lexicon [228], Bing Liu lexicon [229], MPQA lexicon [230]; and automatically generated sentiment lexicons: NRC Hashtag Sentiment lexicon [58] and Sentiment140 lexicon [58].

[3]https://github.com/splorp/wordpress-comment-blacklist/blob/master/blacklist.txt

3.1.4 Selection of Classifier

During the classification and evaluation stage, we tested 5 classification algorithms implemented using scikit-learn[4]: Bernoulli Naive Bayes, K-Nearest Neighbour (KNN), Support Vector Machines (SVM), Decision Tree, and Random Forests. These algorithms were chosen as being the most commonly used in the previous research on spammer detection. We evaluate using the standard information retrieval metrics of recall (R), precision (P) and F1-measure.

In order to select the best classifier for our task, we have used a subset of each dataset (20% for 1KS-10KN dataset and 40% for Social Honeypot dataset, due to the different sizes of the two datasets) to run a 10-fold cross validation for optimising the hyperparameters of each classifier. By doing so it minimises the risk of over-fitting in model selection and hence subsequent selection bias in performance evaluation. Such optimisation was conducted using all 4 feature sets (each feature was normalised to fit the range of values [-1, 1]; we also selected 30% of the highest scoring features using Chi Square for tuning SVM as computationally it is more efficient and gives better classification results). Then we evaluated our algorithm on the rest of the data (i.e. 80% for 1KS-10KN dataset and 60% for Social Honeypot dataset), again using all 4 feature sets in a 10-fold cross validation setting (same as in grid-search, each feature was normalised and Chi square feature selection was used for SVM).

As shown in Table 3.3, tree-based classifiers achieved very promising performances, among which Random Forests outperform all the others when we look at the F1-measure. This outperformance occurs especially due to the high precision values of 99.3% and 94.1% obtained by the Random Forest classifier. While Random Forests show a clear superiority in terms of precision, its performance in terms of recall varies for the two datasets; it achieves high recall for the Social Honeypot dataset, while it drops substantially for the 1KS-10KN dataset due to its approx-

[4]http://scikit-learn.org/

imate 1:10 spam/non-spam ratio. These results are consistent with the conclusion of most spammer detection studies; our results extend this conclusion to the spam detection task.

When we compare the performance values for the different datasets, it is worth noting that with the Social Honeypot dataset the best result is more than 10% higher than the best result in 1KS-10KN dataset. This is caused by the different spam/non-spam ratios in the two datasets, as the Social Honeypot dataset has a roughly 50:50 ratio while in 1KS-10KN it is roughly 1:10 which is a more realistic ratio to reflect the amount of spam tweets existing on Twitter (In Twitter's 2014 Q2 earnings report it says that less than 5% of its accounts are spam[5], but independent researchers believe the number is higher). In comparison to the original papers, [47] reported a best 0.983 F1-score and [48] reported a best 0.884 F1-score. Our results are only about 4% lower than their results, which make use of historical and network-based data, not readily available in our scenario. Our results suggest that a competitive performance can also be obtained for spam detection where only tweet-inherent features can be used.

Classifier	**1KS-10KN Dataset**			**Social Honeypot Dataset**		
	Precision	**Recall**	**F-measure**	**Precision**	**Recall**	**F-measure**
Bernoulli NB	0.899	0.688	0.778	0.772	0.806	0.789
KNN	0.924	0.706	0.798	0.802	0.778	0.790
SVM	0.872	0.708	0.780	0.844	0.817	0.830
Decision Tree	0.788	0.782	0.784	0.914	0.916	0.915
Random Forest	0.993	0.716	**0.831**	0.941	0.950	**0.946**

Table 3.3: Comparison of performance of spam classifiers

3.1.5 Evaluation of Features

We trained our best classifier (i.e. Random Forests) with different feature sets, as well as combinations of the feature sets using the two datasets (i.e. the whole corpora), and under a 10-fold cross validation setting. We report our results in Table

[5] `http://www.webcitation.org/6VyBTJ7vt`

3.4. As seen in 1KS-10KN dataset, the F1-measure for different feature sets ranges from 0.718 to 0.820 when using a single feature set. All feature set combinations except C + S (content + sentiment feature) perform higher than 0.810 in terms of F1-measure, reflecting that feature combinations have more discriminative power than a single feature set.

For the Social Honeypot dataset, we can clearly see User features (U) having the most discriminative power as it has a 0.940 F1-measure. Results without using User features (U) have significantly worse performance, and feature combinations with U give very little improvement with respect to the original 0.940 (except for U + Uni & Bi-gram (Tf) + S). This means U is dominating the discriminative power of these feature combinations and other feature sets contribute very little in comparison to U. This is potentially caused by the data collection approach (i.e. by using social honeypots) adopted by [47], which resulted in the fact that most spammers that they attracted have distinguishing user profile information compared to the legitimate users. On the other hand, Yang et al. [48] checked malicious or phishing URL links for collecting their spammer data, and this way of data collection gives more discriminative power to Content and N-gram features than [47] does (although U is still a very significant feature set in 1KS-10KN). Note that U + Bi & Tri-gram (Tf) resulted in the best performance in both datasets, showing that these two feature sets are the most beneficial to each other irrespective of the different nature of datasets.

3.1.6 Discussion and Conclusion

Our study looks at different classifiers and feature sets over two spam datasets to pick the settings that perform best. First, our study on spam classification buttresses previous findings for the task of spammer classification, where Random Forests were found to be the most accurate classifier. Second, our comparison of four feature sets reveals the features that, being readily available in each tweet, perform best in

Feature Set	1KS-10KN Dataset			Social Honeypot Dataset		
	Precision	**Recall**	**F-measure**	**Precision**	**Recall**	**F-measure**
User features (U)	0.895	0.709	0.791	0.938	0.940	0.940
Content features (C)	0.951	0.657	0.776	0.771	0.753	0.762
Uni + Bi-gram (Tf)	0.959	0.715	0.819	0.783	0.767	0.775
Sentiment features (S)	0.966	0.574	0.718	0.679	0.727	0.702
U + C	0.974	0.708	0.819	0.938	0.949	**0.943**
U + Bi & Tri-gram (Tf)	0.972	0.745	**0.843**	0.937	0.949	**0.943**
U + S	0.948	0.732	0.825	0.940	0.944	0.942
Uni & Bi-gram (Tf) + S	0.964	0.721	0.824	0.797	0.744	0.770
C + S	0.970	0.649	0.777	0.778	0.762	0.770
C + Uni & Bi-gram (Tf)	0.968	0.717	0.823	0.783	0.757	0.770
U + C + Uni & Bi-gram (Tf)	0.985	0.727	0.835	0.934	0.949	0.941
U + C + S	0.982	0.704	0.819	0.937	0.948	0.942
U + Uni & Bi-gram (Tf) + S	0.994	0.720	0.834	0.928	0.946	0.937
C + Uni & Bi-gram (Tf) + S	0.966	0.720	0.824	0.806	0.758	0.782
U + C + Uni & Bi-gram (Tf) + S	0.988	0.725	0.835	0.936	0.947	0.942

Table 3.4: Performance evaluation of various feature set combinations

identifying spam tweets. While different features perform better for each of the datasets when using them alone, our comparison shows that the combination of different features leads to an improved performance in both datasets. We believe that the use of multiple feature sets increases the possibility to capture different spam types, and makes it more difficult for spammers to evade all feature sets used by the spam detection system. For example spammers might buy more followers to look more legitimate but it is still very likely that their spam tweet will be detected as its tweet content will give away its spam nature.

Due to practical limitations, we have generated our spam vs. non-spam data from two spammer vs. non-spammer datasets that were collected in 2011. For future work, we plan to generate a labelled spam/non-spam dataset which was crawled in 2017. This will not only give us a purpose-built corpus of spam tweets to reduce the possible effect of sampling bias of the two datasets that we used, but will also give us insights on how the nature of Twitter spam changes over time and how spammers have evolved since 2011 (as spammers do evolve and their spam content are manipulated to look more and more like normal tweet). Furthermore we will investigate the feasibility of cross-dataset spam classification using domain adaptation methods, and also whether unsupervised approaches work well enough

in the domain of Twitter spam detection.

A caveat of the approach we relied on for the dataset generation is the fact that we have considered spam tweets posted by users who were deemed spammers. This was done based on the assumption that the majority of social spam tweets on Twitter are shared by spam accounts. However, the dataset could also be complemented with spam tweets which are occasionally posted by legitimate users, which our work did not deal with. An interesting study to complement our work would be to look at these spam tweets posted by legitimate users, both to quantify this type of tweets, as well as to analyse whether they present different features from those in our datasets, especially when it comes to the user-based features as users might have different characteristics. For future work, we plan to conduct further evaluation on how our features would function for spam tweets shared by legitimate users, in order to fully understand the effects of bias of pursuing our approach of corpus construction.

In conclusion our approach differs from most previous research works that classified Twitter users as spammers or not, and represents a real scenario where either a user is tracking an event on Twitter, or a tool is collecting tweets associated with an event. In these situations, the spam removal process cannot afford to retrieve historical and network-based features for all the tweets involved with the event, due to high number of requests to the Twitter API that this represents. By conducting extensive evaluation we show our model achieving competitive performance and can be used in a data pipeline for filtering out spam tweets. We have indeed used the proposed spam detection model for our research work in Section 6.1 to improve the data quality.

3.2 Twitter Emotion Analysis

3.2.1 Introduction

In recent years we have also seen a surge of research in sentiment analysis with over 7,000 articles written on the topic [52], for applications ranging from analyses of movie reviews [231] and stock market trends [15] to forecasting election results [13]. Supervised learning algorithms that require labelled training data have been successfully used for in-domain sentiment classification. However, cross-domain sentiment analysis has been explored to a much lesser extent. For instance, the phrase "light-weight" carries positive sentiment when describing a laptop but quite the opposite when it is used to refer to politicians. In such cases, a classifier trained on one domain may not work well on other domains. While a domain-independent classifier would be ideal, it would require a large amount of human labelled corpora, which is very costly. A widely adopted solution to this problem is domain adaptation, which allows building models from a fixed set of source domains and deploy them into a different target domain. It can be considered as a special setting of transfer learning [232] that aims at transferring knowledge across different domains. Recent developments in sentiment analysis using domain adaptation are mostly based on feature-representation adaptation [79, 81, 82], instance-weight adaptation [84, 85, 31] or combinations of both [233, 83]. Despite its recent increase in popularity, the use of domain adaptation for sentiment and emotion classification across topics on Twitter is still largely unexplored [83, 31, 80]. Not surprisingly, [86] conducted experiments on topic-dependent cross-medium sentiment classification, and found that cross-topic adaptation is more challenging on Twitter data than on other kinds of data, owing to the noisy and sparse nature of tweets.

In this section we set out to find an effective approach for tackling the cross-domain emotion classification task on Twitter, while also furthering research in the interdisciplinary study of social media discourse around arts and cultural experi-

ences[6]. We investigate a model-based adaptive-SVM approach that was previously used for video concept detection [94] and compare with a set of domain-dependent and domain-independent strategies. Such a model-based approach allows us to directly adapt existing models to the new target-domain data without having to generate domain-dependent features or adjusting weights for each of the training instances, and thus is more efficient and flexible for our task. We conduct a series of experiments and evaluate the proposed system[7] on a set of Twitter data about museums, annotated by three annotators with social science background. The aim is to maximise the use of the base classifiers that were trained from a general-domain corpus, and through domain adaptation minimise the classification error rate across 5 emotion categories: *anger*, *disgust*, *happiness*, *surprise* and *sadness*. Our results show that adapted SVM classifiers achieve significantly better performance than out-of-domain classifiers and also suggest a competitive performance compared to in-domain classifiers. To the best of our knowledge this is the first attempt at cross-domain emotion classification for Twitter data.

3.2.2 Datasets

We use two datasets, a source-domain dataset and a target-domain dataset, which enables us to experiment on domain adaptation. The source-domain dataset we adopted is the general-domain Twitter corpus created by [18], which was generated through distant supervision using hashtags and emoticons associated with 6 emotions: anger, disgust, fear, happiness, surprise and sadness.

Our target-domain dataset that allows us to perform experiments on emotions associated with cultural experiences consists of a set of tweets pertaining to museums. A collection of tweets mentioning one of the following Twitter handles associated with British museums was gathered between May 2013 and June 2015: *@camunivmuseums*, *@fitzmuseum_uk*, *@kettlesyard*, *@maacambridge*, *@icia-*

[6]SMILE project: `http://www.culturesmile.org/`

[7]The code can be found at `http://bit.ly/1WHup4b`

bath, *@thelmahulbert*, *@rammuseum*, *@plymouthmuseum*, *@tateliverpool*, *@tate_stives*, *@nationalgallery*, *@britishmuseum*, *@_thewhitechapel*. These are all museums associated with the SMILES project. A subset of 3,759 tweets was sampled from this collection for manual annotation. We developed a tool for manual annotation of the emotion expressed in each of these tweets. The options for the annotation of each tweet included 6 different emotions; the six Ekman emotions as in [18], with the exception of 'fear' as it never featured in the context of tweets about museums. Two extra annotation options were included to indicate that a tweet should have *no code*, indicating that a tweet was not conveying any emotions, and *not relevant* when it did not refer to any aspects related to the museum in question. The annotator could choose more than one emotion for a tweet, except when *no code* or *not relevant* were selected, in which case no additional options could be picked. The annotation of all the tweets was performed independently by three sociology PhD students. Out of the 3,759 tweets that were released for annotation, at least 2 of the annotators agreed in 3,085 cases (82.1%). We use the collection resulting from these 3,085 tweets as our target-domain dataset for classifier adaptation and evaluation. Note that tweets labelled as *no code* or *not relevant* are included in our dataset to reflect a more realistic data distribution on Twitter, while our source-domain data doesn't have any *no code* or *not relevant* tweets.

The distribution of emotion annotations in Table 3.5 shows a remarkable class imbalance, where *happy* accounts for 30.2% of the tweets, while the other emotions are seldom observed in the museum dataset. There is also a large number of tweets with no emotion associated (41.8%). One intuitive explanation is that Twitter users tend to express positive and appreciative emotions regarding their museum experiences and shy away from making negative comments. This can also be demonstrated by comparing the museum data emotion distribution to our general-domain source data as seen in Figure 3.1, where the sample ratio of positive instances is shown for each emotion category.

Emotion	No. of tweets	% of tweets
no code	1572	41.8%
happy	1137	30.2%
not relevant	214	5.7%
anger	57	1.5%
surprise	35	0.9%
sad	32	0.9%
happy & surprise	11	0.3%
happy & sad	9	0.2%
disgust & anger	7	0.2%
disgust	6	0.2%
sad & anger	2	0.1%
sad & disgust	2	0.1%
sad & disgust & anger	1	<0.1%

Table 3.5: Target data emotion distribution

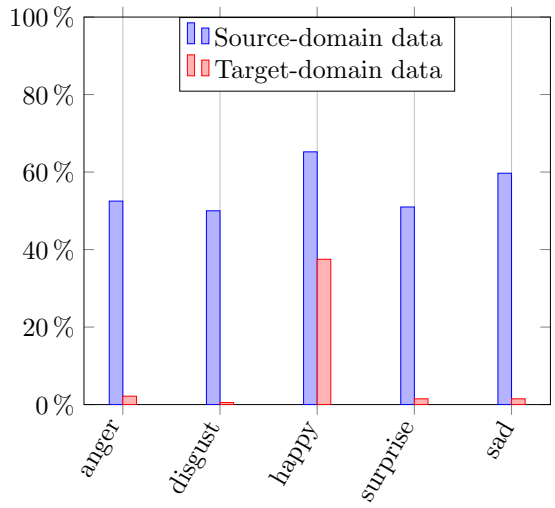


Figure 3.1: Source and target data distribution comparison

To quantify the difference between two text datasets, Kullback-Leibler (KL) divergence has been commonly used before [234]. Here we use the KL-divergence method proposed by [235], as it suggests a back-off smoothing method that deals with the data sparseness problem. Such back-off method keeps the probability distributions summing to 1 and allows operating on the entire vocabulary, by introducing a normalisation coefficient and a very small threshold probability for all the terms that are not in the given vocabulary. Since our source-domain data contains many more tweets than the target-domain data, we have randomly sub-sampled

the former and made sure the two data sets have similar vocabulary size in order to avoid biases. We removed stop words, user mentions, URL links and re-tweet symbols prior to computing the KL-divergence. Finally we randomly split each data set into 10 folds and compute the in-domain and cross-domain symmetric KL-divergence (KLD) value between every pair of folds. Table 3.6 shows the computed KL-divergence averages. It can be seen that KL-divergence between the two data sets (i.e. $\mathrm{KLD}(D_{src} \,||\, D_{tar})$) is twice as large as the in-domain KL-divergence values. This suggests a significant difference between data distributions in the two domain and thus justifies our need for domain adaptation.

Data domain	**Averaged KLD value**
$\mathrm{KLD}(D_{src} \,||\, D_{src})$	2.391
$\mathrm{KLD}(D_{tar} \,||\, D_{tar})$	2.165
$\mathrm{KLD}(D_{src} \,||\, D_{tar})$	4.818

Table 3.6: In-domain and cross-domain KL-divergence values

3.2.3 Methodology

Given the source-domain D_{src} and target-domain D_{tar}, we have one or k sets of labelled source-domain data denoted as $\left\{(\mathbf{x}_i^k, y_i^k)\right\}_{i=1}^{N_{src}^k}$ in D_{src}, where $\mathbf{x}_i^k \in \mathbb{R}^{\mathrm{D}_k}$ is the i_{th} feature vector with each element as the value of the corresponding feature and y_i^k are the emotion categories that the i_{th} instance belongs to. Suppose we have some classifiers $f_{src}^k(\mathbf{x})$ that have been trained on the source-domain data (named as the *auxiliary classifiers* in [94]) and a small set of labelled target-domain data as D_{tar}^l where $D_{tar} = D_{tar}^l \cup D_{tar}^u$, our goal is to adapt $f_{src}^k(\mathbf{x})$ to a new classifier $f_{tar}(\mathbf{x})$ based on the small set of labelled examples in D_{tar}^l, so it can be used to accurately predict the emotion class of unseen data from D_{tar}^u.

Base Classifiers

Our base classifiers are the classifiers that have been trained on the source-domain data $\left\{(\mathbf{x}_i, y_i)\right\}_{i=1}^{N_{src}}$, where $y_i \in \{1, ..., K\}$ with K referring to the number of emotion

categories. Naturally this is a multi-class classification problem, which each target-domain tweet can be classified to one of K classes. Two classic strategies for reducing the problem of multi-class classification to multiple binary classifications (i.e. $y_i \in \{-1, +1\}$) are the "one-versus-rest" approach and "one-versus-one" approach. The former builds K binary classifiers, each trained to separate one class from the rest. To predict a new instance, it chooses the class with the largest decision function value. The latter approach builds $K(K-1)/2$ classifiers and each one trains data from two classes. A voting strategy is used in classification, and it chooses the class that is voted by the most classifiers. In our work, we use Support Vector Machines (SVMs) in a "one-versus-all" setting, which trains K binary classifiers, each separating one class from the rest. We chose this as a better way of dealing with class imbalance in a multi-class scenario, and it is more computationally efficient.

Thus we train K SVM models as our base classifiers. The m_{th} SVM is trained with all the instances in the m_{th} emotion category with positive labels, and all other instances with negative labels. Given a training set of N instance-label pairs $(\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N)$ and $y_i \in \{1, ..., K\}$ where $i = 1, ..., N$, the m_{th} SVM model solves the following optimisation problem [236]:

$$\begin{aligned} \min_{w^m, b^m, \xi^m} \quad & \frac{1}{2}(\mathbf{w}^m)^T \mathbf{w}^m + C \sum_{i=1}^{N} \xi_i^m \\ \text{s.t.} \quad & (\mathbf{w}^m)^T \phi(\mathbf{x}_i) + b^m \geq 1 - \xi_i^m, \text{if } y_i = m, \\ & (\mathbf{w}^m)^T \phi(\mathbf{x}_i) + b^m \leq -1 + \xi_i^m, \text{if } y_i \neq m, \\ & \xi_i^m \geq 0, \forall (\mathbf{x}_i, y_i) \in D_{src} \end{aligned} \tag{3.1}$$

where C is the penalty parameter and $\sum_i \xi_i$ measures the total classification error. This objective function seeks a balance between the regularisation term $\frac{1}{2}(\mathbf{w}^m)^T \mathbf{w}^m$ and the training errors. x_i is assigned to the class which has the largest value of the decision function:

$$\text{argmax}_{m=1,...,K}((\mathbf{w}^m)^T \phi(x_i) + b^m) \tag{3.2}$$

where $w \in \mathbb{R}^{d+1}$ are the model parameters.

Features

The base classifiers are trained on 3 sets of features generated from the source-domain data: (i) n-grams, (ii) lexicon features, (iii) word embedding features.

N-gram models have long been used in NLP for various tasks. It is often criticised for its lack of any explicit representation of long range or semantic dependency, but it is surprisingly powerful for simple text classification with reasonable amount of training data. We used 1-2-3 grams after filtering out all the stop words, as our n-gram features. We construct 32 **Lexicon features** from 9 Twitter specific and general-purpose lexica. Each lexicon provides either a numeric sentiment score, or categories where a category could correspond to a particular emotion or a strong/weak positive/negative sentiment.

The use of **Word embedding features** to represent the context of words and concepts, has been shown to be very effective in boosting the performance of sentiment classification. Here we use a set of word embeddings learnt using a sentiment-specific method in [69] and another set of general word embeddings trained with 5 million tweets by [103]. Training on an additional set of 3 million tweets we trained ourselves did not increase performance. Pooling functions are essential and particularly effective for feature selection from dense embedding feature vectors. [69] applied the *max*, *min* and *mean* pooling functions and found them to be highly useful. We tested and evaluated six pooling functions, namely *sum*, *max*, *min*, *mean*, *std* (i.e. standard deviation) and *product*, and selected *sum*, *max* and *mean* as they led to the best performance.

Classifier Adaptation

[94] proposes a many-to-one SVM adaptation model, which directly modifies the decision function of an ensemble of existing classifiers $f^k_{src}(\mathbf{x})$, trained with one or k

sets of labelled source-domain data in D_{src}, and thus creates a new adapted classifier $f_{tar}(\mathbf{x})$ for the target-domain D_{tar}. The adapted classifier has the following form:

$$f_{tar}(x) = \sum_{k=1}^{M} \boldsymbol{\tau}^k f_{src}^k(x) + \Delta f(\mathbf{x}) \tag{3.3}$$

where $\boldsymbol{\tau}^k \in (0, 1)$ is the weight of each base classifier $f_{src}^k(\mathbf{x})$. $\Delta f(\mathbf{x})$ is the perturbation function that is learnt from a small set of labelled target-domain data in D_{tar}^l. As shown in [94] it has the form:

$$\Delta f(x) = \mathbf{w}^T \phi(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) \tag{3.4}$$

where $\mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \phi(\mathbf{x}_i)$ are the model parameters to be estimated from the labelled examples in D_{tar}^l and α_i is the feature coefficient of the i_{th} labelled target-domain instance. Furthermore $K(\mathbf{x}_i, \mathbf{x})$ is the similarity between $\mathbf{x}_i$ and $\mathbf{x}$ in the transformed feature space. $\Delta f(\mathbf{x})$ is learnt in a framework that aims to minimise the regularised empirical risk [237]. The adapted classifier $f_{tar}(\mathbf{x})$ learnt under this framework tries to minimise the classification error on the labelled target-domain examples and the distance from the base classifiers $f_{src}^k(\mathbf{x})$, to achieve a better bias-variance trade-off.

In this work we use the extended multi-classifier adaptation framework proposed by [95], which allows the weight controls $\{\boldsymbol{\tau}^k\}_{k=1}^M$ of the base classifiers $f_{src}^k(\mathbf{x})$ to be learnt automatically based on their classification performance of the small set of labelled target-domain examples. To achieve this, [95] adds another regulariser to the regularised loss minimisation framework, with the objective function of training

the adaptive classifier now written as:

$$\begin{aligned} \min_{w,\boldsymbol{\tau},\xi} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + \frac{1}{2}B(\boldsymbol{\tau})^T\boldsymbol{\tau} + C\sum_{i=1}^{N}\xi_i \\ \text{s.t.} \quad & y_i\sum_{k=1}^{M}\boldsymbol{\tau}^k f_{src}^k(\mathbf{x}) + y_i w^T\phi(\mathbf{x}_i) \geq 1-\xi_i, \\ & \xi_i^m \geq 0, \forall(\mathbf{x}_i, y_i) \in D_{src} \end{aligned} \tag{3.5}$$

where $\frac{1}{2}(\boldsymbol{\tau})^T\boldsymbol{\tau}$ measures the overall contribution of base classifiers. Thus this objective function seeks to avoid over reliance on the base classifiers and also over-complex $\Delta f(\cdot)$. The two goals are balanced by the parameter B. By rewriting this objective function as a minimisation problem of a Lagrange (primal) function and set its derivative against $\mathbf{w}$, $\boldsymbol{\tau}$, and ξ to zero, we have:

$$\mathbf{w} = \sum_{i=1}^{N}\alpha_i y_i\phi(\mathbf{x}_i), \quad \boldsymbol{\tau}^k = \frac{1}{B}\sum_{i=1}^{N}\alpha_i y_i f_{src}^k(\mathbf{x}_i) \tag{3.6}$$

where $\boldsymbol{\tau}^k$ is a weighted sum of $y_i f_{src}^k(\mathbf{x}_i)$ and it indicates the classification performance of f_{src}^k on the target-domain. Therefore we have base classifiers assigned with larger weight if they classify the labelled target-domain data well. Now given (3.3), (3.4) and (3.6), the new decision function can be formulated as:

$$\begin{aligned} f_{tar}(\mathbf{x}) &= \frac{1}{B}\sum_{k=1}^{M}\sum_{i=1}^{N}\alpha_i y_i f_{src}^k(\mathbf{x}_i) f_{src}^k(\mathbf{x}) + \Delta f(\mathbf{x}) \\ &= \sum_{i=1}^{N}\alpha_i y_i\Big(K(\mathbf{x}_i,\mathbf{x}) + \frac{1}{B}\sum_{k=1}^{M} f_{src}^k(\mathbf{x}_i) f_{src}^k(\mathbf{x})\Big) \end{aligned} \tag{3.7}$$

Comparing (3.7) with a standard SVM model $f(x) = \sum_{i=1}\alpha_i y_(+ 1, -1)K(\mathbf{x}_i, \mathbf{x})$, this multi-classifier adaptation model can be interpreted as a way of adding the predicted labels of base classifiers on the target-domain as additional features. Under this interpretation the scalar B balances the contribution of the original features and additional features. The dual form of this multi-classifier SVM can be obtained by

plugging (3.6) into the primal Lagrangian (3.5), and it can be solved by a variation of the standard minimal optimisation (SMO) algorithm proposed in [95].

3.2.4 Results and Evaluation

In this section we present the experimental results and compare our adaptation system with a set of domain-dependent and domain-independent strategies. We also investigate the effect of different sizes of the labelled target-domain data in the classification performance.

Adaptation Baselines

The baseline methods and our proposed system are the following:

- **BASE**: the base classifiers use either one set of features or all three feature sets (i.e. BASE-all). As an example, the BASE-embedding classifier is trained and tuned with all source-domain data using only word-embedding features, then tested on 30% of our target-domain data. We use the LIBSVM implementation [238] of SVM for building the base classifiers.

- **TARG**: trained and tuned with 70% labelled target-domain data. Since this model is entirely trained from the target domain, it is very hard to beat.

- **AGGR**: an aggregate model trained from all source-domain data and 70% labelled target-domain data.

- **ENSEMBLE**: combines the base classifiers in an ensemble model as proposed in [31]. Then perform classification on 30% of the target-domain data to generate new training data, as described in Section 2.2.2.

- **ADAPT**: our domain adapted models use either one base classifier trained with all feature sets (i.e. ADAPT-1-model) or an ensemble of three standalone

base classifiers with each trained with one set of features (i.e. ADAPT-3-model). We use 30% of the labelled target-domain data for classifier adaptation and parameter tuning described in Section 3.2.3.

The above methods are all tested on the same 30% labelled target-domain data in order to make their results comparable. We use an RBF kernel function with default setting of the gamma parameter γ in all the methods. For the cost factor C and class weight parameter (except the SRC-all model) we conduct cross-validated grid-search over the same set of parameter values for all the methods, for parameter optimisation. This makes sure our ADAPT models are comparable with BASE, TARG, ENSEMBLE and AGGR.

Experimental Results

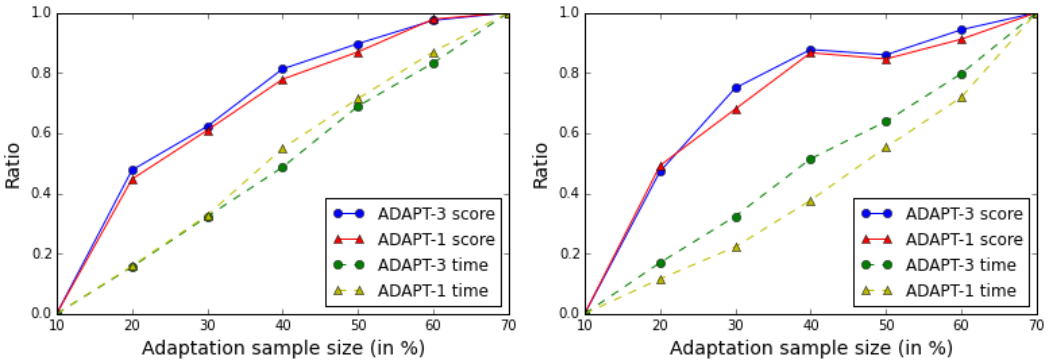
We report the experimental results in **Table 3.7**, with three categories of models: 1) in-domain no adaptation methods, i.e. BASE and TARG models, TARG being the *upper-bound* for performance evaluation; 2) the domain adaptation baselines, i.e. AGGR and ENSEMBLE and 3) our adaptation systems (ADAPT models). As can be seen the classification performances reported for emotions other than "happy" are below 50 in terms of $\mathbf{F_1}$ score with some results being as low as 0.00. This is caused by the class imbalance issue within these emotions as shown in Table 3.5 and Figure 3.1, especially for the emotion "disgust" which has only 16 tweets. We tried to balance this issue using a class weight parameter, but it still is very challenging to overcome without acquiring more labelled data than we currently have. It especially effects our domain adaptation as all the parameters in Eq.(3.5) cannot be properly optimised.

Since there are very few tweets annotated as "disgust", we decide not to consider the "disgust" emotion as part of our experiment evaluation here. As seen in Table 3.7, BASE models are outperformed significantly by all other methods (except ENSEMBLE, which performs only slightly better than the BASE models) positing

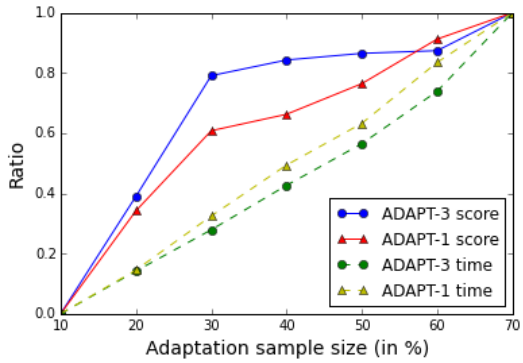
the importance of domain adaptation. With the exception of the ADAPT-3-model for "Anger", our ADAPT models consistently outperform AGGR-all and ENSEMBLE while showing competitive performance compared to the *upper-bound* baseline, TARG-all. We also observe that the aggregation model AGGR-all is outperformed by TARG-all, indicating such domain knowledge cannot be transferred effectively to a different domain by simply modelling from aggregated data from both domains. In comparison, our ADAPT models are able to leverage the large and balanced source-domain data (as base classifiers) unlike TARG, while adjusting the contribution of each base classifier unlike AGGR. When comparing our ADAPT models, we find that in most cases models adapted from multiple base classifiers beat the ones adapted from one single base classifier, even though the same features are used in both scenarios. This shows the benefit of the multi-classifier adaptation approach, which aims to maximise the utility of each base classifier.

Model	Anger			Disgust			Happy			Surprise			Sad		
	P	**R**	**F_1**	**P**	**R**	**F_1**	**P**	**R**	**F_1**	**P**	**R**	**F_1**	**P**	**R**	**F_1**
BASE-ngrams	5.77	40.91	10.11	0.49	100.0	0.97	37.62	100.0	54.67	1.46	100.0	2.87	1.50	100.0	2.96
BASE-lexicon	2.59	90.91	5.03	0.55	100.0	1.10	38.43	98.96	55.36	0.00	0.00	0.00	2.54	93.33	4.94
BASE-embedding	2.06	72.73	4.02	0.00	0.00	0.00	39.18	96.11	55.66	2.00	60.00	3.88	1.49	80.00	2.92
BASE-all	2.01	59.09	3.88	5.00	20.00	8.00	38.75	98.19	55.57	1.69	66.67	3.29	1.58	86.67	3.11
TARG-all	36.00	40.91	**38.30**	0.00	0.00	0.00	78.04	84.72	81.24	20.83	33.33	**25.64**	18.75	20.00	**19.35**
AGGR-all	10.71	27.27	15.38	33.33	20.00	25.00	64.79	86.27	74.00	5.88	11.11	7.69	4.17	20.00	6.90
ENSEMBLE	2.11	100.0	4.13	0.49	100.0	0.97	45.20	83.55	58.66	2.70	11.11	4.35	1.46	100.0	2.88
ADAPT-1-model	16.28	31.82	21.54	0.59	80.00	1.18	79.34	80.57	79.95	11.11	13.33	12.12	100.0	6.67	12.50
ADAPT-3-model	20.00	9.09	12.50	0.00	0.00	0.00	82.11	80.83	81.46	8.14	46.67	13.86	8.77	33.33	13.89

Table 3.7: Emotion Classification model performance comparison



(a) $C = 1$ (b) $C = 3$



(c) $C = 10$

Figure 3.2: Performance of each ADAPT model with $C = 1,3,10$ vs. its computation time

We can also evaluate the performance of each model by comparing its efficiency in terms of computation time. Here we report the total computation time taken for all the above methods except BASE, for the emotion "happiness", on a laptop with 2.8 GHz Intel Core i7 processor and 16 GB memory. Such computation process consists of adaptation training, grid-search over the same set of parameter values and final testing. As seen in Table 3.8, compared to other out-of-domain strategies the proposed ADAPT models are more efficient to train especially in comparison with AGGR, which is an order of magnitude more costly due to the inclusion of source-domain data. Within the ADAPT models, ADAPT-1-model requires less time to train since it only has one base classifier for adaptation.

Model	Total computation time in minutes
TARG-all	7.72
ENSEMBLE	209.72
AGGR-all	1238.24
ADAPT-1-model	26.30
ADAPT-3-model	118.41

Table 3.8: Total computation time for each classification method

Effect of Adaptation Training Sample ratios

Here we evaluate the effect of different ratios of the labelled target-domain data on the overall classification performance for the emotion "happiness". Figure 3.2 shows the normalised $\mathbf{F_1}$ scores and computation time of each ADAPT model across different adaptation training sample sizes ranging from 10% to 70% of the total target-domain data (with the same 30% held out as test data) and with the cost factor $C = 1$, 3 and 10 (as the same choices of C are used in [94] for conducting their experiment). We observe a logarithmic growth for the $\mathbf{F_1}$ scores obtained from every model, against a linear growth of computation time cost. Thus even though there is a reasonable increase in classification performance when increasing the adaptation sample size from 50% to 70%, it becomes much less efficient to train such models and we require more data, which may not be available. Since we have a trade-off between model effectiveness and efficiency here, it is appropriate to use 30% of our labelled target-domain data for classifier adaptation as we have done so in ADAPT-1-model and ADAPT-3-model. One should select the adaptation training sample size accordingly based on the test data at hand, but empirically we think 1,000 labelled target-domain tweets would be enough for an effective adaptation to classify 3,000-4,000 test tweets.

3.2.5 Conclusion

Domain adaptation for sentiment and emotion analysis across topics on Twitter is challenging due to the noisy and sparse nature of tweets. We have studied a model-

based multi-class adaptive-SVM approach for cross-domain emotion recognition and compared against a set of domain-dependent and domain-independent strategies. We evaluated our proposed system on a set of newly annotated Twitter data about museums, thus furthering research in the interdisciplinary study of social media discourse around arts and cultural experiences. We find that our adapted SVM model outperforms the out-of-domain base models and domain adaptation baselines while also showing competitive performance against the in-domain model. Moreover, in comparison to other adaptation strategies our approach is computationally more efficient especially compared to the classifier trained on aggregated source and target data. Finally, we shed light on how different ratios of labelled target-domain data used for adaptation can effect classification performance. We show there is a trade-off between model effectiveness and efficiency when selecting adaptation sample size. Our code and data[8] are publicly available, enabling further research and comparison with our approach.

In the future we would like to study how to use deep learning for domain adaptation without retraining on source domain data or fine-tuning target domain labeled data, by effectively applying teacher's knowledge learned from the source domain to the target domain. We would also like to investigate the possibility of applying multi-task learning using an auxiliary task to help the main task of cross-topic emotion classification on Twitter. Another future direction is to study how to best resolve the remarkable class imbalance issue in social media emotion analysis when some emotions are rarely expressed.

[8] `http://bit.ly/1SddvIw`

CHAPTER 4

Target-specific Sentiment Recognition

Classifying sentiment towards multiple targets in a tweet

In the previous chapter we have explored the challenge of tackling cross-domain emotion classification when we have low training resource for the target domain. In this chapter we continue our research on Twitter sentiment classification by addressing the task of target-specific sentiment recognition.

In the recent years we have seen an increasing interest in mining Twitter to assess public opinion on political affairs and controversial issues [13, 16] as well as products and brands [239]. Opinion mining from Twitter is usually achieved by determining the sentiment polarity of tweets and has mostly focused on the overall sentiment expressed in an entire tweet. However, inferring the sentiment towards specific targets (e.g. people or organisations) is severely limited by such an approach since a tweet may contain different types of sentiment expressed towards each of the targets mentioned. An early study by Jiang et al. [11] showed that 40% of classification errors are caused by using tweet-level approaches that are independent of the target. Consider the tweet:

"I will b voting 4 ***Greens*** *... 1st reason: 2 remove 2 party alt. of* ***labour*** *or* ***conservative*** *every 5 years. 2nd:* ***fracking****"*

The overall sentiment of this tweet is positive but there is a negative sentiment towards "labour", "conservative" and "fracking" and a positive sentiment towards "Greens". Examples like this are common in tweets discussing topics like politics, as is the case in the corpus of political tweets harvested prior to the UK General elections in 2015, which we present in Section 4.3. As has been demonstrated by the failure of election polls in both referenda and general elections [240], it is important to understand not only the overall mood of the electorate, but also to distinguish and identify sentiment towards different key issues and entities, many of which are discussed on social media on the run up to elections. Therefore in this chapter we will address the following research question:

RQ1: *How can we infer the sentiment towards a specific target as opposed to tweet-level sentiment? Can we find an effective approach for identifying sentiment towards multiple targets within a tweet?*

To answer this research question, we participated in a Twitter sentiment analysis challenge as our pilot research on classifying single-target sentiment. In this work we develop a set of different strategies which use either syntactic dependencies or token-level associations with the target word in combination with our phrase-level classifier to produce sentiment annotations. Then, we propose a method for multi-target specific sentiment recognition, which we develop by using the context around a target as well as syntactic dependencies involving the target. We also present a corpus of UK election tweets, with an average of 3.09 entities per tweet and more than one type of sentiment in half of the tweets, making it the most suitable dataset for this task and thus a valuable resource to the community.

4.1 Single-target-specific Sentiment Recognition using Graph Kernel

Participating in SemEval-2015[1] Task-10 Sub-task C [96], our goal is to identify the sentiment targeted towards a particular target entity within a tweet. This is closely linked to aspect-based sentiment [241] and is very important for understanding the reasons behind the manifestation of different reactions. We develop several strategies for selecting a target-relevant portion of a tweet and use it to produce a sentiment annotation. Our approach is based on using a phrase-based sentiment identification model [39] to annotate the target-relevant selections.

4.1.1 Target Relevance Through Syntactic Relations

A syntactic parser generates possible grammatical relations between words in a sentence, which are potentially useful for capturing the context around a target entity. We experimented with the Stanford parser [242] and the recently released TweeboParser [111]. TweeboParser is explicitly designed to parse tweets – supporting multi-word annotations and multiple roots – but instead of the popular Penn Treebank annotation it uses a simpler annotation scheme and outputs much less dependency type information and was therefore not deemed suitable for our purpose. We use the Stanford parser with a caseless parsing model, expected to work better for short documents. We define the target-relevant portion of a tweet as the weakly connected components of the dependency graph containing a given target-entity word.

4.1.2 Generating Per-Token Annotations

Our target-specific models use per-token sentiment annotations generated in advance by a linear SVM and random forest-based classifiers [39], using the balanced and

[1] `http://alt.qcri.org/semeval2015/task10/`

imbalanced versions of SemEval-2015 Task-10 subtask A's training data. We found the SVM outperformed the random forest classifier, with all the submission models performing best with the balanced version, and the baseline model performing best using the imbalanced training data.

4.1.3 Classification Without Dependency Relations

The simplest classification method (Baseline) identifies the target entity and only considers those tokens around it. Then the target sentiment is determined by majority voting from the token sentiments. Despite being rudimentary, we found Baseline difficult to beat when used with our per-token sentiment classifier, producing an F1-score of 46.59 with a window of 8 tokens.

4.1.4 Using Dependency Relations

Our submitted model, named Submission, builds a directed co-dependency graph from the supplied parse, trims some of the relations[2], then attempts to match it against parse trees seen previously, to capture syntactic features that may be relevant to the target's sentiment. Because subgraph isomorphism is a computationally difficult problem, we use a diffusion kernel (as in [244]) to normalise the adjacency matrix for SVM classification. We also add unigrams within the same window used for Baseline as an additional feature. Submission-Retokenized updates the result and replaces whitespace tokenisation with that used by [227], and improves the preprocessing pipeline, improving performance by +5 in F_1. Submission-Sentiment changes the structure of the dependency graph by connecting tokens to their 1-window sentiment derived from the per-token classifier, improving performance further still.

[2]We select 9 dependency relations – 'amod', 'nsubj', 'advmod', 'dobj', 'xcomp', 'ccomp', 'rcmod', 'cop' and 'acomp' which feasibly impact sentiment [243].

Experiment	$\mathbf{F_1}$ score
TwitterHawk	50.51
SUBMISSION	22.79
SUBMISSION-SENTIMENT	29.37
SUBMISSION-RETOKENIZED	27.88
BASELINE	46.59

Table 4.1: Performance comparison of our submitted sentiment classifiers.

4.1.5 Discussion

Table 4.1 shows the performance of our submitted classifiers and baseline model comparing to the best performing system - *TwitterHawk*, for SemEval-2015 Task-10 Sub-task C. The official scoring metric for this task is 2-class macro-averaged (i.e. negative and positive) F_1 score[3].

Surprisingly, our simple baseline system outperforms the 3 SUBMISSION models, which aim to construct syntactic features relevant to the target. The two best performing systems for this challenge including *TwitterHawk*, both opted to use the tweet-level target-independent approach. We think there are two potential explanations for why tweet-level models perform well for this task: 1) It may merely be the nature of this dataset containing 2382 tweets as final test data; 2) tweet-level approach indeed is the most suitable for the scenario where the tweet only mentions one single target entity.

To answer to our hypotheses, in the following sections we investigate both single-target and multi-target-specific tasks. We propose a more effective way of using the syntactic dependencies involving the target, achieving state-of-the art performance on two different datasets. We also study the relationship among tweet-level, single-target and multi-target tasks, and thus show the importance of of distinguishing target entity sentiment.

[3]Note that this isn't a binary classification task as it is still effected by the neutral tweets.

4.2 Multi-target-specific Sentiment Classification

Recent developments on target-specific Twitter sentiment classification have explored different ways of modelling the association between target entities and their contexts. Jiang et al. [11] propose a rule-based approach that utilises dependency parsing and contextual tweets. Dong et al. [1], Tang et al. [104] and Zhang et al. [105] have studied the use of different recurrent neural network models for such a task but the gain in performance from the complex neural architectures is rather unclear[4]

In the following section we introduce the multi-target-specific sentiment recognition task, building a corpus of tweets from the 2015 UK general election campaign suited to the task. In this dataset, target entities have been semi-automatically selected, and sentiment expressed towards multiple target entities as well as high-level topics in a tweet have been manually annotated. Unlike all existing studies on target-specific Twitter sentiment analysis, we move away from the assumption that each tweet mentions a single target; we introduce a more realistic and challenging task of identifying sentiment towards multiple targets within a tweet. To tackle this task, we propose TDParse, a method that divides a tweet into different segments building on the approach introduced by Vo and Zhang [103]. TDParse exploits a syntactic dependency parser designed explicitly for tweets [111], and combines syntactic information for each target with its left-right context.

We evaluate and compare our proposed system on our new multi-target UK election dataset, as well as on the benchmarking dataset for single-target dependent sentiment [1]. We show the state-of-the-art performance of TDParse over existing approaches for tweets with multiple targets, which encourages further research on the multi-target-specific sentiment recognition task.[5]

[4]They have yet to show a clear out-performance on a benchmarking dataset and our multi-target corpus, possibly because they usually require large amount of training data.

[5]The data and code can be found at `https://goo.gl/S2T1GO`.

4.3 Creating a Corpus for Multi-target-specific Sentiment in Twitter

A tweet, though constrained by its 140-character limit, often contain more than one target entity with opposite sentiments. In this section we describe the design, collection and annotation of a multi-target sentiment corpus of tweets about the 2015 UK election.

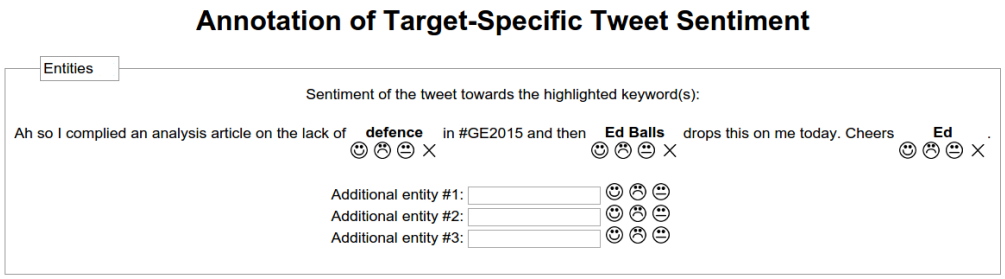


Figure 4.1: Annotation tool for human annotation of target specific sentiment analysis

4.3.1 Data Harvesting and Entity Recognition

We collected a corpus of tweets about the UK elections, as we wanted to select a political event that would trigger discussions on multiple entities and topics. Collection was performed through Twitter's streaming API and tracking 14 hashtags[6] that were obtained by using our hashtag seeding algorithm described in Appendix A. Data harvesting was performed between 7th February and 30th March 2015, to capture the ongoing discussion in the weeks running up to the election. This led to the collection of 712k tweets, from which a subset was sampled for manual annotation of target-specific sentiment. We also created a list of 438 topic keywords relevant

[6] #ukelection2015, #ge2015, #ukge2015, #ukgeneralelection2015, #bbcqt, #bbcsp, #bbcdp, #marrshow, #generalelection2015, #ge15, #generalelection, #electionuk, #ukelection and #electionuk2015

to 9 popular election issues[7] for data sampling. The initial list of 438 seed words provided by a team of journalists was augmented by searching for similar words within a vector space on the basis of cosine similarity. Keywords are used both in order to identify thematically relevant tweets and also targets. We also consider named entities as targets.

Sampling of tweets was performed by removing retweets and making sure each tweet contained at least one topic keyword from one of the 9 election issues, leading to 52,190 highly relevant tweets. For the latter we ranked tweets based on a "similarity" relation, where "similarity" is measured as a function of content overlap [245]. Formally, given a tweet S_i being represented by the set of N words that appear in the tweet: $S_i = W_i^1, W_i^2, ..., W_i^N$ and our list of curated topic keywords T, the ranking function is defined as:

$$log(|S_i|) * |W_i \in S_i \cap W_i \in T| \tag{4.1}$$

where $|S_i|$ is the total number of words in the tweet; unlike Mihalcea [245] we prefer longer tweets. We used exact matching with flexibility on the special characters at either end. TF-IDF normalisation and cosine similarity were then applied to the dataset to remove very similar tweets (empirically we set the cosine similarity threshold to 0.6). We also collected all external URLs mentioned in our dataset and their web content throughout the data harvesting period, filtering out tweets that only contain an external link or snippets of a web page. Finally we sampled 4,500 top-ranked tweets keeping the representation of tweets mentioning each election issue proportionate to the original dataset.

For annotation we considered sentiment towards two types of targets: entities and topic keywords. Entities were processed in two ways: firstly, named entities (people, locations, and organisations) were automatically annotated by combining

[7]EU and immigration, economy, NHS, education, crime, housing, defense, public spending, environment and energy

the output of Stanford Named Entity Recognition (NER) [246], NLTK NER [247] and a Twitter-specific NER [248]. All three were combined for a more complete coverage of entities mentioned in tweets and subsequently corrected by removing wrongly marked entities through manual annotation. Secondly, to make sure we covered all key entities in the tweets, we also matched tweets against a manually curated list of 7 political-party names and added users mentioned therein as entities. The second type of targets matched the topic keywords from our curated list. During test time, target entities can be extracted automatically by matching the curated topic keyword and party name lists, as well as performing named entity recognition.

4.3.2 Manual Annotation of Target Specific Sentiment

We developed a tool for manual annotation of sentiment towards the targets (i.e. entities and topic keywords) mentioned in each tweet. The annotation was performed by nine PhD-level journalism students, each of them annotating approximately a ninth of the dataset, i.e. 500 tweets. Additionally, they annotated a common subset of 500 tweets consisting of 2,197 target entities, which was used to measure inter-annotator agreement (IAA). Annotators were shown detailed guidelines[8] before taking up the task, after which they were redirected to the annotation tool itself (see Figure 4.1).

Tweets were shown to annotators one by one, and they had to complete the annotation of all targets in a tweet to proceed. The tool shows a tweet with the targets highlighted in bold. Possible annotation actions consisted in: (1) marking the sentiment for a target as being positive, negative, or neutral, (2) marking a target as being mistakenly highlighted (i.e. 'doesnotapply') and hence removing it, and (3) highlighting new targets that our preprocessing step had missed, and associating a sentiment value with them. In this way we obtained a corrected list of targets for each tweet, each with an associated sentiment value.

[8]This guidelines can be found along with our released corpus: https://goo.gl/CjuHzd

We measure inter-annotator agreement in two different ways. On the one hand, annotators achieved $\kappa = 0.345$ ($z = 92.2, p < 0.0001$) (fair agreement)[9] when choosing targets to be added or removed. On the other hand, they achieved a similar score of $\kappa = 0.341$ ($z = 77.7, p < 0.0001$) (fair agreement) when annotating the sentiment of the resulting targets. It is worth noting that the sentiment annotation for each target also involves choosing among not only positive/negative/neutral but also a fourth category 'doesnotapply'. The resulting dataset contains 4,077 tweets, with an average of 3.09 entity mentions (targets) per tweet. As many as 3,713 tweets have more than a single entity mention (target) per tweet, which makes the task different from 2015 Semeval 10 subtask C [96] and a target-dependent benchmarking dataset of Dong et al. [1] where each tweet has only one target annotated and thus one sentiment label assigned. The number of targets in the 4,077 tweets to be annotated originally amounted to 12,874. However, the annotators unhighlighted 975 of them, and added 688 new ones, so that the final number of targets in the dataset is 12,587. These are distributed as follows: 1,865 are positive, 4,707 are neutral, and 6,015 are negative. This distribution shows the tendency of a theme like politics, where users tend to have more negative opinions. This is different from the Semeval 2015/2016 dataset, which has a majority of neutral sentiment. Looking at the annotations provided for different targets within each tweet, we observe that 2,051 tweets (50.3%) have all their targets consistently annotated with a single sentiment value, 1,753 tweets (43.0%) have two different sentiments, and 273 tweets (6.7%) have three different sentiment values. These statistics suggest that providing a single sentiment for the entire tweet would not be appropriate in nearly half of the cases confirming earlier observations [11].

We also labelled each tweet containing one or more topics from the 9 election issues, and asked the annotators to mark the author's sentiment towards the topic. Unlike entities, topics may not be directly present in tweets. We compare

[9]We report the strength of agreement using the benchmarks by Landis and Koch [249] for interpreting Fleiss' kappa.

topic sentiment with target/entity sentiment for 3963 tweets from our dataset adopting the approach by Vargas et al. [20]. Table 4.2 reports the individual $c(s_{target})$, $c(s_{topic})$ and joint $c(s_{target}, s_{topic})$ distributions of the target/entity s_{target} and topic s_{topic} sentiment. While s_{target} and s_{topic} report how often each sentiment category occurs in the dataset, the joint distribution $c(s_{target}, s_{topic})$ (the inner portions of the table) shows the discrepancies between target and topic sentiments. We observe marked differences between the two sentiment labels. For example it shows the topic sentiment is more neutral (1438.7 vs. 1104.1) and less negative (1930.7 vs. 2285.5) than the target sentiment. There is also a number of tweets expressing neutrality towards the topics mentioned but polarised sentiment towards targets (i.e. we observe $c(s_{topic} = neu \cap s_{targets} = neg) = 258.6$ also $c(s_{topic} = neu \cap s_{targets} = pos) = 101.4$), and vice versa. This emphasises the importance of distinguishing target entity sentiment not only on the basis of overall tweet sentiment but also in terms of sentiment towards a topic.

$c(s_{target}, s_{topic})$		s_{topic} negative	neutral	positive	$c(s_{topic})$
s_{target}	negative	**1553.9**	258.6	118.3	1930.9
	neutral	557.6	**744.1**	137.0	1438.7
	positive	174.0	101.4	**318.1**	593.5
	$c(s_{target})$	2285.5	1104.1	573.4	3963.0

Table 4.2: Individual $c(s_{target})$, $c(s_{topic})$ and joint $c(s_{target}, s_{topic})$ distributions of sentiments

4.4 Developing a state-of-the-art approach for target-specific sentiment

4.4.1 Model development for single-target benchmarking data

Firstly we adopt the context-based approach by Vo and Zhang [103], which divides each tweet into three parts (left context, target and right context), and where the

sentiment towards a target entity[10] results from the interaction between its left and right contexts. Such sentiment signal is drawn by mapping all the words in each context into low-dimensional vectors (i.e. word embeddings), using pre-trained embedding resources, and applying neural pooling functions to extract useful features. Such context set-up does not fully capture the syntactic information of the tweet and the given target entity, and by adding features from the full tweet (as done by Vo and Zhang [103]) interactions between the left and right context are only implicitly modeled. Here we use a syntactic dependency parser designed explicitly for tweets [111] to find the syntactically connected parts of the tweet to each target. This is achieved by treating each target as the root node, and performing breath-first search to find all the tokens that its head (i.e. the syntactic parent node) either is the target word or connects to the target along any particular path. As an example in tweet *"so my latebus still sucks, but my **Ipod** isn't dead this time"*, "ipod" is the target and it has the following syntactically connected parts:

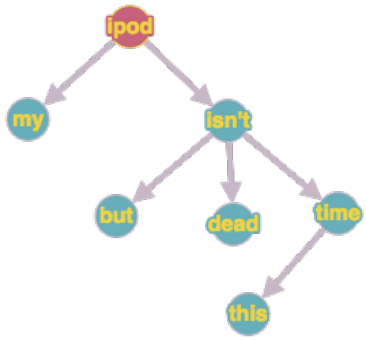


Figure 4.2: Syntactically connected parts to the target "ipod"

We then extract word embedding features from these syntactically dependent tokens $[D_1, ..., D_n]$ along its dependency path in the parsing tree to the target[11], as well as from the left-target-right contexts (i.e. $L-T-R$). Feature vectors generated

[10] As described in Section 4.3, target entities include named entities by automatic entity extraction, political party names, user mentions and topic keywords from a journalist-curated list.

[11] Empirically the proximity/location of such syntactic relations have not made much difference when used in feature weighting and is thus ignored.

from different contexts are concatenated into a final feature vector as shown in (4.2), where $P(X)$ presents a list of k different pooling functions on an embedding matrix X. Not only does this proposed framework make the learning process efficient without labor intensive manual feature engineering and heavy architecture engineering for neural models, it has also shown that complex syntactic and semantic information can be effectively drawn from tweets by simply concatenating different types of context together without the use of deep learning (other than pretrained word embeddings).

$$\begin{aligned} F &= [P(D), P(L), P(T), P(R)]; \\ \text{with } P(X) &= [f_1(X), ..., f_k(X)] \end{aligned} \tag{4.2}$$

Data set: We evaluate and compare our proposed system to the state-of-the-art baselines on a benchmarking corpus [1] that has been used by several previous studies [103, 104, 105]. This corpus contains 6248 training tweets and 692 testing tweets with a sentiment class balance of 25% negative, 50% neutral and 25% positive. Although the original corpus has only annotated one target per tweet, without specifying the location of the target, we expand this notion to consider cases where the target entity may appear more than once at different locations in the tweet, e.g.:

*"**Nicki Minaj** has brought back the female rapper. - really? **Nicki Minaj** is the biggest parody in popular music since the Lonely Island."*

Semantically it is more appropriate and meaningful to consider both target appearances when determining the sentiment polarity of "Nicki Minaj" expressed in this tweet. While it isn't clear if Dong et al. [1] and Tang et al. [104] have considered this realistic **same-target-multi-appearance scenario**, Vo et al. [103] and Zhang et al. [105] do not take it into account when extracting target-dependent contexts. Contrary to these studies we extend our system to fully incorporate the situation where a target appears multiple times at different locations in the tweet. We add

another pooling layer in (4.2) where we apply a *median* pooling function to combine extracted feature vectors from each target appearance together into the final feature vector for the sentiment classification of such targets. Now the feature extraction function $P(X)$ in (4.2) becomes:

$$\begin{aligned} P(X) = [P_{median}([f_1(X_1), ..., f_1(X_m)]), \\ ... \ ... \ , \\ P_{median}([f_k(X_1), ..., f_k(X_m)])] \end{aligned} \tag{4.3}$$

where m is the number of appearances of the target and P_{median} represents the dimension-wise *median* pooling function.

Models: To investigate different ways of modelling target-specific context and evaluate the benefit of incorporating the same-target-multi-appearance scenario, we build these models:

- **Semeval-best**: is a tweet-level model using various types of features, namely ngrams, lexica and word embeddings with extensive data pre-processing and feature engineering. We use this model as a target-independent baseline as it approximates and beats the best performing system [97] in SemEval 2015 task 10 by (+1.4) in 2-class F_1 using the same set of training data. It also outperforms the highest ranking system in SemEval 2016 task 4, Tweester [250], on the same corpus (by +4.0% in macro-averaged recall[12]) and therefore constitutes a state-of-the art tweet level baseline.

- **Naive-seg models**: **Naive-seg-** slices each tweet into a sequence of sub-sentences by using punctuation (i.e. ',' '.' '?' '!'). Embedding features are extracted from each sub-sentence and pooling functions are applied to combine word vectors. **Naive-seg** extends it by adding features extracted from the left-target-right contexts, while **Naive-seg+** extends Naive-seg by adding lexicon

[12]Official scoring metric for SemEval 2016 task 4.

filtered sentiment features.

- **TDParse models**: as described in Section 4.4.1. **TDParse-** uses a dependency parser to extract a syntactic parse tree to the target and map all child nodes to low-dimensional vectors. Final feature vectors for each target are generated using neural pooling functions. While **TDParse** extends it by adding features extracted from the left-target-right contexts, **TDParse+** uses three sentiment lexica for filtering words. **TDParse+ (m)** differs from TDParse+ by taking into account the 'same-target-multi-appearance' scenario. Both **TDParse+** and **TDParse+ (m)** outperform state-of-the-art target-specific models.

- **TDPWindow-N**: the same as **TDParse+** with a window to constrain the left-right context. For example if $N = 3$ then we only consider 3 tokens on each side of the target when extracting features from the left-right context.

4.4.2 Experimental Settings

To compare our proposed models with Vo & Zhang [103], we have used the same pre-trained embedding resources and pooling functions (i.e. *max*, *min*, *mean*, *standard deviation* and *product*). For classification we have used LIBLINEAR [251], which approximates a linear SVM. In tuning the cost factor C we perform five-fold cross validation on the training data over the same set of parameter values for both Vo and Zhang [103]'s implementation and our system. This makes sure our proposed models are comparable with those of Vo and Zhang [103].

Evaluation metrics: We follow previous work on target-dependent Twitter sentiment classification, and report our performance in accuracy, 3-class macro-averaged (i.e. negative, neutral and positive) F_1 score as well as 2-class macro-averaged (i.e. negative and positive) F_1 score[13], as used by the 2015 SemEval

[13]Note that this isn't a binary classification task; the F_1 score is still effected by the neutral tweets.

competition [96] for measuring Twitter sentiment classification performance.

4.4.3 Experimental results and comparison with other baselines

We report our experimental results in **Table 4.3** on the single-target benchmarking corpus [1], with three model categories: 1) tweet-level target-independent models, 2) target-dependent models without considering the 'same-target-multi-appearance' scenario and 3) target-dependent models incorporating the 'same-target-multi-appearance' scenario. We include the models presented in the previous section as well as models for target specific sentiment from the literature where possible.

Among the target-independent baseline models **Target-ind** [103] and **Semeval-best** have shown strong performance compared with **SSWE** [69] and **SVM-ind** [11] as they use more features, especially rich automatic features using the embeddings of Mikolov et al. [76]. Interestingly they also perform better than some of the target-dependent baseline systems, namely **SVM-dep** [11], **Recursive NN** and **AdaRNN** [1], showing the difficulty of fully extracting and incorporating target information in tweets. Basic **LSTM** models [104] completely ignore such target information and as a result do not perform as well.

Among the target-dependent systems neural network baselines have shown varying results. The adaptive recursive neural network, namely **AdaRNN** [1], adaptively selects composition functions based on the input data and thus performs better than a standard recursive neural network model (**Recursive NN** [1]). However, due to the challenges of using recursive neural networks discussed in Chapter 2, both of these models under-perform. **TD-LSTM** and **TC-LSTM** from Tang et al. [104] model left-target-right contexts using two LSTM neural networks and by doing so incorporate target-dependent information. **TD-LSTM** uses two LSTM neural networks for modeling the left and right contexts respectively. **TC-LSTM** differs from (and outperforms) **TD-LSTM** in that it concatenates target word vectors with embedding vectors of each context word. We also test the

Gated recurrent neural network models proposed by Zhang et al. [105] on the same dataset. The gated models include: **GRNN**, that includes gates in its recurrent hidden layers, **G3** that connects left-right context using a gated NN structure, and a combination of the two - **GRNN+G3**. Results show these gated neural network models do not achieve state-of-the-art performance. When we compare our target-dependent model **TDParse+**, which incorporates target-dependent features from syntactic parses, against the target-dependent models proposed by Vo and Zhang [103], namely **Target-dep** which combines full tweet (pooled) word embedding features with features extracted from left-target-right contexts and **Target-dep+** that adds target-dependent sentiment features on top of **Target-dep**, we see that our method beats both of these, without using full tweet features[14].

TDParse+ also outperforms the state-of-the-art **TC-LSTM**. It is worth mentioning here deep learning models such as LSTM, require large amount training data, especially when attention mechanism is used. We have approximated the two target-dependent LSTM models[15] proposed by [104]. Given the training data here is rather small, we have observed the instability in training resulting in inconsistent performance (even with the same initialisation). This is also reported by other users evaluating the same implementation code and set-up but running on a different machine on this corpus[16]. We show that a simple linear SVM model, can perform just as competitive or better for a small training corpus. More importantly, it is much easier and more efficient to optimise and train, and gives the same performance all the time.

When considering the 'same-target-multi-appearance' scenario, our best model - **TDParse+** improves its performance further (shown as **TDParse+ (m)** in Table 4.3). Even though **TDParse** does not use lexica, it shows competitive results

[14]Note that the results reported in Vo and Zhang [103] (**71.1** in accuracy and **69.9** in F_1) were not possible to reproduce by running their code with very fine parameter tuning, as suggested by the authors

[15]Code can be found at: `https://goo.gl/9nvNAt`.

[16]E.g. `https://goo.gl/ApD5ku` and `https://goo.gl/4H7HSv`

to **Target-dep+** which uses lexicon filtered sentiment features. In the case of **TDParse-**, which uses exclusively features from syntactic parses, while it performs significantly worse than **Target-ind**, that uses only full tweet features, when the former is used in conjunction with features from left-target-right contexts it achieves better results than the equivalent **Target-dep** and **Target-dep+**. This indicates that syntactic target information derived from parses complements well with the left-target-right context representation. Clausal segmentation of tweets or sentences can provide a simple approximation to parse-tree based models [252]. In Table 4.3 we can see our naive tweet segmentation models **Naive-seg** and **Naive-seg+** also achieve competitive performance suggesting to some extent that such simple parse-tree approximation preserves some semantic structure of text and that useful target-specific information can be drawn from each segment or clause rather than the entire tweet.

4.5 Evaluation for target-specific sentiment in a multi-target setting

We perform multi-target-specific sentiment classification on our election dataset by extending and applying our models described in Section 4.4.1. We compare the results with our other developed baseline models in Section 4.4.1, including a tweet-level model **Semeval-best** and clausal-segmentation models that provide simple parse-tree approximation, as well as state-of-the-art target-dependent models by Vo and Zhang [103] and Zhang et al. [105]. The experimentation set-up is the same as described in Section 4.4.2[17].

Data set: Our election data has a training/testing ratio of 3.70, containing 3210 training tweets with 9912 target entities and 867 testing tweets with 2675 target entities.

Models: In order to limit our use of external resources we do not include

[17]Class weight parameter is not optimised for all experiments, though better performances can be achieved here by tuning the class weight due to the class imbalance nature of this dataset.

Model	Accuracy	3 Class F_1	2 Class F_1
SSWE	62.4	60.5	
SVM-ind	62.7	60.2	
LSTM	66.5	64.7	
Target-ind	67.05	63.4	58.5
Semeval-best	67.6	64.3	59.2
SVM-dep	63.4	63.3	
Recursive NN	63.0	62.8	
AdaRNN	66.3	65.9	
Target-dep	70.1	67.4	63.2
Target-dep+	70.5	68.1	64.1
TD-LSTM	70.8	69.0	
TC-LSTM	71.5	69.5	
GRNN	68.5	65.8	61.0
G3	68.5	67.0	63.9
GRNN+G3	67.9	65.2	60.5
TDParse+	**72.1**	**69.8**	**66.0**
Target-dep+ (m)	70.7	67.8	63.4
Naive-seg-	63.0	57.6	51.5
Naive-seg	70.8	68.4	64.5
Naive-seg+	70.7	67.7	63.2
TDParse-	61.7	57.0	51.1
TDParse	71.0	68.4	64.3
TDParse+ (m)	**72.5**	**70.3**	**66.6**
TDPWindow-2	68.2	64.7	59.2
TDPWindow-7	71.2	68.5	64.2
TDPWindow-12	70.5	67.9	63.8

Table 4.3: Performance comparison on the benchmarking data [1]

Naive-seg+ and **TDParse+** for evaluation as they both use lexica for feature generation. Since most of our tweets here contain $N > 1$ targets and the target-independent classifiers produce a single output per tweet, we evaluate its result N times against the ground truth labels, to make different models comparable.

Results: Overall the models perform much poorer than for the single-target benchmarking corpus, especially in 2-class F_1 score, indicating the challenge of the multi-target-specific sentiment recognition. As seen in Table 4.4 though the feature-rich tweet-level model **Semeval-best** gives a reasonably strong baseline performance (same as in Table 4.3), both it and **Target-ind** perform worse than the target-dependent baseline models **Target-dep**/**Target-dep+** [103], indicating the need to capture and utilise target-dependent signals in the sentiment classification model. The Gated neural network models - **G3**/**GRNN**/**GRNN+G3** [105] also perform worse than **Target-dep+** while the combined model - **GRNN+G3** fails to boost performance over each separate model, presumably due to the small corpus size (suggested by its authors).

Our approximated version of two target-dependent LSTM models [104] show strong performance with **TC-LSTM*** having the highest 3-class F_1 score. As mentioned in Section 4.4.3, again we found unstable training process leading to different final performance with the same network initialisation due to insufficient amount of training data. It is also time-consuming to optimise the network even with Bayesian Optimisation.

Our final model **TDParse** achieves competitive performance in all three categories scoring the highest in 2-class F_1 and 2nd highest in 3-class F_1. This indicates that our proposed models can provide better and more balanced performance between precision and recall. It also shows the target-dependent syntactic information acquired from parse-trees is beneficial to determine the target's sentiment particularly when used in conjunction with the left-target-right contexts originally proposed by Vo and Zhang [103] and in a scenario of multiple targets per tweet. Efficiency-

Model	Accuracy	3 Class F_1	2 Class F_1
Semeval-best	54.09	42.60	40.73
LSTM	51.59	41.92	40.24
Target-ind	52.30	42.19	40.50
Target-dep	54.36	41.50	38.91
Target-dep+	55.85	43.40	40.85
GRNN	54.92	41.22	38.57
G3	55.70	41.40	37.87
GRNN+G3	54.58	41.04	39.46
TD-LSTM*	54.28	45.82	43.33
TC-LSTM*	55.74	**46.62**	42.91
Naive-seg-	51.89	39.94	37.17
Naive-seg	55.07	43.89	40.69
TDParse-	52.53	42.71	40.67
TDParse	56.45	46.09	**43.43**
TDPWindow-2	55.10	43.81	41.36
TDPWindow-7	55.70	44.66	41.35
TDPWindow-12	**56.82**	45.45	42.69

Table 4.4: Performance comparison on the election dataset

wise TDParse is efficient to optimise and does not require large amount of training resource. Our clausal-segmentation baseline - **Naive-seg** models approximate such parse-trees by identifying segments of the tweet relevant to the target, and as a result **Naive-seg** achieves competitive performance compared to other baselines.

S1	Semeval-best	Target-dep+	TDParse
Macro 3-class-F1	**50.11**	46.24	47.08
Micro 3-class-F1	**59.72**	55.82	57.47
Macro 2-class-F1	**46.59**	43.42	42.95
S2	**Semeval-best**	**Target-dep+**	**TDParse**
Macro 3-class-F1	37.15	41.81	**43.07**
Micro 3-class-F1	45.17	51.66	**52.05**
Macro 2-class-F1	37.05	39.75	**40.92**
S3	**Semeval-best**	**Target-dep+**	**TDParse**
Macro 3-class-F1	35.08	42.83	**51.26**
Micro 3-class-F1	38.16	46.05	**53.07**
Macro 2-class-F1	35.17	40.53	**50.14**

Table 4.5: Performance analysis in **S1**, **S2** and **S3** scenarios

4.5.1 State-of-the-art tweet level sentiment vs target-specific sentiment in a multi-target setting

To fully compare our multi-target-specific models against other target-dependent and target-independent baseline methods, we conduct an additional experiment by dividing our election data test set into three disjoint subsets, on the basis of number of distinct target sentiment values per tweet: the first subset (**S1**) contains tweets having one or more target entities but only one target sentiment, where the sentiment towards each target is the same; (**S2**) and (**S3**) contain two and three different types of targeted sentiment respectively (i.e. in **S3**, positive, neutral and negative sentiment are all expressed in each tweet). As described in Section 4.3.2, there are 2,051, 1,753 and 273 tweets in S1, S2 and S3 respectively.

Table 4.5 shows results achieved by the tweet-level target-independent model - **Semeval-best**, the state-of-the-art target-dependent baseline model - **Target-dep+**, and our proposed final model - **TDParse**, in each of the three subsets. We observe **Semeval-best** performs the best in **S1** compared to the two other models but its performance gets much worse when different types of target sentiment are mentioned in the tweet. It has the worst performance in **S2** and **S3**, which again emphasises the need for multi-target-specific sentiment classification. Finally, our proposed final model **TDParse** achieves better performance than **Target-dep+** consistently over all subsets indicating its effectiveness even in the most difficult scenario **S3**.

4.6 Discussion and Conclusion

In this chapter we have showed why target-specific sentiment recoginition is essential for understanding public sentiment on Twitter, and how tweet-level approaches are inadequate for such task. We studied different ways of recognising single-target-specific sentiment where each tweet mentions only one target entity. In our pilot

work, we found our graph kernel and per-token sentiment annotation based methods fail to achieve good performance. Surprisingly, our models are outperformed by a much simpler baseline method as well as the tweet-level systems.

In the subsequent work, we introduced the challenging task of multi-target-specific sentiment classification for tweets. To help answering the main research question raised at the beginning of this chapter, we have generated a multi-target Twitter corpus on UK elections which is made publicly available. We developed a much more effective approach which utilises the syntactic information from parse-tree in conjunction with the left-right context of the target. We found our proposed approach allows the syntactic target information derived from parses to complement well with the left-target-right context representation. Our approach outperforms previous methods on a benchmarking single-target corpus as well as our new multi-target election data, providing answers for **RQ1**:

RQ1: *How can we infer the sentiment towards a specific target as opposed to tweet-level sentiment? Can we find an effective approach for identifying sentiment towards multiple targets within a tweet?*

While recent work on sentiment analysis in general has largely focused on exploring deep learning models such as LSTM with attention, RNN models are not panacea for sentiment classification and in need for healthy scrutiny to give us a clear view on what works and what their limitations are. Firstly, while RNNs have an inductive bias towards sequential recency, syntax-guided linguistic structure is important[18], even for microposts such as tweets. By using a Twitter-specific parser, we have shown our proposed system can robustly utilise syntactic dependencies in tweets for our purpose, and as a result it outperforms the Recursive Neural Network

[18]During the 2017 CoNLL keynote, Chris Dyer argued language is inherently hierarchical, and syntactic recency is a preferable inductive bias to sequential recency: `http://www.conll.org/keynotes-2017`.

models and is as competitive as the LSTM and attention models. Secondly, despite the good performance, because our system uses simple linear SVM, it makes the learning process much more efficient than the neural models which often require heavy architecture engineering and time-consuming optimisation. Lastly as mentioned in Section 4.4.3, due to the insufficient amount of training data, we found the LSTM and attention models having unstable training process even with the same initilisation. On the contrary, our system gives consistent performance given the same search space, and does not need large amount of labeled training data which is not available for some domains.

In Section 4.5.1 we have showed our tweet-level model performing the best for tweets containing the same target sentiment type while it is the worst when different types of targeted sentiment are mentioned in the tweet. Our proposed multi-target system outperforms two other target-independent and target-dependent models, for tweets containing two or three different target sentiments. This not only answers the our hypotheses raised at Section 4.1.5 on why simple tweet-level methods achieving the best performance for the SemEval-2015 Task-10 competition, but also shows the need for the multi-target model as single-target is not always sufficient. Future work could investigate sentiment connections among all targets appearing in the same tweet[19] as a multi-target learning task, as well as a hybrid approach that applies either Semeval-best or TDParse depending on the number of targets detected in the tweet. There is a lot of scope for jointly learning sentiments for multiple targets in our data. It is also worth evaluating and comparing our proposed system with RNN models on a much larger corpus.

We have addressed the data quality issue caused by social spamming in Chapter 3, and two different problems on Twitter sentiment/emotion analysis in the previous chapter and this chapter. In the next chapter, we change our research angle to the topical clustering of tweets.

[19]With the application in the financial markets, we also would like to study sentiment connections among target entities in the same data set.

CHAPTER 5

Topical Clustering of Tweets

A hierarchical topic modelling approach

In the previous chapter we have discussed our work on target-specific sentiment analysis for tweets. In this chapter, we change our research angle to effectively group tweets to a number of clusters, with each cluster representing a topic, story or event. We can also cluster tweets containing the same sentiment towards a topic/entity on a day, with each cluster assumed to represent a common theme or reason underlying the particular choice of sentiment. Therefore this chapter serves as a bridge between our multi-target-specific sentiment research described in Chapter 4 and the subsequent work on tweet summarisation which is presented in Chapter 6.

5.1 Introduction

In recent years social media platforms are increasingly being used as data sources to collect all kinds of updates posted by people. Updates that are of interest range from journalistic information that news practitioners can utilise for news gathering and reporting [253, 254], as well as opinions expressed by people towards a broad range

of topics. While social media is a rich resource to shed light on public opinion and to track newsworthy stories ranging from political campaigns to terrorist attacks, it is often difficult for humans to keep track of all the relevant information provided by the large volumes of data. Automatic identification of topics can help to produce a manageable list that is easier to digest for users, enabling for instance identification of real-world events among those topics.

In contrast to the well-studied task of Topic Detection and Tracking [113], which is concerned with topic detection from newswire articles, detecting topics in social media such as Twitter not only has all the issues of conventional document clustering such as scalability to large datasets, ability to work with high-dimensional data and reliance on the user pre-defined number of clusters [255], it also poses the challenges of dealing with unmoderated, user-generated content. This presents caveats such as inconsistent vocabulary across different users as well as the brevity of microposts that often lack sufficient context. As a consequence, traditional document clustering approaches using bag-of-words representation and topic models relying on word co-occurrence fall short of achieving competitive performance. Therefore, we ask the following research question listed in Chapter 1:

RQ2: *Can we develop a system to effectively group tweets to a number of clusters, with each cluster representing a thematic topic?*

To answer the above question, in this chapter, we present a two-stage hierarchical topic modelling system shown in Figure 5.1, which: 1) uses a collapsed Gibbs Sampling algorithm for the Dirichlet Multinomial Mixture model (GSDMM) [23] for tweet clustering; 2) aggregates each tweet cluster to form a virtual document; 3) applies the second stage of topic modelling to the virtual documents but this time incorporates word embeddings as latent features (LFLDA) [138]. This not only alleviates the noisy nature of tweets but also generates thematic and interpretable

topics. Finally we conduct extensive evaluation on two datasets, using clustering evaluation metrics as well as topic model quality metrics. We compare our proposed approaches with other clustering-based methods and topic models, reporting the best scores in both clustering performance and topic coherence.

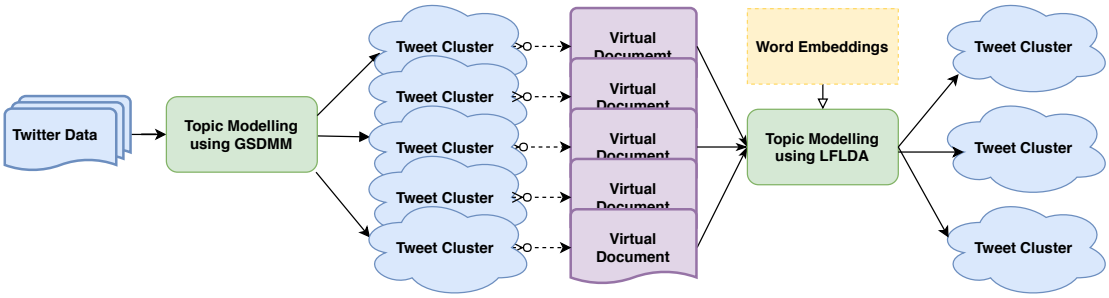


Figure 5.1: Overview of the proposed topic modelling system

5.2 Methodology

As described in Chapter 2, many studies have tried to tackle the challenge of clustering tweets into topics using different strategies, and yet it is still proven to be a difficult task to solve. Inspired by the two-stage online-offline approach in Twitter event detection studies [117, 115], we propose a two-stage hierarchical topic modelling system consisting of two state-of-the-art topic models, namely GSDMM [23] and LFLDA [138], with a tweet-pooling step streamlining the whole clustering process.

In the collapsed Gibbs Sampling algorithm for the Dirichlet Multinomial Mixture model [23] (GSDMM), the probability of a document belonging to a cluster is proportional to: 1) the cluster size; 2) the similarity between the document and the cluster (defined by the frequency of each word of the document in the cluster), which represents the two goals of clustering: Completeness and Homogeneity. After the initialisation step where documents are randomly assigned to K clusters, in each iteration it re-assigns a cluster to each document in turn according to the

conditional distribution: $p(z_d = z|\vec{z}_{\neg d}, \vec{d})$, where the documents $\vec{d}$ are observed, cluster assignments $\vec{z}$ are latent, and $\neg d$ means the cluster label of document d is removed from $\vec{z}$. [256] shows the probability of document d choosing the cluster z_d given the information of other documents and their cluster labels as follows:

$$p(z_d = z|\vec{z}_{\neg d}, \vec{d}, \alpha, \beta) \propto (m_{z,\neg d} + a)\frac{\prod_{w \in d}\prod_{j=1}^{N_d^w}(n_{z,\neg d}^w + \beta + j - 1)}{\prod_{i=1}^{N_d}(n_{z,\neg d} + V\beta + i - 1)} \tag{5.1}$$

where m_z is the number of documents in cluster z, n_z is the number of words in cluster z, n_z^w is the number of occurrences of word w in cluster z, N_d^w is the number of occurrences of word w in document d, V is the number of words in the vocabulary, α and β are two parameters to select. Therefore at each iteration it updates three count variables, namely m_z, n_z and n_z^w, to record the information of each cluster and thus resigns a cluster to a document accordingly[1]. Given its proven record on clustering tweets, we use GSDMM as the first stage of topic modelling and set K to be a very large number which allows GSDMM to automatically infer the final number of clusters.

As shown in Figure 5.1, we then assign every tweet to its corresponding cluster and aggregate each cluster to form a virtual document that consists of every tweet in that cluster. This pooling step is very similar to previous work [130, 131, 132], with the difference that it does not use any metadata which may not be available always (e.g. not every tweet mentions a hashtag or named entity).

Finally we apply the second stage of topic modelling to the previously generated virtual documents. Here we are motivated to take advantage of word embeddings [77] trained on a large external corpus which have been shown to perform well in various NLP tasks, and combine it with topic models. [138] achieves this by replacing its topic-word multinomial distribution with a two-component mixture of a Dirichlet multinomial component estimated from our smaller corpus and a latent

[1] As comparison, common similarity-based methods like K-means and Hierarchical Agglomerative clustering usually represent the documents with the vector space model.

feature representation trained on a large external corpus (i.e. the word embedding component). The model uses a topic indicator z_{d_i} and a binary indicator s_{d_i} for determining whether the word w_{d_i} is to be generated by which component. As a result, each word is modelled by either the Dirichlet multinomial distribution or the probability estimated by using word embeddings with respect to the sampled topic. We choose the better performing LFLDA model for our second-stage of topic modelling. Thus each tweet is assigned a topic with the highest topic proportion[2] given the virtual document cluster that it is in.

5.3 Datasets

We compare this two-stage system with aforementioned approaches on two datasets, with different characteristics that help us generalise our results to different topic modelling tasks:

- A first story detection (FSD) corpus [2] collected from the beginning of July to mid-September 2011. We downloaded the tweets using the Twitter search API[3] with the provided tweet IDs, obtaining 2204 tweets with each tweet annotated as one of 27 real-world stories such as "Death of Amy Winehouse" and "Terrorist attack in Delhi". It has some overlap of stories as well, e.g. four of the stories are related to the London riots in 2011, makes it also applicable to the task of sub-story detection.

- A large-scale event detection (ED) corpus [140], collected during October and November of 2012. Using Wikipedia and crowdsourcing as well as event detection methods [120, 257], it generated 150,000 tweets over 28 days covering more than 500 events. Each event label represents a specific topic or story line, e.g. "British prime minister David Cameron and Scottish first minister

[2]Topic proportion: the proportion of words in document d that are assigned to topic t or the topic probabilities of a document, i.e. $p(t|d)$

[3]`https://dev.twitter.com/rest/public/search`

Alex Salmond agree a deal". After retrieving 78,138 tweets we decide to use the first five days of data for evaluation, resulting in five sets of *tweets/labels*: *3330/32, 2083/41, 6234/48, 2038/36 and 3468/43*.

5.4 Evaluation

To investigate the performance of our proposed hierarchical topic modelling system for effectively clustering tweets, we compare it against: 1) six topic models including four state-of-the-art standalone Twitter topic models, 2) hierarchical clustering methods using learnt topic proportions as features, and 3) three neural-embedding-based clustering approaches. Experiments are conducted on two datasets. Moreover, document clustering metrics as well as topic model quality metrics are used for evaluation.

5.4.1 Experimental setup

Compared Methods: Both topic modelling and document clustering methods are evaluated. The topic modelling methods are:

- **OLDA** [258]: An online variational Bayes (VB) algorithm for LDA, based on online stochastic optimisation.
- **TOLDA** [259]: An online version of LDA specific for tracking trends on Twitter over time. Due to the limitation of the FSD corpus, this method is only evaluated in the event detection data [140].
- **GSDMM** [23]: A collapsed Gibbs Sampling algorithm for the Dirichlet Multinomial Mixture (DMM) model, proven to work well for short texts.
- **LFTM** [138]: Consists of **LFLDA** which is an extension of LDA by incorporating word embeddings, and **LFDMM** that integrates such word embeddings information into DMM.

- **LCTM** [139]: A latent concept topic model, where each latent concept is a localised Gaussian distribution over the word embedding space.

For the above models we assign the topic with the highest topic proportion to each tweet.

As for document clustering baseline methods, we use the learnt topic proportion from the above topic models as feature for each tweet and apply a clustering algorithm, e.g. **OLDA+HC**. We also evaluate three neural-embedding based clustering approaches:

- **GloveWR** [126]+**HC**: Represents sentences by a weighted average of word vectors and modified by PCA. It was reported to achieve good performance on a Twitter textual similarity corpus. Here we use *GloVe* [78] pretrained from 2 billion tweets for word vectors.

- **STV** [123]+**HC**: Skip-Thought Vectors (STV) trains an encoder-decoder model that tries to reconstruct the surrounding sentences of an encoded passage. We use their pretrained encoder model to generate tweet representations for clustering.

- **Tweet2Vec+HC** [124]: Uses character-based tweet embeddings (i.e. Tweet2Vec [125]) and outperforms the winner [118] of the 2014 SNOW breaking news detection competition[4][5] which was defined as a topic detection task.

All document clustering baselines employ a hierarchical agglomerative clustering algorithm as it is proven to be effective in [124]. We also conducted extensive experiments using Affinity Propagation [260] since it is reported to be the most effective for clustering tweets in [119], but decided not to include the results here due to its very poor performance in most cases.

[4] `http://www.snow-workshop.org/2017/challenge/`
[5] Their data is not evaluated due to its lack of annotated tweets.

The same preprocessing steps are applied to all methods to reduce the noise level. This includes removing hashtag symbols, URL links, user mention symbols and punctuation as well as lower-casing and the tokenisation of each tweet. For **LFTM** and **LCTM**, words that are out of the word embedding vocabulary are removed as is required for each respective model.

Experimental Settings: **GSDMM** infers the number of clusters automatically based on a pre-defined upper bound, we set this initial number to 100 (which is a large number comparing to the true number of clusters). For all other topic models including the ones in our proposed system we set the number of topics, $K = 100$, even if they are in the second stage of topic modelling. We use *GloVe*[6] word embedding representation for **LFTM** and **LCTM**.

For **LFTM** we empirically set $\beta = 0.2$, $\lambda = 0.6$ for processing tweets; and $\beta = 0.1$, $\lambda = 0.6$ for virtual documents in the second stage of topic modelling. The number of latent concepts S in **LCTM** is set to 500. The number of iterations in **GSDMM** is set to 100. Other parameters are kept to their default settings.

For **Tweet2Vec+HC** we directly use the Tweet2Vec model from [125] trained using 2 million tweets, also the same hierarchical clustering algorithm implementation from *fast-cluster* library [261]. Hierarchical clustering requires to choose a distance metric, linkage method and criterion in forming flat clusters. We evaluate the performance of different linkage methods and a wide range of distance metrics, using the Cophenetic Correlation Coefficient (CPCC) [262] and mean Silhouette Coefficient [263][7] on a validation dataset containing 9770 tweets, and pick the best performing combination. We specifically cut the tree at the level that generates 100 clusters. This way we make sure our comparisons are reasonable and unbiased. Additionally we also search and evaluate the optimal settings in a grid-search set-up without cutting the tree, and as a result the model generates a large number

[6] `https://nlp.stanford.edu/projects/glove/`

[7] It is a cluster validity index, was found to be the most effective among 30 validity indices for measuring the quality of the produced clusters [264].

clusters.

5.4.2 Tweet Clustering Evaluation

With topic models, we can represent each tweet with its topic distribution $p(topic|tweet)$. Hence we can evaluate the performance of each topic model on a document clustering task, by using the topic proportion directly as the final cluster assignment or indirectly as feature representations for a further round of clustering or topic modelling. We then compare the resulting clusters to the true cluster labels in two datasets.

Evaluation Metrics: We use Purity (P), Homogeneity (H), Completeness (C), V-Measure (V), and Adjusted Mutual Information (AMI) as our evaluation metrics. Purity is simply measured by counting the number of correctly assigned documents and dividing by the total number of documents, with each cluster being assigned to the class that is most frequent in the cluster. Defined in [265], Homogeneity measures the extent to which each cluster contains only documents of the same ground truth label while Completeness measures the extent to which all documents of a given true label are assigned to the same cluster. V-Measure is the harmonic mean of Homogeneity and Completeness.

Adjusted Mutual Information (AMI) [266] is an adjustment of the Mutual Information (MI) and Normalised Mutual Information (NMI) to account for chance. More specifically AMI subtracts the expectation value of the MI, so that the AMI is zero when two different clusterings are random, and one when two clusterings are identical:

$$\Delta AMI(U,V) = \frac{MI(U,V) - E\{MI(U,V)\}}{\max\{H(U),H(V)\} - E\{MI(U,V)\}} \tag{5.2}$$

where $H(U) = -\sum_{i=1}^{R} P(i)\log P(i)$ is the entropy of the clustering U. It accounts for the fact that the MI or NMI score is generally higher with larger number of clusters (e.g. they would give a high score for a clustering method that recognise

each tweet as a cluster). All the aforementioned scores range from 0.0 (worst) to 1.0 (best). We only report the AMI score for the Event Detection corpus for clarity and concision.

Results: Table 5.1 presents the performance of the different methods on the FSD corpus. Among the standalone topic models, GSDMM outperforms the others by a large margin however it generates 17 more clusters than the ground truth (44 comparing to 27). For the hierarchical clustering methods, we observe the neural-embedding based models generate large number of clusters and thus very poor results especially in Completeness and AMI. Among the two-stage topic modelling methods, all have improved performance over GSDMM alone except LFDMM. The proposed GSDMM+LFLDA proved to achieve consistent best performance over the important metrics including V-Measure and AMI as well as the closest number of clusters to the ground truth. It is also worth mentioning a Hierarchical Dirichlet Process (HDP) model is proposed in [267] and evaluated on the same FSD corpus. Our GSDMM+LFLDA system outperforms their best result by 20.4% in AMI.

Table 5.2 shows how each method performing over a 5-day stretch on the ED corpus [140]. GSDMM again performs the best among the standalone topic models, except for day-2 where it is beaten by OLDA by a small margin. OLDA showing surprisingly good performance across the board, credits to the online nature of its optimisation. The models that incorporate word embeddings, namely LFLDA, LFDMM and LCTM, show inconsistent performance over the two datasets. We also observe LFDMM has the tendency to generate relatively small number of clusters (even with the predefined $K = 100$). Different to what is reported in [139], we found that LCTM performs worse than LFLDA consistently[8], potentially caused by the noisy nature of tweets and its adverse effect on constructing latent concepts. As for the two online topic models, in general they perform reasonably well for this task. Interestingly we find Twitter Online LDA (TOLDA) performs worse than OLDA on

[8]We have also evaluated LCTM with number of concepts setting to 600 and 1000, however we observed little difference in the performance.

the ED corpus, due to the large number of clusters it assigns to the tweets.

Across the two datasets, we observe mixed results by employing hierarchical clustering using topic proportions as features. In many cases it is showing to give almost equivalent performance than using any topic model alone. This shows by simply using topic proportion as features for clustering is not a promising approach. We also see among the neural-embedding based approaches, Skip-Thought Vectors (STV) + HC performs the best but in most of the cases they perform worse than the topic models. When we tried cutting the tree to generate 50 clusters (which is closer to the true number of clusters)[9], we found there is no noticeably difference in clustering performance and in many case the performance drops.

Our two-stage topic modelling methods have shown to be rather effective in improving clustering performance, as only in 2 out of the 34 cases over the two datasets we have seen performance drop when comparing to either one of the topic models employed by the method (i.e. TOLDA+OLDA performs worse than OLDA at day-2, and GSDMM+LFDMM performs worse than LFDMM on the FSD corpus). This shows the promising result of using our proposed hierarchical topic modelling process with a pooling step. The proposed GSDMM+LFLDA proved to achieve consistent best performance over different datasets except at day-4 of the ED corpus it is beaten by GSDMM+OLDA.

5.4.3 Topic Coherence Evaluation

Here we examine the quality of our hierarchical topic modelling system[10] by topic coherence metrics. As described in Chapter 2 such metrics determine how semantically "cohesive" the topics inferred by a model are, by measuring to what extent the top topic words or the words that have high probability in each topic are semantically coherent. This includes using word/topic intrusion [143], Pointwise Mutual

[9] All clustering settings are re-tuned in the validation set.

[10] LCTM is not evaluated here since its topic is defined as a distribution over latent concepts, not over words.

Model	*N*	P	H	C	V	AMI	Silhouette
OLDA	52	0.894	0.874	0.679	0.764	0.656	0.444
GSDMM	44	**0.968**	**0.970**	0.815	0.886	0.802	**0.988**
LFLDA	92	0.895	0.881	0.728	0.797	0.704	0.340
LFDMM	15	0.812	0.764	0.744	0.754	0.735	0.846
LCTM	93	0.937	0.933	0.557	0.697	0.515	0.280
OLDA+HC	42	0.890	0.881	0.724	0.795	0.707	0.449
LFLDA+HC	53	0.900	0.863	0.765	0.811	0.749	0.364
LFDMM+HC	16	0.819	0.784	0.750	0.766	0.740	0.819
LCTM+HC	90	0.950	0.944	0.580	0.718	0.541	0.271
GloveWR+HC	100	0.565	0.499	0.274	0.354	0.196	0.025
STV+HC	100	0.645	0.561	0.546	0.553	0.504	0.067
Tweet2Vec+HC	100	0.441	0.295	0.275	0.285	0.193	0.016
GSDMM+OLDA	26	0.870	0.866	0.885	0.876	0.859	0.708
GSDMM+LCTM	33	0.952	0.951	0.864	0.906	0.856	0.858
GSDMM+LFLDA	26	0.960	0.954	**0.909**	**0.931**	**0.904**	0.795
GSDMM+LFDMM	8	0.316	0.044	0.547	0.081	0.035	0.104

Table 5.1: Document clustering performance on the FSD corpus [2] (N=Number of resulting clusters; P=Purity; H=Homogeneity; C=Completeness; V=V-measure; AMI=Adjusted Mutual Information)

Model	Day-1		Day-2		Day-3		Day-4		Day-5	
	N	AMI	*N*	AMI	*N*	AMI	*N*	AMI	*N*	AMI
OLDA	58	0.775	45	0.831	72	0.374	55	0.535	55	0.525
TOLDA	100	0.560	100	0.575	100	0.314	100	0.409	100	0.397
GSDMM	46	0.827	53	0.824	53	0.550	51	0.649	42	0.672
LFLDA	97	0.698	88	0.752	99	0.365	97	0.520	98	0.510
LFDMM	8	0.420	15	0.485	14	0.310	13	0.412	11	0.331
LCTM	94	0.583	83	0.672	100	0.301	99	0.419	97	0.406
OLDA+HC	39	0.791	40	0.833	62	0.366	45	0.557	49	0.528
TOLDA+HC	99	0.561	100	0.574	100	0.313	100	0.421	100	0.405
LFLDA+HC	32	0.732	51	0.720	82	0.373	75	0.519	68	0.529
LFDMM+HC	8	0.422	15	0.482	14	0.311	13	0.408	11	0.330
LCTM+HC	66	0.653	80	0.716	9	0.030	8	0.078	10	0.167
GloveWR+HC	100	0.212	100	0.256	100	0.117	100	0.232	100	0.168
STV+HC	100	0.327	100	0.484	100	0.238	100	0.423	100	0.418
Tweet2Vec+HC	100	0.288	100	0.360	100	0.194	100	0.256	100	0.242
TOLDA+OLDA	32	0.807	34	0.826	35	0.509	38	0.594	35	0.634
TOLDA+LFLDA	48	0.728	46	0.789	40	0.441	41	0.610	35	0.634
TOLDA+LCTM	45	0.728	45	0.795	58	0.397	48	0.521	47	0.531
GSDMM+OLDA	26	0.842	34	0.827	38	0.620	25	**0.759**	26	0.715
GSDMM+LFLDA	28	**0.870**	29	**0.834**	30	**0.681**	27	0.757	22	**0.752**
1 GSDMM+LCTM	41	0.835	39	0.825	43	0.644	39	0.703	35	0.681

Table 5.2: Document clustering performance (AMI only) on the Event Detection corpus

Information (PMI) [147] or Normalised PMI (NPMI) [144, 150]. For evaluating our proposed models, we adopt the automatic word intrusion method [144] as well as the word embedding-based topic coherence metric [152], which is shown to have a high agreement with human judgments for tweets.

For computing the word intrusion metric, following the findings in [151] we use a large Twitter corpus collected between 2014 and 2016 as background dataset to extract PMI and conditional probabilities of word pairs as features. For any given word pair, the target value of a intruder word is assigned with 2 and normal topic words are assigned with 1. We then use these features along with corresponding target values to train a SVM^{rank} for identifying the intruder word or the word that has the highest predicted ranking score, as proposed in [144]. The final score is averaged over 10 iterations of cross validation. A more detailed description of the word intrusion task can be found in Section 2.3.3. We run this classification task 3 times, and take the average score as the final word intrusion for each model. For the ED corpus, we average all the results over the 5-day period. As shown in table 5.3 GSDMM+LFLDA has the highest word intrusion scores for both datasets.

For computing the word embedding-based coherence metric we use two pre-trained word embedding models learnt from Twitter data[11], resulting in two metrics G-T-WE (GloVe) and W-T-WE (Word2Vec) based on the cosine similarity between topic word pairs. We also adopt the approach in [148], computing coherence for top-5/10/15/20 words and then take the mean over the 4 values. As shown in Table 5.4, GSDMM+LFLDA achieves the best topic coherence in 3 out of 4 cases, with TOLDA+OLDA outperforming the others for W-T-WE on the ED data. When we compare the the two-stage topic modelling approach (i.e. TOLDA+* or GSDMM+*) to its respective topic model used in the first stage (i.e. TOLDA or GSDMM), we observe in 10 out of 12 cases its topic coherence has improved. Though our results for coherence are not perfect, it is demonstrated the usefulness of ag-

[11]The GloVe model was trained using 2 billion tweets while the Word2Vec model was trained on 5 million tweets using the skip-gram algorithm.

gregating first round tweet clusters into virtual documents without the use of any metadata and then performing second round of topic modelling. As a result it is able to create not only more discriminative but also more coherent clusters.

Model	Word Intrusion	
	FSD	Event Detection
OLDA	0.059	0.104
TOLDA		0.178
GSDMM	0.062	0.194
LFLDA	0.054	0.075
TOLDA+OLDA		0.155
TOLDA+LFLDA		0.151
GSDMM+OLDA	0.071	0.150
GSDMM+LFLDA	**0.115**	**0.213**

Table 5.3: Averaged word intrusion score for both datasets

Model	Topic Coherence			
	FSD		Event Detection	
	G-T-WE	W-T-WE	G-T-WE	W-T-WE
OLDA	0.217	0.123	0.302	0.135
TOLDA			0.329	0.141
GSDMM	0.277	0.121	0.363	0.132
LFLDA	0.275	0.108	0.323	0.127
TOLDA+OLDA			0.349	**0.154**
TOLDA+LFLDA			0.371	0.137
GSDMM+OLDA	0.282	0.142	0.349	0.150
GSDMM+LFLDA	**0.315**	**0.144**	**0.385**	0.142

Table 5.4: Averaged topic coherence for both datasets

A recent study by Feng et al. [268] introduced a metric named **topic mixing degree** (TMD), which measures to what extent a generated topic is a mixture of several topic themes (i.e. a multi-theme topic). They use word vectors along with cosine similarity to compute the topic similarity in the entire topic model containing K topics, as seen in (5.3). The higher the similarity, the more likely the model has more multi-theme topics.

$$\Delta TMD(w) = \sum_{k_i} \sum_{k_j} cosine\left(w_{k_i}, w_{k_j}\right) / \left|\mathbf{w}\right|^2 \tag{5.3}$$

Here we compute the topic mixing degree for top-5/10/15/20 topic words of our

models using the GloVe Twitter word vectors (i.e. G-T-WE), and then take the mean over the 4 values as is done in computing the topic coherence. As seen in Table 5.5 our proposed system GSDMM+LFLDA has the lowest TMD scores for both datasets showing it is the least likely to contain multi-theme topics comparing to other methods. We also observe the TMD scores increased from TOLDA and GSDMM to TOLDA+OLDA and GSDMM+OLDA respectively, implying in these cases the second stage of topic modelling have indeed introduced more topic themes in some of their topics. Overall we can conclude the proposed GSDMM+LFLDA system generate meaningful and coherent topics with each topic containing a single dominant theme, as is demonstrated in the following section.

Model	**Topic Mixing Degree**	
	FSD	Event Detection
OLDA	0.261	0.286
TOLDA		0.281
GSDMM	0.296	0.267
LFLDA	0.252	0.294
TOLDA+OLDA		0.297
TOLDA+LFLDA		0.256
GSDMM+OLDA	0.314	0.303
GSDMM+LFLDA	**0.240**	**0.251**

Table 5.5: Averaged topic mixing degree for both datasets

5.4.4 Qualitative Evaluation of Topics

We also present a set of randomly selected example topics generated by GSDMM+LFLDA, on both the first story detection (FSD) corpus and the first day of the event detection (ED) corpus, as seen in Table 5.6 and Table 5.7. Each detected topic is presented with its top-10 topic words, and is matched with the corresponding topic description or story from the ground truth (given by the creators of these data sets), as well as a sample tweet retrieved using the topic keywords.

As shown in Table 5.6 and Table 5.7, words in obtained topics are mostly coherent and well aligned with a ground-truth topic description. We can also discover more useful information with regard to the corresponding real-world story, by

simply looking at its topic words. For example, in the first topic of Table 5.6 we see the Twittersphere has mentioned 'Amy Winehouse' and 'death' along with the word 'drug'. This information may have been missed if one only chooses to read a set of randomly sampled tweets mentioning 'Amy Winehouse'.

Detected topic	Corresponding topic description	Sample tweet
amy winehouse rip amywinehouse die dead sad dy talent drug	Death of Amy Winehouse.	jesus, amy winehouse found dead. v sad #winehouse
tottenham riot police news fire shoot car london north thur	Riots break out in Tottenham.	RT @itv_news: Police cars set on fire in Tottenham, north London, after riots connected to the shooting of a young man by police on Thur ...
mars water nasa flow found evidence may scientist saltwater liquid	NASA announces discovery of water on Mars.	RT @CalebHowe: NASA reporting live right now that they have circumstantial evidence for flowing, liquid water on Mars.
house debt bill pass us vote ceiling the representatives raise	US increases debt ceiling.	RT @politico: On Monday evening the House passed a bill to raise the debt ceiling, 269 to 161.
delhi high blast court outside injured explosion attack kill bomb	Terrorist attack in Delhi.	Bomb Blast outside of High Court Delhi just few minutes ago. http://t.co/MejKWlC
pipeline fire kenya least kenyans people gasoline kill dead lunga	Petrol pipeline explosion in Kenya	RT @AKenyanGirl: RT @CapitalFM_kenya: Dozens suffer burns in Kenya #Pipeline fire in Lunga Lunga, Nairobi. Firefighters battling inferno ...

Table 5.6: Example topics detected on FSD corpus

5.5 Conclusions and Future Work

In addition to the existing issues in conventional document clustering and topic modelling, inferring thematic topics in tweets is more challenging due to the short and noisy nature of tweets. In this chapter we proposed a two-stage hierarchical topic modelling system, named GSDMM+LFLDA, that leverages a state-of-the-art Twitter topic model, a topic model with word embeddings incorporated and a tweet pooling step without the use of metadata in any form. We performed extensive experiments on two Twitter corpora, in order to answer the main research question of this chapter:

Detected topic	Corresponding topic description	Sample tweet
merkel angela greece visit athens merkels greek chancellor protests protest	An estimated 25,000 protest in Athens as German Chancellor Angela Merkel visits Greece.	thousands protest merkel s greece visit http://t.co/sXGTX3jE
syrian plane turkey passenger turkish land ankara force syria intercepts	A Syrian passenger plane is forced by Turkish fighter jets to land in Ankara due to the allegations of carrying weapons.	BreakingNews: Turkish fighter jets force Syrian passenger plane to land in Ankara: Anadolu Agency
malala yousafzai taliban activist pakistan shot girl attack bullet shooting	Malala Yousafzai, a 14 year old activist for women's education rights is shot by Taliban gunmen in the Swat Valley.	Taliban Says It Shot Pakistani Teen, Malala Yousafzai, For Advocating Girls Rights... http://t.co/EjFR5in4
lenovo hp pc top market battle spot computerworld gartner shipments	HP and Lenovo battle for top spot in PC market of Computerworld.	HP, Lenovo battle for top spot in PC market - Computerworld http://t.co/zwzPdN8Q #googlenews
merger eads bae systems aerospace plans talks cancel defence firms	BAE and EADS announce their merger talks are cancelled over political disagreements.	BAE-EADS merger plans are 'off': Aerospace and defence firms BAE and EADS have cancelled their planned merger, t... http://t.co/UYFOiysX
pussy riot court appeal moscow member one freed russian punk	A court in Moscow, Russia, frees one of the three Pussy Riot members at an appeal hearing.	One Pussy Riot Member Freed by Moscow Court — News — The Moscow Times http://t.co/m60lwaWU #FreePussyRiot

Table 5.7: Example topics detected on ED corpus - day one

RQ2: *Can we find a method to effectively group tweets to a number of clusters, with each cluster representing a thematic topic?*

The experimental results show our proposed approach outperforms other clustering-based methods and topic models, in both clustering performance and topic coherence. The obtained topics by the proposed model are also mostly coherent and well aligned with the real-world stories. Besides GSDMM+LFLDA, GSDMM+OLDA has also shown competitive performance in many categories. In general the two-stage hierarchical topic modelling framework has effectively improved performance over each individual model, proven to be a promising direction for a further research. For future work, we also plan to evaluate our system in tracking the same set of topics across adjacent time intervals, which is a different task to document clustering and topic detection.

We have already addressed target-specific sentiment classification and topical clustering for tweets in Chapter 4 and 5. In order to understand the sentiment towards different target entities from a micro-level and thus develop a more in-depth analysis, we study the task of multi-tweet summarisation in the following chapter.

CHAPTER 6

Twitter Opinion Summarisation

Towards neural abstractive summarisation of tweets

In the previous two chapters we have discussed our work on target-specific sentiment recognition and topical clustering of tweets. Continuing our work towards understanding public opinion on Twitter, in this chapter we study the task of summarising opinionated tweets on common topics, with the goal of adding explanation and justification behind the sentiments expressed towards different issues and entities. We formulate it as a multi-document summarisation problem for tweets.

In recent years social media such as Twitter have gained prominence as a rich resource for opinion mining or sentiment analysis on diverse topics. However, analysing sentiment about diverse topics and how it evolves over time in large volumes of tweets is a difficult task. In Section 6.1, we present an interactive visualisation system for analysing sentiment about specific topics or entities over time while providing fine-grained extractive summaries to give insights into the underlying reasons. We illustrate its use with examples of topics discussed on Twitter during the 2017 UK general election.

Most existing tweet summarisation approaches rely on extractive methods, which rank and select tweets according to various relevance criteria for a summary. This approach has the inherent limitations of unavoidably including incomplete or redundant information, its generated summaries also typically lack cohesion and coherence. On the contrary, abstractive summarisation aims to resemble the way humans write abstracts, and produce a summary which is not limited to the vocabulary of the original document. Such abstract summaries are less redundant and more informative. Due to its challenges, there has been few work using abstractive summarisation on tweets. In this chapter, we ask the following research question:

RQ3: *How can we generate abstractive summaries for tweets towards common topics expressed on Twitter? Is it possible to generate tweet abstracts from scratch with limited training resources?*

Neural sequence-to-sequence (or seq2seq) model, consisting of an encoder and a decoder, has shown promising results in various NLP tasks including abstractive summarisation on traditional news articles. To address **RQ3**, we study the feasibility of applying state-of-the-art neural abstractive summarisation for tweets. We investigate how to overcome the limitation of insufficient training resources, and evaluate the performance of cross-medium summarisation. To the best of our knowledge, there is no existing work on applying seq2seq model to multi-document tweet summarisation.

6.1 Topic-based, Temporal Sentiment Summarisation for Twitter

Our problem formulation is related to work on prospective information needs, represented by the Microblog [205], Temporal Summarisation [206] and Real-Time Sum-

marisation [207] tracks at recent Text Retrieval Conferences (TRECs). However, while the aim of these tasks is to keep users up-to-date with topics of interest via push notifications or email digests, our aim in this section is to provide an interactive user interface that shows how sentiment towards specific entities or topics develops over time. We have incorporated an automatic summarisation feature to assist users in understanding the underlying reasons. Thus, our motivation is related to [25], which also proposes a topic-oriented opinion summarisation framework. However, we use state-of-the-art methods enabling intuitive and interactive visualisation of sentiments in chronological order. This provides a useful tool for analysing an important event over time, such as elections, both quantitatively and qualitatively.

Here, we describe our system that aims at the aforementioned objectives. Its interactive web interface is accessible online[1]. We also present two use cases to demonstrate how the system can be used in analysing public sentiment.

6.1.1 System Design

An overview of the system is depicted in Figure 6.1 and comprises: 1) Data collection and sampling; 2) Sentiment classification; 3) Tweet summarisation; and 4) Data visualisation.

Data Collection and Sampling: We collected a corpus of tweets about the 2017 UK general election through Twitter's streaming API by tracking 15 hashtags[2]. Data harvesting was performed between 26 May and 21 June 2017 to capture discussions in the two weeks running up to and after the election. To identify relevant topics and entities in each tweet, we match tweets against two manually curated lists of keywords (both were created during the 2015 UK election cycle) which include 438 topic keywords relevant to nine popular election issues (e.g., immigration, NHS) and a list of 71 political party aliases (e.g. 'tories', 'lib dems'). The resulting

[1]Live demo: `http://elections.iti.gr/uk2017/`

[2]#ukelection2017, #ge2017, #ge17, #ukge2017, #ukgeneralelection2017, #bbcqt, #bbcdp, #marrshow, #generalelection2017, #generalelection, #electionuk, #ukelection, #electionuk2017 and #brexit

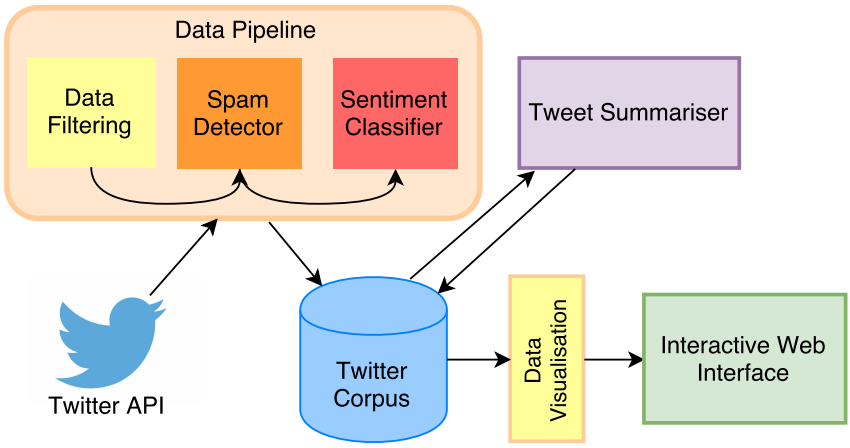


Figure 6.1: Overview of the proposed summarisation system

corpus contains 3,663,090 tweets, with each tweet mentioning at least one keyword. To increase data quality and reduce noise in the corpus, we trained and applied a Twitter spam detection model using features described in Chapter 3.

Sentiment Classification: We use the multi-target-specific approach described in Chapter 4 for identifying 'negative', 'positive' or 'neutral' sentiment of each topic entity. The whole data pipeline of Figure 6.1 is designed to dispatch work to many machines in parallel[3], processing many data batches simultaneously, which makes it scalable and efficient.

Tweet Summarisation: Here we aim to extract a list of representative tweets summarising the sentiment(s) expressed towards each topic/entity on each day (e.g. tweets containing positive sentiment towards 'NHS' posted on 26 June 2017).

As a prerequisite for summarisation, we group tweets containing the same sentiment towards a topic/entity on a day into a number of clusters, with each cluster assumed to represent a common theme or reason underlying the particular choice of sentiment. We use the two-stage hierarchical topic modelling approach described in Chapter 5 and select the GSDMM+OLDA model for this task due to

[3]We ran it on a server with 40 CPU cores and 64 GB of RAM.

its efficiency. If there are fewer than 10 unique tweets containing the same sentiment towards a topic (or entity) on a particular day, we skip clustering and treat each of these tweets as a cluster.

To extract representative tweets summarising each cluster, we place every tweet in one common embedding space and identify 20 tweets closest (by cosine distance) to the cluster centroid (also known as metroid tweets) as summary candidates. The embedding space here is constructed using a simple but effective sentence embedding method proposed by Arora et al. [126], which reported good performance on 22 textual similarity data sets, including a Twitter corpus. We then rank the 20 summary candidates based on weighted average tf-idf scores in the cluster; these scores can be regarded as a measure of informativeness.

We select the most informative tweet from the 20 candidates as the summary for that cluster and the final summary for the sentiment expressed towards the topic entity is the summaries combined from all its clusters (e.g., tweets containing positive sentiment towards 'NHS' posted on 26 June 2017, comprise 8 clusters with a summary consisting of the summary tweet from each one).

6.1.2 Data Visualisation

For each topic/entity we calculate the following daily features: *# of tweets, # of unique users, # of tweets per sentiment type (pos, neg, neutral)* and *# of unique users per sentiment*. These features were selected based on past studies on the domain of predicting election results with social media [269], as well as on the basis of providing potentially useful insights on the election monitoring process. These are accompanied by the daily summaries of each sentiment type for a given topic/entity as described above.

In addition to showing the raw values of the above features, we also normalised sentiment features (*# of tweets per sentiment, # of unique users per sentiment*) to reflect the percentage of sentiment of a particular type towards a topic/entity

on a particular day. To allow time series comparisons across different topics/entities we normalised the *# of tweets* and *# of unique users* of all topics/entities across all days in the range [0, 1]. Finally, to account for differences in popularity, we calculated the average (per-topic and across all days) *# of tweets* and *# of unique users*.

The web interface is implemented on Web standards (HTML5/CSS3). The timeline graphs are built using the NVD3[4] library (reusable charts for `d3.js`), while the auto-complete functionality is based on the 'Ajax AutoComplete for jQuery' library[5]. In addition, jQuery from Google Hosted Libraries[6] and D3.js from Cloudfare Hosted Libraries[7] are used also for DOM manipulation (click events, add/remove elements etc.) and accessing data (from tsv files) respectively.

6.1.3 Use Case #1 – Party Sentiment

In section 6.1.3 and section 6.1.4, we use two use cases to demonstrate how our system can help to analyse public sentiment on Twitter.

Recent election campaigns suggest that the Twittersphere tends to contain more negative sentiment during the election period. Hence, in the first case study, we compare negative sentiment trends on Twitter for the two major UK political parties, 'Conservative' and 'Labour', before and after the 2017 UK general election. As described in section 6.1.2, the negative sentiment reflects the percentage of negative sentiment for each party on each day over all sentiment bearing tweets.

Figure 6.2 reveals consistently more negative sentiment towards 'Conservative' than 'Labour', especially for the week before election day (8 June). Interestingly, we also observe that, whereas negative sentiment towards both parties dipped one day after the election, negative sentiment towards 'Labour' rose between June 9 and 11 to be on par with 'Conservative' and then dropped sharply to reach its

[4] `http://nvd3.org/`
[5] `https://www.devbridge.com/sourcery/components/jquery-autocomplete/`
[6] `https://developers.google.com/speed/libraries/`
[7] `https://cdnjs.com/`

lowest point on June 17. During this same post-election period, negative sentiment towards 'Conservative' was on a steady and gradual rise.

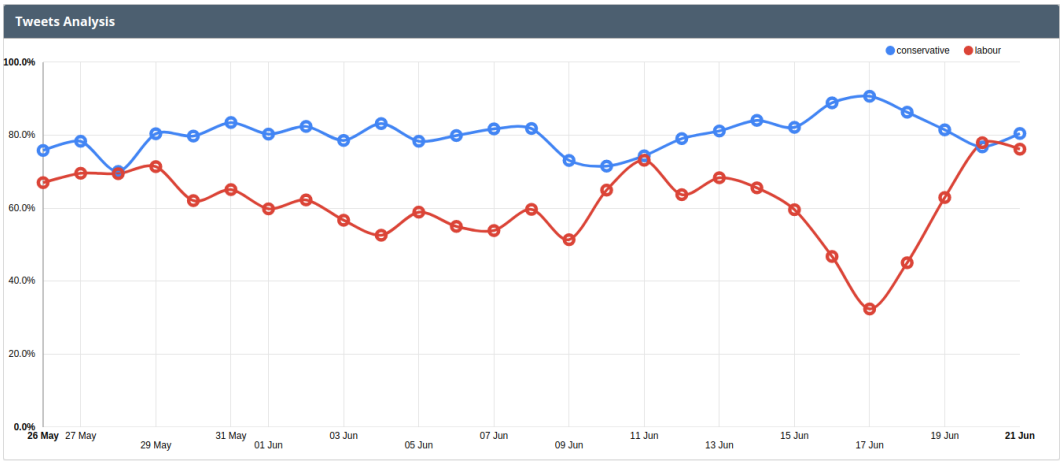


Figure 6.2: Negative sentiment trends for 'Labour' (red) and 'Conservative' (blue).

6.1.4 Use Case #2 – Grenfell Tower Fire

To provide deeper insight into the advantages of our opinion summarisation system, we present a case study on how public sentiment towards the topic 'housing' developed before and after the Grenfell Tower Fire disaster[8]. Figure 6.3 shows the percentage of users expressing negative sentiment towards 'housing' as well as the governing party 'conservative' over the period covering the incident. Our web interface allows users to click on each circle shown on the graph to display tweet summaries for that topic on that particular day.

We can see the number of users expressing negative sentiment for the topic 'housing' fluctuated throughout the election period while it remained fairly constant for 'Conservative'. Negative sentiment peaked in both cases on June 16th. We also observe a huge dip for users expressing negative sentiment towards 'housing' between June 17 and 20 and an increase in neutral sentiment at the same time.

Table 6.1 presents a negative sentiment summary for each day between June

[8] `https://en.wikipedia.org/wiki/Grenfell_Tower_fire`

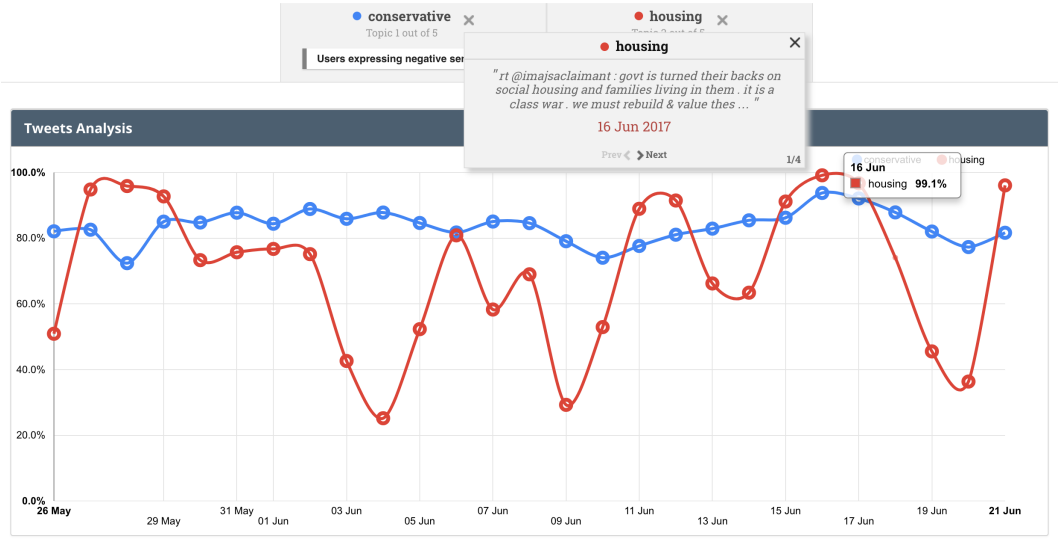


Figure 6.3: Negative sentiment trends for 'housing' (red) and 'conservative' (blue), with a summary tweet displayed for the former.

12 and 15, and all three negative opinion summary tweets on the peak day of June 16 showing each summary tweet represents a different aspect of the topic. Along with the graph shown in Figure 6.3, this summary is a tight integration of topic, sentiment and insight into reasons behind the sentiment. Before the fire, negative sentiment towards 'housing' was austerity related; after the fire, the incident dominated the 'housing' discussion on Twitter. A large portion of users blame the Conservative government for the decline of social housing and ultimately the Grenfell Tower fire. Finally, on June 16 each of the negative opinion summaries represents one theme related to this topic, namely 'the decline of social housing', 'immigration and housing' and 'the votes on housing safety'.

6.1.5 Conclusion

Here we present a system for monitoring topic-entity sentiment on Twitter and summarising public opinion around the sentiment towards each entity. The system deployment for the 2017 UK election, provides an interactive visualisation for com-

Topic entity	Opinion Summaries	Date
housing	rt @user1 : the audacity to even refer to tackling a " housing crisis " after being in government for 7 years . https://t.co/lifwybhryp	12 June 2017
housing	austerity is still here , bedroom tax , foodbanks , pip , housing cap , universal credit taper , welfare freeze , esa cuts , inflation is up . #ge17	13 June 2017
housing	@bbcnews @skynews @itvnews tories cuts in society kill just look at social housing #grenfelltower sold to cheapest bidding #ge17 #bbcqt	14 June 2017
housing	tory capitalism cutting kills social housing on the cheap #grenfelltower cuts in fire ambulance police nhs services #victorialive #ge17	15 June 2017
housing	rt @user2 : govt is turned their backs on social housing and families living in them . it is a class war . we must rebuild & value thes ...	16 June 2017
housing	rt @user3 : laura perrins again blaming the death toll of #grenfelltower on immigration - putting pressure on housing . laura bt ...	16 June 2017
housing	rt @user4 : it is a shame the ministers hearts did not go out to the people in grenfell tower when they were voting on housing safety #bbcqt	16 June 2017

Table 6.1: Negative opinion summary for 'housing' before and after the Grenfell Tower fire

paring sentiment trends and display opinion summaries on the graph. In the future, we plan to improve our system to produce more concise summaries and allow near real-time processing of new events.

6.2 Neural Abstractive Multi-tweet Opinion Summarisation

Recently sequence-to-sequence (seq2seq) models [35], in which recurrent neural networks (RNNs) read text via an encoder and freely generate text via a decoder, has made abstractive summary generation from scratch viable [182, 183, 187, 184, 185, 186]. Although recent literature shows neural abstractive summarisation is a very promising direction forward, we have not seen any work on applying seq2seq models to multi-document abstractive summarisation on tweets. This is possibly due to the lack of labelled training resources which is the key requirement for seq2seq models. Here we study the feasibility of applying seq2seq model with attention mechanism for such task and how to overcome its limitations.

6.2.1 Problem Formulation

Different to other neural abstractive summarisation studies, our input consists of a number of tweets mentioning the same topic, denoted as $\{\mathbf{x} = x^1, ..., x^N\}$. Each input unit (i.e. a tweet) x^k is composed by a sequence of words $x_1^k, ..., x_L^k$, where L is the number of words in this input unit. Each word takes the form of a fixed-sized vector representation, which can be initialised randomly or by pre-trained embedding vectors, and updated during training.

Our summarisation task here is defined as finding $\mathbf{y}$, which is the most likely sequence of words $y_1, ..., y_M$ that preserve the meaning of $\mathbf{x}$:

$$\mathbf{y} = \text{argmax}_y P(y|\mathbf{x}) \tag{6.1}$$

Where $P(y|\mathbf{x})$ denotes the conditional probability of the output (i.e. summary) sequence y, given the input sequence x. $P(y|\mathbf{x})$ can be modelled by a parametric function with parameters θ, as $P(y|\mathbf{x};\theta)$. The training in this task aims to find the θ that maximises the conditional probability of document-summary pairs in the training corpus.

6.2.2 Sequence-to-Sequence Attentional Model

As described in Section 2.4.2, a seq2seq model consists of an encoder and a decoder, where the encoder is fed by the tokens of the input sequence one by one to produce a fixed length hidden state representation, and the decoder generates its own hidden state from the representation of the previous token, the previous decoder state, and the embedding representation of the current input token. We use a single-layer unidirectional LSTM [74] as the decoder. For the encoder, we use a single-layer bidirectional LSTM, adding an attention layer [181] to produce a weighted sum of the encoder hidden states. This is fed to the decoder so the decoder knows where to look in the input sequence to generate the next summary token.

6.2.3 Extractive-Abstractive Summarisation Framework

A key difference between our task and majority of the existing abstractive summarisation studies, is that our input consists of multiple separate input units (i.e. tweets of a topic). A simple solution would be to concatenate them into one document. However this would make our input sequence so long that the training will become extremely inefficient and time-consuming especially with attention mechanism added. By manually evaluating our datasets, we find even though each cluster of tweets mentions the same topic, many of them contain redundant or secondary information. Therefore we think the summarisation task can be divided into two steps. The first step serves the purpose of information compression, which promotes topical and diverse information. It samples smaller set of tweets from the original

cluster and feeds to our seq2seq model as high-quality input. Then our second step performs abstractive summarisation that learns to pick important information and add to the final summary.

While Wang and Ling [187] opt to train a regression model for estimating the importance of each input unit using manually engineered features, we propose a two-stage framework similar to [199] which consists of an extractive summarisation step for selecting important tweets as input for the subsequent abstractive summarisation using seq2seq. Since our extractive summarisation is completely unsupervised, such sub-sampling step does not need the ground-truth summary for creating gold-standard importance score as is required in [187].

6.2.4 Pointer-Generator Network for Abstractive Summarisation

As mentioned in Chapter 2, abstractive summarisation models using seq2seq have the tendency to generate repeating summaries and suffer from out-of-vocabulary words (OOVs). To alleviate these issues, we choose to adopt a pointer-generator network [185] for our tweets summarisation. The pointer-generator model learns to generate a summary sequence of tokens y_i based on the following conditional probability:

$$p(y_i = w|y_1, ..., y_{i-1}, x) = p_{gen}P_{vocab}(w) + (1 - p_{gen})\Sigma_{i:w_i=w}a_i^t \tag{6.2}$$

Where P_{vocab} denotes the probability to generate a new word from the vocabulary, p_{gen} is learnt parameter used as a soft switch for choosing between generating a word or copying a word from the input sequence depending on the attention distribution a^t and hidden states of the decoder. This gives the network the ability to produce OOV words that is not restricted to the pre-set vocabulary. We also adopt the coverage mechanism proposed in [185], which sums the attention distributions over all previous decoding time steps to obtain a coverage vector. Such coverage vector is

added to the attention mechanism to keep track of its previous decisions (on choosing where to attend). This alleviates the issue of generating repetitive summary text.

The overview of our extractive-abstractive summarisation system (including the pointer-generator network) is presented in Figure 6.4, where $C_1, ..., C_n$ represents the tweet clusters; $w_1, ..., w_n$ is the words of the input document in the training data and $x_1, ..., x_n$ is the words of the corresponding summary. $h_1, ..., h_n$ and $s_1, ..., s_n$ are the hidden states of the encoder and the decoder respectively. Here we use LexRank as an example to demonstrate our information compression/data sub-sampling step using an extractive summarisation method.

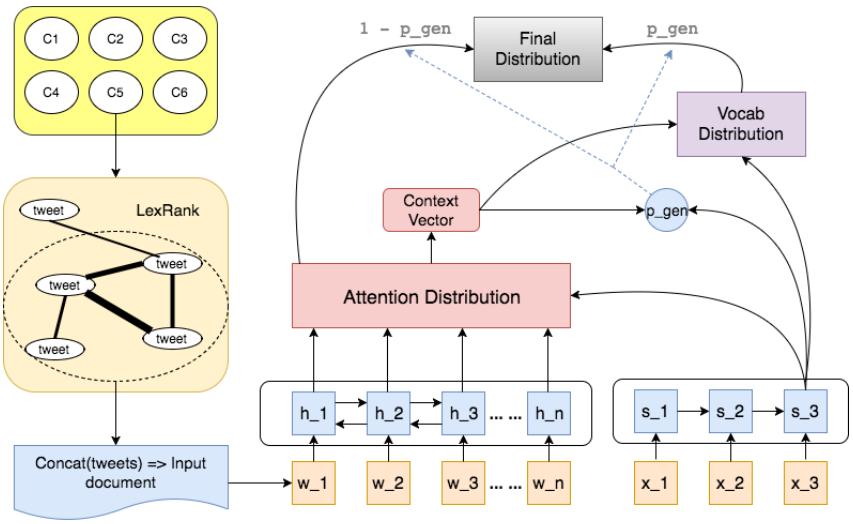


Figure 6.4: Overview of extractive-abstractive summarisation system

6.2.5 Unsupervised Pretraining for Model Initialisation

As described in Chapter 2, one key requirement for seq2seq models or deep neural networks in general, is they require a large set of labelled training data to avoid overfitting. Several studies [189, 191, 192] have experimented using weights from pretrained language model for initialising seq2seq networks to improve the performance and convergence under a low-training-resource constraint. Due to the lack of good-quality Twitter corpus with human-generated reference summaries, we set

to use a traditional news article-abstract corpus as our training data for this tweet summarisation task. As a result, our training and testing data are from two different domains and mediums.

To combat these data issues, we opt to pre-train language models using large-scale unlabelled Twitter data for initialising our pointer-generator summarisation model. More specifically since the encoder of our summarisation model uses a single-layer bidirectional LSTM, we first pre-train its forward-LSTM on a large set of tweets to predict the next word given the previous ones, and then pre-train the backward-LSTM using the same parameter settings on the same set of tweets but the words are in a reverse order. For initialising our decoder, we use the same pre-trained weights used for the forward-LSTM in the encoder. Lastly, the embedding layers are initialised with existing word embeddings. More details are described in Section 6.3.3.

6.3 Experiments and Results

We conduct two experiments for evaluating our proposed neural abstractive summarisation system, namely event summarisation and opinion summarisation, and in each experiment we compare with other extractive and abstractive baseline models. This allows us to evaluate these systems on two different tasks under the constraint for human-generated reference summaries.

6.3.1 Datasets

For **training** our neural summarisation model, due to the lack of good-quality Twitter summarisation corpus, we use the *CNN/Daily Mail* dataset [270] which has been used in several recent news article summarisation work [184, 185]. It has an averaging 781 tokens per article, and 3.75 sentences or 56 tokens per summary. We have obtained 287,226 training pairs, 13,368 validation pairs and 11,490 test pairs,

Event class	Event summary	Sample tweet
Science & Technology	Alpha Centauri Bb, an exoplanet, is discovered orbiting around Alpha Centauri.	earth sized exoplanet found in nearest star system to earth - alpha centauri b #awesome #whencanigo
Law, Politics & Scandals	Chief Whip of the British Conservative Party Andrew Mitchell resigns over remarks he made to police officers in Downing Street, and following a lengthy political row over the issue.	chief whip andrew mitchell has resigned in wake of row over outburst at police in downing street. about time
Sports	Finnish racing driver Kimi Raikkonen wins Formula One's 2012 Abu Dhabi Grand Prix.	yea :) "@USER : kimi raikkonen has won the abu dhabi grand prix with fernando alonso 2nd & sebastian vettel 3rd #ssf1"

Table 6.2: Example event summaries with corresponding sample tweets

using the script provided by See et al. [185]. In contrast to [184], we use the original (i.e. non-anonymised) version of the corpus without replacing named entities with unique tokens as placeholder, as we believe it is a more realistic summarisation setting.

For **event summarisation**, we use the large scale event detection corpus introduced in [140] containing 150,000 tweets over 28 days covering more than 500 events, as our test data. In addition to tweets, this corpus also has the event descriptions written by human workers from the original crowdsourced corpus evaluation. Each description captures the essence of its corresponding event, and contains important named entities (e.g. people or places). We use these descriptions as the reference summary for each event. After retrieving 78,138 tweets using the Twitter API, we obtain 161 events in which each event contains no less than 100 tweets, each tweet contains more than 5 tokens and each event summary has no less than 10 tokens, to ensure the quality of our summarisation corpus. Our final event summarisation corpus has averaging 328 tweets per event (max. 7713 tweets and min. 100 tweets for an event), and averaging 19 tokens per reference summary (max. 60 tokens and min. 10 tokens). Some example event summaries with the corresponding randomly sampled tweets are shown in Table 6.2.

For **opinion summarisation**, we use the 2017 UK general election corpus

from Section 6.1 after performing target-specific sentiment classification and topical clustering, as our test data. Our goal is to summarise the sentiment(s) expressed towards each topic/entity on a particular day during the election period. Contrary to our approach in Section 6.1 where we extract representative tweets as summary, in this section we evaluate abstractive summary for each cluster. The final summary for a target-sentiment is the combination of all its cluster summaries. We use a 5-day sample set consisting of every tweet from the election corpus that is posted between 01/06/2017 and 05/06/2017. After performing clustering and removing clusters that have less than 100 unique tweets, we obtain 231 clusters for evaluation. Note that we do not have human-generated reference summary for this task, therefore we opt to use input-summary similarity based metrics for evaluation which we describe in the following section.

The same **preprocessing** steps are applied for both datasets to reduce the noise level of the tweets. This includes removing hashtag symbols, URL links, user mentions and punctuations as well as lower-casing and the tokenisation of each tweet. We also remove tweets that have no more than 5 tokens, and clusters/events that contain less than 100 tweets.

6.3.2 Automatic Summary Evaluation Metrics

Evaluating summary quality has remained as a challenging problem due to not only the subjectivity of the task but also it is still unclear how to quantify many aspects of the summary quality such as clarity, informativeness or coherence [174]. The ultimate goal of summarisation is to improve users' reading experience and to acquire important information from the source document faster. Therefore, some studies [271, 272] carried out task-specific evaluations to measure if the summarisation improves the users' performance on a downstream task such as information retrieval or question answering. This is known as extrinsic evaluation for summarisation. However, extrinsic evaluation is time-consuming, and its existing experimental designs

are still far from being well developed (e.g. it needs to address other factors such as user interface that also affects the users' task performance). We also decide to leave human evaluation, which usually involves ranking and comparing between the proposed system and other baseline models, for further work.

In our work, we adopt several popular intrinsic evaluation metrics which measure the summary quality based on the coverage between system generated summary and human reference summary or the original document. ROUGE [273] is the most widely used metric for automatic summary evaluation based on content coverage. We report the F_1 scores for ROUGE-1, ROUGE-2 and ROUGE-L, which respectively measure unigram-overlap, bigram-overlap and longest common subsequence (LCS) between the reference summary and our model-generated summary. Our ROUGE scores are computed using the `pyrouge` package[9]. We also evaluate with a recall-oriented metric, METEOR [274][10]. We report both in exact match mode and full mode (which also matches stems, synonyms and paraphrases)[11].

Additionally, we report several input-summary similarity based scores including Jensen-Shannon divergence (JSD) and percentage of input topic words which are found to have high correlation with both responsiveness and pyramid evaluation [275] (which relies on human summaries as the gold-standard) at the macro-level even though they do not require the use of any reference summary [276]. We report both divergence and topic signature-based feature scores[12]. The divergence features consist of Kullback Leibler divergence (KLD), smoothed and unsmoothed Jensen-Shannon divergence (JSD). The three topic signature based features are APTT (i.e. averaged percentage of tokens in the summary that are topic words of the input), AFTW (i.e. averaged fraction of topic words of the input that appear in the summary), and ATWO (i.e. averaged cosine similarity using all words of the summary

[9] `https://pypi.python.org/pypi/pyrouge/0.1.3`

[10] Note that recall-based metrics tend to have bias towards longer summaries.

[11] `http://www.cs.cmu.edu/~alavie/METEOR/`

[12] Lower divergence scores indicate better summary quality, while for other metrics the higher the scores are the better.

but only the topic words from the input)[13].

As suggested in [276], while such word distribution similarity-based metrics can provide reliable estimates of system summary quality when averaged over all test inputs (i.e. system-level evaluation as opposed to individual-input level), they work well only for cohesive-type inputs. Given that our opinion summarisation task in Section 6.3.5 aims to summarise each cluster of tweets that hold the same sentiment towards common entity on the same day, we think such input-summary similarity based metrics are appropriate for evaluating the opinion summarisation task. For the event summarisation evaluation, we use these metrics to complement ROUGE and METEOR.

6.3.3 Experimental Setup

For our neural abstractive summarisation models, we use 100 dimensional word embeddings and 256-dimensional hiddens states in both encoder and decoder. For training, we adopt the same settings as used in [185]. This includes a 50k-word vocabulary for both source and target sentences, Adagrad [277] with a learning rate of 0.15 and an initial accumulator value of 0.1 for optimisation, and a maximum gradient norm of 2 for gradient clipping. As described in Section 6.2.3, we use extractive summarisation for sub-sampling our data to produce high-quality input for the subsequent abstractive summarisation. We choose LexRank as the extractive method for for sampling 30 tweets as an input for abstraction. LexRank has shown to achieve very good performance for multi-document summarisation, and it promotes diversity by adding Cross-Sentence Informational Subsumption (CSIS) as its heuristic final step. We denote our summarisation system as *Our System*.

Training: We truncate the source sequences to 400 tokens and limit the length of summary to 100 tokens for training and 30 for testing[14]. For training we start

[13]The topic signature based features require the supply of occurrence counts for words, which we produce by using 675 million of UK tweets as the background corpus.

[14]We then use the first sentence as our final summary.

with highly-truncated sequences and then gradually increase maximum timesteps, as suggested in [185]. For each model we train on a Tesla K80 GPU with a batch size of 16. During testing, we use beam search with beam size of 4. We also add the coverage mechanism at the end of our training process for a further 3,000 iterations.

Pre-training: Using Twitter's full firehose from Gnip, we collected a large number of high-quality geo-tagged tweets posted in the UK posted between May and October 2015[15]. After performing basic preprocessing steps including removing URL links, retweet symbols, user mentions, tokenisation as well as removing tweets that contain less than 5 tokens, we have a corpus of 675 million tweets which we use for pre-training.

We train two language models (LMs) using the aforementioned Twitter corpus and the same corpus but with words in reverse order for each tweet, respectively. Each LM has the embedding size of 100, and a one-layer LSTM with state size of 256. Both models are trained on 4 Tesla K80 GPUs with a batch size of 128, for $\sim$ 557k iterations, in a similar fashion to [278]. The LSTM layers of our encoder and decoder are initialised with the corresponding trained weights of these LMs. We use the `GloVe` word vectors [78] trained from 2 billion tweets for initialising the embedding layers in our summarisation models, same as in [184].

Baselines: To fully evaluate our proposed models, we compare with both extractive and abstractive summarisation methods, namely:

- **Centroid-based**: The same centroid-based extractive summarisation method used in Section 6.1.
- **TextRank** [279]: A graph-based ranking algorithm for extracting sentences as summary.
- **LexRank** [160]: Similar to **TextRank**, as both models compute text centrality based on PageRank algorithm. However the techniques used in computing

[15]Note that unlike the Twitter REST API, its full firehose provides 100% of the tweets that match the user defined criteria.

similarity, weight graph edges, post-processing etc. are different.

- **SumBasic** [211]: A term-frequency based summarisation system. It uses a redundancy factor for minimising redundancy in the summary.
- **Hybrid-Tfidf** [24]: A Tf-idf based model adapted for multi-document summarisation. It employs a similarity threshold, for reducing redundancy.
- **ILP-based** [280]: Extending an Integer Linear Programming (ILP) based concept summarisation model [164] to obtain one single optimal solution. This is also an extractive baseline.
- **Opinosis** [177]: A graph-based abstractive summarisation algorithm, aimed to reduce repetitive information and merge opinionated expressions based on syntactic structure of product reviews.

We also compare between our neural abstractive models with and without the pre-training. For both event and opinion summarisation tasks, our goal is to generate and evaluate one-sentence summary for each event or cluster[16]. We leave the evaluation of multi-sentence summary generation for the future work.

6.3.4 Results for Event Summarisation

As shown in Table 6.3, it is clear that extractive systems tend to achieve higher ROUGE and METEOR scores than the abstractive ones except for our system #2 (i.e. LexRank+seq2seq), which is in line with the findings in [185] for news article summarisation. We think the nature of our event summarisation corpus (i.e. each event has only one reference summary in our corpus) and the inflexibility of ROUGE make extractive approaches difficult to beat. As explained in [185], ROUGE rewards safe strategies such as preserving original phrasing, and as a result safer strategies

[16]In the opinion summarisation task, we evaluate on the cluster-level rather than entity-sentiment-level, as we believe this is more appropriate for measuring the quality of the generated summaries.

like extractive approaches score higher. In comparison, abstraction introduces more choices of phrasing, which leads to less chance of matching the reference summary.

Among the extractive approaches, the two graph-based methods, namely TextRank and LexRank, achieving the highest scores across the board, while frequency-based SumBasic has lower scores on METEOR for extracting shorter summaries. We also observe our neural abstractive models receive relatively higher performance for METEOR, but still perform worse than most of the extractive baselines. This is possibly due to the language used on Twitter that is not compatible with the pre-defined list of synonyms and paraphrases used for computing the METEOR metric.

Our summarisation systems both with and without pre-training, outperform the abstractive summarisation baseline, Opinosis. We have also tested using other extractive methods for sub-sampling inputs for abstractive summarisation. In general we find the performance improves in comparison to just using extractive summarisation alone. Comparing among our abstractive summarisation models, we find by initialising the network using weights from pretrained models, the performance drops especially when we only pre-train encoder but not decoder and vice versa.

Table 6.4 shows the results for divergence and topic signature-based similarity scores, for measuring the content similarity between summary and original inputs. We find again most of the extractive baselines perform better than the abstractive models including our 2-stage extractive-abstractive systems, while three of our systems (i.e. "Our System", "+pre-embed", and "+pre-emb-enc-dec") outperform the abstractive summarisation baseline Opinosis.

Additionally, we evaluate our models on a Twitter corpus for summarising important information that is relevant to each topic of an earthquake happened in Italy in 2016 [200]. This dataset contains two levels and four topics for each level, thus overall 8 topics, making it a small corpus. It does contain a summary of the tweets for each topic, extracted by human annotators. The summaries were

Model	Length	ROUGE			METEOR	
		1	2	L	exact match	+ stem/syn/para
Centroid	13.4	25.62	7.77	21.27	9.34	10.51
TextRank	18.6	32.74	13.32	26.60	**13.64**	**14.97**
LexRank	13.5	**33.69**	**12.66**	**27.24**	13.09	14.57
SumBasic	8.7	30.83	10.88	25.55	10.95	11.84
Hybrid-Tfidf	14.7	29.62	10.70	24.61	12.15	13.23
ILP-based	18.8	27.32	8.76	23.00	11.33	12.81
Opinosis	10.6	23.81	9.27	20.99	9.74	10.45
Our System	15.43	31.86	12.65	26.73	13.08	14.67
+ pre-emb	13.71	29.79	10.87	25.01	12.12	13.33
+ pre-emb-enc	11.61	19.17	5.40	15.96	7.70	8.28
+ pre-emb-dec	16.06	10.94	3.90	8.76	4.53	4.92
+ pre-emb-enc-dec	14.06	25.08	8.98	21.18	10.05	11.19

Table 6.3: ROUGE F_1 and METEOR scores on the event test set. This table is divided into 3 sections: extractive baselines, abstractive baseline, and our systems.

Model	JSD		KLD		APTT	AFTW	ATWO
	Un—	Smoothed					
Centroid	0.389	0.237	1.634	1.235	0.736	0.0744	0.603
TextRank	0.323	0.208	1.424	1.010	0.790	0.0947	0.724
LexRank	0.314	**0.169**	**1.238**	**0.614**	0.888	0.0845	0.738
SumBasic	0.332	0.170	1.268	0.618	**0.897**	0.0645	**0.754**
Hybrid-Tfidf	0.320	0.188	1.333	0.818	0.82	0.0846	0.735
ILP-based	0.356	0.246	1.597	1.392	0.672	0.1033	0.648
Opinosis	0.402	0.227	1.700	0.908	0.808	0.0618	0.573
Our System	**0.289**	0.213	1.480	0.973	0.779	**0.128**	0.747
+ pre-emb	0.290	0.218	1.524	1.119	0.766	0.126	0.729
+ pre-emb-enc	0.479	0.399	2.886	5.804	0.466	0.057	0.286
+ pre-emb-dec	0.508	0.462	3.772	7.477	0.323	0.073	0.296
+ pre-emb-enc-dec	0.312	0.233	1.641	1.184	0.769	0.118	0.682

Table 6.4: Content similarity scores on the event test set.

prepared by the same human annotators who judged the relevance of the tweets, and are of 300 words at most, which makes their gold-standard summaries much longer than the ones in the event detection corpus (averaging 19 tokens per reference summary). The results for this dataset presented in Table 6.5, are consistent with our findings for our event test data.

Model	JSD		KLD		APTT	AFTW	ATWO
	Un—	Smoothed					
Centroid	0.535	0.411	2.290	3.165	0.282	0.174	0.189
TextRank	**0.488**	0.395	2.293	3.062	0.415	**0.351**	**0.435**
LexRank	0.507	**0.354**	**1.940**	**2.266**	**0.501**	0.251	0.369
SumBasic	0.497	0.364	2.006	2.426	0.444	0.238	0.387
Hybrid-Tfidf	0.526	0.423	2.426	3.396	0.357	0.240	0.314
ILP-based	0.537	0.454	2.688	4.008	0.204	0.227	0.217
Opinosis	0.554	0.369	1.947	2.668	0.413	0.160	0.283
Our System	0.507	0.400	2.320	2.848	0.370	0.272	0.321
+ pre-emb	0.525	0.427	2.504	3.421	0.271	0.227	0.209
+ pre-emb-enc	0.601	0.464	2.524	4.086	0.303	0.106	0.093
+ pre-emb-dec	0.602	0.500	3.084	6.457	0.150	0.155	0.111
+ pre-emb-enc-dec	0.549	0.441	2.520	3.965	0.248	0.192	0.194

Table 6.5: Content similarity scores on the SMERP corpus.

6.3.5 Results for Opinion Summarisation

Our results for the election dataset are given in Table 6.6 using the content similarity-based metrics described in Section 6.3.2. Our extractive baselines again show strong performances, outperforming both the abstractive baseline model and our summarisation systems. Among the abstractive methods, we observe comparable results obtained by Opinosis and our systems. Lastly, we observe little performance improvement by using pre-trained word vectors for initialising the embedding layers (i.e. "+pre-emb") or pretraining encoder, decoder and the embedding layer (i.e. "+pre-emb-enc-dec").

We also find the summaries generated by our systems are highly extractive. This is due to the integration of the pointer component, which tends to encourage the copying behaviour during summarisation. Table 6.7 displays a typical example of our generated summary. The model is able to combine key information from multiple tweet sources, and generate a concise and cohesive summary that is grammatically correct. Unnecessary phrases and unimportant expressions are omitted from the summary. However, we do not see any novel words in the summary, in fact, all the words are from the original tweets, indicating a lower degree of abstraction. This

Model	JSD		KLD		APTT	AFTW	ATWO
	Un—	Smoothed					
Centroid	0.502	0.259	1.564	1.506	0.513	0.035	0.487
TextRank	**0.443**	0.214	1.339	1.077	0.668	0.039	**0.686**
LexRank	0.448	0.200	1.278	0.886	0.680	0.038	0.617
SumBasic	0.461	**0.179**	**1.224**	**0.803**	**0.730**	0.029	0.685
Hybrid-Tfidf	0.458	0.221	1.348	1.156	0.613	0.040	0.601
ILP-based	0.481	0.276	1.633	1.611	0.524	**0.043**	0.575
Opinosis	0.498	0.208	1.453	0.870	0.612	0.025	0.482
Our System	0.487	0.238	1.535	1.172	0.580	0.031	0.492
+ pre-emb	0.494	0.234	1.524	1.358	0.552	0.029	0.467
+ pre-emb-enc	0.566	0.284	1.880	3.375	0.401	0.013	0.254
+ pre-emb-dec	0.650	0.416	2.737	7.395	0.105	0.009	0.071
+ pre-emb-enc-dec	0.499	0.228	1.537	1.075	0.605	0.027	0.488

Table 6.6: Content similarity scores on the election opinion test set.

Source	Entity: *Tories*; Sentiment: *Negative*; Date: *02/05/2017*; Cluster: *#3*
Sample tweets	"the tories are coming for your pension , your winter fuel allowance and your house . do not let them . #votesnp #ge17" ... ... "the tories want to cut your pension . do not let them away with it . make sure and #votesnp on june 8th . #ge2017"
Summary	tories want to cut your pension , your winter fuel allowance .

Table 6.7: Example summary with corresponding sample tweets

shows the full abstraction of multiple opinionated tweets is still a challenge yet to be solved by our work.

6.4 Conclusions and Further Work

In the first part of this chapter, we have presented a system for time-sensitive, topic-based summarisation of sentiment around target entities and topics in collections of tweets. By enabling intuitive and interactive visualisation of sentiments in chronological order (its home page is shown in Figure 6.5), it can be used for analysing an important event over time, such as elections.

RQ3: *How can we generate abstractive summaries for tweets towards com-*

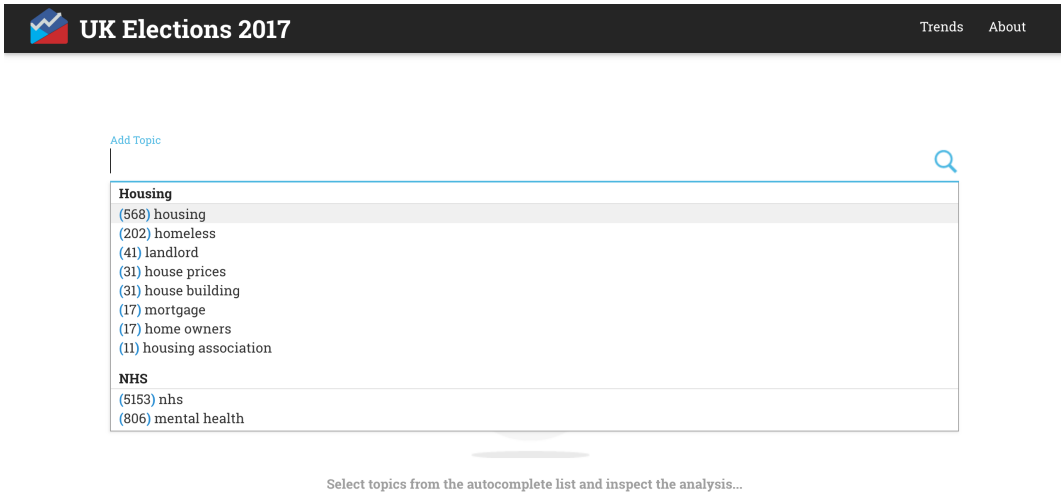


Figure 6.5: Home page for our interactive visualisation interface

mon topics expressed on Twitter? Is it possible to generate tweet abstracts from scratch with limited training resources?

In the second part of this chapter, we aim to tackle the challenge of abstractive summarisation for tweets on common topics, raised in **RQ3**. We have identified two main problems: the abstraction of multiple text units (i.e. tweets), and insufficient training resource. For the first problem, we adopt the pointer-generator network introduced in [185], and propose an extractive-abstractive framework including a state-of-the-art seq2seq model, for information compression and abstract generation. For the latter issue, we experiment with various pre-training techniques though we have observed little performance improvement, which shows the difficulty of cross-medium summarisation. As illustrated in [195], out-of-domain training can detect summary-worthy content but is not able to match the generation style in the target domain. In addition, the different nature of Twitter language makes the task even more challenging.

Albeit the difficulty of beating the extractive baselines when measured in ROUGE and the content-similarity metrics, we have demonstrated it is possible to

generate less extractive summaries by using the seq2seq model, even without any target-domain training data. As the first study on neural abstractive summarisation of tweets, we show it is a promising direction for future work.

As to future work, we plan to study different approaches to alleviate the lack of in-domain training resources, via sequence autoencoding for pre-training [189], transfer learning [193] or multi-task learning [170]. We also would like to explore character or sub-word level abstractive summarisation. The lack of human-generated reference summaries also limits our ability to fully evaluate our models.

While ROUGE scores have a good correlation with human judgment in general, the summaries with the highest ROUGE are not necessarily the most readable or natural ones. In addition, ROUGE favors lexical similarities between generated summaries and reference summaries, which makes it biases towards extracted summaries over abstractive summaries. Although the input-summary similarity metrics [276] correlate with human judgements for generic summaries, they may not work as well for opinionated summaries. In fact, we still do not know how well humans would perform using pyramid method [275] on opinions. All of these show the need for carefully constructed human evaluation to properly judge abstract generation. In the future, we plan to recruit human judges for the qualitative evaluation, which will consist of 3 rating tasks on the basis of 'informativeness', 'coherence' and 'grammaticality' (each with a 1-5 scale), and another task for ranking on all summary variations according to their overall quality. At last, we will also evaluate multi-sentence summary generated by our system and compare with the strong extractive baselines.

CHAPTER 7

Conclusions

In the recent years, we have witnessed an explosive growth of user-generated content from social media sites such as Twitter. It has provided a platform for the general public and high-profile governmental figures such as the incumbent president of the US, to express opinions towards a broad range of topics. While social media is thus potentially a rich resource for policy makers, government sectors and social organisations to shed light on public opinion, understanding the sentiment towards different issues and entities as manifested in the large volume of tweets (i.e. information overload), has remained a difficult task.

Motivated by this challenge, in this thesis we have devoted four chapters to address the research problems for understanding public opinion in social media from both macro and micro perspectives. We have pursued our work from three different yet interconnected angles: *sentiment*, *topics* and *summary*. Specifically, in Chapter 3 we have addressed the problem of spam detection to improve data quality, as well as cross-domain emotion classification; in Chapter 4 we have studied the challenge of target-specific sentiment recognition on Twitter; in Chapter 5 we have studied topical clustering of tweets; and lastly in Chapter 6 we have worked on multi-tweet summarisation by presenting a temporal sentiment summarisation

system and studying neural abstractive summarisation for tweets on common topics.

Recognising target-specific sentiment allows researchers to analyse sentiment towards different issues and understand how it evolves over time, on the macro-level. With topical clustering of tweets and opinion summarisation, we provide the explanation or the reason underlying a choice of sentiment towards a particular entity on a particular day, thus offers a micro perspective. Although the approaches presented in this thesis by no means fully solve all these problems, they have showed promising directions for understanding public opinion manifested in the large amount of social media textual data.

In this chapter, we list our main findings and present an outlook on our future research directions. In Section 7.1, we provide a summary of our findings and contributions to each research question listed in Chapter 1. In Section 7.2, we discuss directions for our future work.

7.1 Main Findings

In Chapter 3, we presented two preliminary studies focusing on social spam detection to improve the signal-to-noise ratio for our sentiment corpus, and also a model-based multi-class adaptive-SVM approach to tackle the task of cross-domain emotion classification on Twitter. We begin addressing our main research questions in Chapter 4, where we find existing tweet-level sentiment classification inadequate for identifying different types of sentiment expressed towards all the target entities mentioned in a tweet. Therefore in Chapter 4 we aimed to address the following research question:

RQ1: *How can we infer the sentiment towards a specific target as opposed to tweet-level sentiment? Can we find an effective approach for identifying sentiment towards multiple targets within a tweet?*

In our pilot work, we have experimented several methods for recognising single-target-specific sentiment where each tweet mentions only one target entity. We found our graph kernel based models are outperformed by much simpler tweet-level systems. Subsequently, we introduced the task of multi-target-specific sentiment classification by generating a multi-target Twitter corpus on UK elections. To tackle this challenge, we have proposed an approach which utilises the syntactic information from parse tree in conjunction with the left-right context of the target. We found our proposed model achieving state-of-the-art performances on both single-target and multi-target datasets, even over the more complex neural networks.

We have also showed our multi-target system performs the best for tweets containing two or three different target sentiments, against other target-independent and target-dependent models. However, our approach is limited by its simple way of utilising the syntactic parser, which is to be improved in the future possibly in the expense of acquiring more training data.

Keeping track of all the relevant information provided by large volumes of opinionated tweets is difficult if possible for humans. After studying target-specific sentiment classification, in Chapter 5 we turned our research angle to topical clustering of tweets to alleviate this information overload. Due to the short and noisy nature of tweets, traditional document clustering approaches and conventional topic models fall short of achieving good performance. We were thus motivated to ask the following research question:

RQ2: *Can we develop a system to effectively group tweets to a number of clusters, with each cluster representing a thematic topic?*

To answer the above question, we have proposed a two-stage hierarchical topic modelling system, integrating a state-of-the-art Twitter topic model, a word embedding-incorporated topic model and a tweet pooling step without the use of

any metadata. We have performed extensive evaluations and the results show our proposed system outperform other clustering-based methods and topic models in both clustering performance and topic coherence. In addition, the topics obtained by our system are well aligned with the real-world stories, thus makes it a useful tool for the analysing corresponding events through the lens of social media.

Finally, to improve our understanding of public opinion on Twitter from the micro perspective, in Chapter 6 we have studied the task of summarising tweets on common topics, with the goal of adding justification behind the target-sentiment. We first presented a topic-based temporal summarisation system that provides interactive visualisation of sentiments with corresponding extractive summaries in chronological order. Such extractive summaries unavoidably contain secondary or redundant information. Therefore we aimed to investigate the following research question:

RQ3: *How can we generate abstractive summary for opinions towards common topics expressed on Twitter? Is it possible to generate tweet abstract from scratch with limited training resources?*

Working towards addressing this question, we have identified two challenges: the abstraction of multiple tweets, and insufficient training resource. For the first challenge, we have proposed an extractive-abstractive framework for creating high-quality inputs to a sequence-to-sequence network that allows both copying and generating words. For the second problem, we used a medium-size news article corpus for training, and experimented with various ways of pre-training to alleviate the different domain/medium issue.

We conducted evaluation for both events summarisation where we have human-generated reference summaries, and opinion summarisation where we do not. We found the extractive baselines showing strong performances comparing to

our abstractive neural models. We also didn't observe any noticeable improvement by using pre-training to initialise our seq2seq networks, showing the challenge of transferring information between two different mediums (i.e. from news articles to opinionated tweets) for the abstractive summarisation task. Albeit the difficulty of beating the extractive baselines measured in ROUGE and content-similarity based metrics, we have showed it is possible to generate less extractive summaries using the state-of-the-art seq2seq model.

7.2 Future Directions

In this section we discuss the potential future directions for our three lines of work: sentiment classification, tweets clustering and abstractive summarisation.

7.2.1 Multi-target-specific Sentiment Classification

One of the promising future directions in the area of recognising multi-target-specific sentiment, is to explore sentiment connections among all targets appearing in the same tweet as a multi-target learning task. This is somewhat discussed in a recent study [281] for stance classification, where the authors experimented with standard attentional sequence-to-sequence models for jointly modelling the overall position toward two related targets. Our multi-target election corpus introduced in Chapter 4 poses a more challenging task by having many more targets and target types. This makes the multi-target learning more difficult and a very interesting task for future research and experimentation.

Another direction is to investigate syntax-guided hierarchical architectures for tweets in the context of detecting sentiment for each target mentioned within the tweet. The current trend of using attentional recurrent neural networks has its limitations. The linguistic structure can reduce the search space for optimisation, and is important to understand the relationship among targets, even for tweets.

Lastly, one general future direction of natural language processing is to have a more explicit model of morphology than just character or word composition. Such model will enable the morphologically-aware word representations that can improve the modelling of sentiment for social media posts.

7.2.2 Topical Clustering of Tweets

We think one key aspect of tweet clustering is still the representation or the embedding of tweets, despite the poor performance of tweet2vec based clustering baseline model in our experiments presented in Chapter 5. Recently we have seen several studies [282, 283] using the neural embedding approach for generating topic or sentence representations, which have shown to be a promising direction.

Another key aspect is similarity learning. We plan to experiment with the relaxed version of Word Mover's Distance (RWMD) [284] and develop a distance metric learning model similarly to WMD by measuring the optimal transportation from one document to another. It is worth noting many participating systems (including the best performing ones) of the semantic textual similarity for tweets task in the SemEval-2015 competition [285], have used extensive set of heavily engineered features, which shows the challenge of this task.

Another interesting research area of tweets clustering is to study the modelling of topics over time.

7.2.3 Abstractive Opinion Summarisation on Twitter

The abstractive summarisation of tweets has remained a difficult task. The key challenge, as demonstrated in Chapter 6, is the lack of tweets-summary training data. One approach to alleviate this data issue is to investigate the use of pre-training, transfer learning or multi-task learning (as described in Section 2.4.2), borrowing ideas from neural machine translation. Another approach is to construct

a corpus using techniques like distant supervision[1].

Finally, evaluating the quality of a summary is a difficult task by itself. The widely adopted metrics like ROUGE and METEOR are limited by their inflexibility and inability to measure the semantic similarity between a summary and the original document. While we do think human evaluation is important for judging a summarisation system and we plan to hire human judges for qualitative evaluation of our proposed systems, an alternative to ROUGE but yet effective metric would be very useful for the development of automatic text summarisation. We have seen ongoing efforts to improve on automatic summarisation evaluation measures [276, 287] but much is left for future research.

[1]Hu et al. [286] have built a large scale Chinese short text summarisation corpus. However, a Chinese microblog post (i.e. Weibo) is inherently different to a typical tweet, as one weibo post can contain both one sentence summary and a short paragraph of text.

APPENDIX A

Seeding Keywords for Twitter Data Collection

For many social media mining projects, the ability to collect as much data with as little noise as possible is crucial for producing meaningful results for the downstream tasks. Most research work on Twitter data have used one or multiple of the methods listed below for data collection:

- Keywords filtering.
- User IDs filtering.
- Geo-location filtering.
- All (unfiltered) public tweets within a time period.

Most researches on Twitter socio-political opinion mining have relied on keywords filtering by manually selecting relevant terms or hashtags as data filters. This not only requires one's domain knowledge but is also time consuming and laborious. A seeding algorithm aims to automatically and incrementally generate more keywords from an initial list of seeding keywords, for the purpose of fetching more relevant tweets, with minimal human effort and domain knowledge. In this section we describe our proposed hashtag seeding algorithm for achieving this.

A.0.1 Seeding Hashtags Using Association Rule Learning

A hashtag is a word or unspaced phrase prefixed with the number sign '#'. Hashtags make it possible to group tweets that have a common topic, and therefore it is an effective and convenient way to search for relevant tweets. In [288] the author used two seeding politics-related hashtags and identified a set relevant hashtags with which it co-occurred in at least one tweet, and ranked the results using the Jaccard coefficient. In this work we aim to improve both the quantity and quality of our data collection, by using association rule learning.

Association rule learning is a popular and well researched data mining technique for discovering frequent itemsets and strong association links (in the form of rules) between different arrays of items in large databases by using one or multiple different measures of interestingness. An association rule is the form $\{X\} \rightarrow \{Y\}$, where X is the antecedent item(s) and Y is the derived consequent item(s). It discovers and reveals interesting associations embedded in huge datasets, which may include hidden information that can be useful for decision making. Therefore association rule learning has been employed in many application areas such as market basket analysis, web usage mining and bio-informatics. Here we use it to measure the likelihood of co-occurrence of hashtags. We apply the well known association rule learning algorithm - Apriori [289], to identify more and more relevant hashtags as filters over time from an initial small set of seeding hashtags.

Four measures for filtering useful hashtag associations are used, namely:

- Support threshold, which denotes the frequency counts of the antecedent hashtag(s) (as X) and consequent hashtag(s) (as Y).
- Confidence threshold, which denotes how often a tweet containing X also contains Y. It is an estimation of conditioned probability.

$$Confidence(\ X -> Y\ )\ =\ \frac{Support(\ X\ \bigcup\ Y\ )}{Support(\ X\ )}$$

- Lift ratio, is the confidence of the rule divided by the confidence assuming Y and X are independent (as in some cases if X and Y have high support, we can have a high confidence value even when they are independent). Lift is known to assess the interestingness of a rule. The larger the life ratio, the greater is the strength (or interestingness) off the association.

$$Lift(\ X \text{-}\!> Y\ )\ =\ \frac{Support(\ X\ \bigcup\ Y\ )}{Support(\ X\ )\ \cdot\ Support(\ Y\ )}$$

- Conviction threshold, intuitively, states by what factor the correctness of the rule (as expressed by its confidence) would reduce if X and Y were independent.

$$Conviction(\ X \text{-}\!> Y\ )\ =\ \frac{1\ -\ Support(\ Y\ )}{1\ -\ Confidence(\ X\ \text{-}\!>\ Y\ )}$$

We adopt the Apriori algorithm and adapt it to a process of discovering relevant hashtags, described in the following steps:

1. Manually select a small set of hashtags (e.g. '#ep2014' for the 2014 European Parliament election in the UK) for collecting a set of data to initialise the hashtag seeding process.

2. For every N days depending on the data traffic, we execute one iteration of hashtag seeding process and add the resulting hashtag(s) in the existing data filters:

   2.1. **Frequent Hashtag-set Generation**: Find all the frequent hashtag-set whose Support $\geq$ *minsup*, where *minsup* is the corresponding support threshold. It uses a level-wise generate-and-prune strategy, and can result in a significant reduction in the number of candidate hashtag-set to be considered.

   2.2. **Rule Generation**: Generate rules from the frequent hashtag-set, using

Confidence, Lift and Conviction thresholds. Same as step 2.1 it uses a level-wise approach.

2.3. **New Hashtag Addition**: Identify relevant hashtags from the generated association rules, and add to the set of filters used for harvesting data.

3. Repeat step 2 throughout data collection for snowballing hashtags to follow newly developed or emerging trends, or terminate the the seeding process manually.

A.0.2 Use Case

We collected a set of Twitter data about the 2014 European Parliament election between 26/03/2014 and 12/04/2014 by tracking the hashtag '#ep2014'. We applied our hashtag seeding algorithm on this dataset to extract more hashtags as additional data filters for the purpose of harvesting more data. With 200 as support threshold, 0.85 as confidence threshold, 20.0 as lift threshold and 5.0 as conviction threshold, the program generated the following association rules:

Support Count Threshold = 200

$|C1| = 28954$

$|L1| = 96$

$|C2| = 1516$

$|L2| = 47$

$|C3| = 12$

$|L3| = 1$

Time spent finding frequent itemsets = 0.88 seconds.

Confidence Threshold = 0.8; Lift Threshold = 10.0; Conviction Threshold = 5.0

$\{\text{eudebate}, \text{withjuncker}\} \rightarrow \{\text{ep2014}\}$, conf = 1.00, lift = 14.67, conv = inf

{withjuncker} → {ep2014}, conf = 1.00, lift = 14.67, conv = inf

{wwfpledge} → {ep2014}, conf = 1.00, lift = 14.67, conv = inf

{hrw} → {ep2014}, conf = 1.00, lift = 14.66, conv = 1087.62

{notreeurope} → {ep2014}, conf = 1.00, lift = 14.65, conv = 720.32

{knockthevote} → {ep2014}, conf = 1.00, lift = 14.64, conv = 472.44

{nowschulz} → {ep2014}, conf = 1.00, lift = 14.62, conv = 277.69

{bluehand} → {ukip}, conf = 0.99, lift = 24.36, conv = 72.92

{epduel} → {ep2014}, conf = 0.99, lift = 14.46, conv = 64.61

{edl} → {ukip}, conf = 0.95, lift = 23.33, conv = 17.55

{ue} → {ep2014}, conf = 0.93, lift = 13.58, conv = 12.49

{europa} → {ep2014}, conf = 0.88, lift = 12.97, conv = 8.04

{eu2014} → {ep2014}, conf = 0.87, lift = 12.75, conv = 7.11

{bnp} → {ukip}, conf = 0.86, lift = 21.20, conv = 6.81

Time spent finding association rules = 0.00 second.

We ignored most of the hashtags due to the hashtag being overly-broad and ambiguous such as #hrw or overly-specific such as #wwfpledge. We chose #eudebate, #epduel and #eu2014 as our data filters in addition to #ep2014.

We also think this hashtag association rule learning can be used for understanding emerging events or topics that are only popular online, as well as the political dynamics in the Twittersphere. For example, We observe #ukip is strongly associated with #bluehand[1], #edl and #bnp, in other words, #ukip is very likely to appear in the same tweet with #bluehand, #edl or #bnp in our dataset.

[1] #bluehand is a self-proclaimed "online campaign against left-wing political correctness" movement.

Bibliography

[1] Dong, L., Wei, F., Tan, C., Tang, D., Zhou, M., Xu, K.: Adaptive recursive neural network for target-dependent twitter sentiment classification. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Baltimore, Maryland, Association for Computational Linguistics (June 2014) 49–54

[2] Petrović, S., Osborne, M., Lavrenko, V.: Using paraphrases for improving first story detection in news and twitter. In: Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics (2012) 338–346

[3] Jamie Bartlett, Sid Bennett, R.B., Wibberley, S. In: Virtually Members: The Facebook and Twitter Followers of UK Political Parties, London, United Kingdom, Demos (April 2013)

[4] Zafarani, R., Abbasi, M.A., Liu, H.: Social media mining: an introduction. Cambridge University Press (2014)

[5] Diao, Q., Jiang, J., Zhu, F., Lim, E.P.: Finding bursty topics from microblogs. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1, Association for Computational Linguistics (2012) 536–544

[6] Yan, X., Guo, J., Lan, Y., Xu, J., Cheng, X.: A probabilistic model for bursty topic discovery in microblogs. In: AAAI. (2015) 353–359

[7] Pennacchiotti, M., Popescu, A.M.: A machine learning approach to twitter user classification. Icwsm **11**(1) (2011) 281–288

[8] Pennacchiotti, M., Popescu, A.M.: Democrats, republicans and starbucks afficionados: user classification in twitter. In: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM (2011) 430–438

[9] Wagner, C., Asur, S., Hailpern, J.: Religious politicians and creative photographers: Automatic user categorization in twitter. In: Social Computing (SocialCom), 2013 International Conference on, IEEE (2013) 303–310

[10] Go, A., Bhayani, R., Huang, L.: Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford **1**(2009) (2009) 12

[11] Jiang, L., Yu, M., Zhou, M., Liu, X., Zhao, T.: Target-dependent twitter sentiment classification. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, Oregon, USA, Association for Computational Linguistics (June 2011) 151–160

[12] Balabantaray, R.C., Mohammad, M., Sharma, N.: Multi-class twitter emotion classification: A new approach. International Journal of Applied Information Systems **4**(1) (2012) 48–53

[13] Tumasjan, A., Sprenger, T.O., Sandner, P.G., Welpe, I.M.: Predicting elections with twitter: What 140 characters reveal about political sentiment. ICWSM **10** (May 2010) 178–185

[14] Culotta, A.: Towards detecting influenza epidemics by analyzing twitter messages. In: Proceedings of the first workshop on social media analytics, ACM (2010) 115–122

[15] Bollen, J., Mao, H., Zeng, X.: Twitter mood predicts the stock market. Journal of computational science **2**(1) (2011) 1–8

[16] Wang, H., Can, D., Kazemzadeh, A., Bar, F., Narayanan, S.: A system for real-time twitter sentiment analysis of 2012 u.s. presidential election cycle. In: Proceedings of the ACL 2012 System Demonstrations, Jeju Island, Korea, Association for Computational Linguistics (July 2012) 115–120

[17] Hu, X., Tang, L., Tang, J., Liu, H.: Exploiting social relations for sentiment analysis in microblogging. In: Proceedings of the sixth ACM international conference on Web search and data mining, ACM (2013) 537–546

[18] Purver, M., Battersby, S.: Experimenting with distant supervision for emotion classification. In: Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, Association for Computational Linguistics (2012) 482–491

[19] Wang, W., Chen, L., Thirunarayan, K., Sheth, A.P.: Harnessing twitter" big data" for automatic emotion identification. In: Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Confernece on Social Computing (SocialCom), IEEE (2012) 587–592

[20] Vargas, S., McCreadie, R., Macdonald, C., Ounis, I.: Comparing overall and targeted sentiments in social media during crises. In: Proceedings of the Tenth International Conference on Web and Social Media, Cologne, Germany, May 17-20, 2016. (2016) 695–698

[21] Rosa, K.D., Shah, R., Lin, B., Gershman, A., Frederking, R.: Topical clustering of tweets. Proceedings of the ACM SIGIR: SWSM (2011)

[22] Aiello, L.M., Petkos, G., Martin, C., Corney, D., Papadopoulos, S., Skraba, R., Göker, A., Kompatsiaris, I., Jaimes, A.: Sensing trending topics in twitter. IEEE Transactions on Multimedia **15**(6) (2013) 1268–1282

[23] Yin, J., Wang, J.: A dirichlet multinomial mixture model-based approach for short text clustering. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM (2014) 233–242

[24] Inouye, D., Kalita, J.K.: Comparing twitter summarization algorithms for multiple post summaries. In: Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on, IEEE (2011) 298–306

[25] Meng, X., Wei, F., Liu, X., Zhou, M., Li, S., Wang, H.: Entity-centric topic-oriented opinion summarization in twitter. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '12, New York, NY, USA, ACM (2012) 379–387

[26] Ma, S., Sun, X., Xu, J., Wang, H., Li, W., Su, Q.: Improving semantic relevance for sequence-to-sequence learning of chinese social media text summarization. In: Proceedings of The 55th Annual Meeting of the Association for Computational Linguistics (ACL). (2017)

[27] Wang, A.H.: Don't follow me: Spam detection in twitter. In: Security and Cryptography (SECRYPT), Proceedings of the 2010 International Conference on, IEEE (2010) 1–10

[28] McCord, M., Chuah, M.: Spam detection on twitter using traditional classifiers. In Calero, J.M.A., Yang, L.T., Mármol, F.G., Garcá-Villalba, L.J., Li, X.A., 0002, Y.W., eds.: ATC. Volume 6906 of Lecture Notes in Computer Science., Springer (2011) 175–186

[29] Martinez-Romo, J., Araujo, L.: Detecting malicious tweets in trending topics using a statistical analysis of language. Expert Systems with Applications **40**(8) (2013) 2992 – 3000

[30] Peddinti, V.M.K., Chintalapoodi, P.: Domain adaptation in sentiment analysis of twitter. In: AAAI Workshops. (2011)

[31] Tsakalidis, A., Papadopoulos, S., Kompatsiaris, I.: An ensemble model for cross-domain polarity classification on twitter. In: WISE. Springer (2014) 168–177

[32] Barbieri, F., Saggion, H., Ronzano, F.: Modelling sarcasm in twitter, a novel approach. In: WASSA@ ACL. (2014) 50–58

[33] Zubiaga, A., Liakata, M., Procter, R., Bontcheva, K., Tolmie, P.: Towards detecting rumours in social media. In: AAAI Workshop on AI for Cities. (2015)

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

[35] Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in neural information processing systems. (2014) 3104–3112

[36] Wang, B., Zubiaga, A., Liakata, M., Procter, R.: Making the most of tweet-inherent features for social spam detection on twitter. In: 5th Workshop on Making Sense of Microposts (#Microposts2015) WWW. Volume 1395. (2015) 10–16

[37] Wang, B., Liakata, M., Zubiaga, A., Procter, R., Jensen, E.: Smile: Twitter emotion classification using domain adaptation. In: CEUR Workshop Proceedings. Volume 1619., Sun SITE Central Europe (2016) 15–21

[38] Wang, B., Liakata, M., Zubiaga, A., Procter, R.: Tdparse-multi-target-specific sentiment recognition on twitter. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics. (2017)

[39] Townsend, R., Tsakalidis, A., Zhou, Y., Wang, B., Liakata, M., Zubiaga, A., Cristea, A.I., Procter, R.: Warwickdcs: From phrase-based to target-specific sentiment recognition. In: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Association for Computational Linguistics (2015) 657–663

[40] Wang, B., Liakata, M., Zubiaga, A., Procter, R.: A hierarchical topic modelling approach for tweet clustering. In: International Conference on Social Informatics, Springer (2017) 378–390

[41] Wang, B., Liakata, M., Tsakalidis, A., Georgakopoulos Kolaitis, S., Papadopoulos, S., Apostolidis, L., Zubiaga, A., Procter, R., Kompatsiaris, Y.: Totemss: Topic-based, temporal sentiment summarisation for twitter. In: Proceedings of the 8th International Joint Conference on Natural Language Processing (IJCNLP). (2017)

[42] Zubiaga, A., Voss, A., Procter, R., Liakata, M., Wang, B., Tsakalidis, A.: Towards real-time, country-level location classification of worldwide tweets. IEEE Transactions on Knowledge and Data Engineering (2017)

[43] Zubiaga, A., Wang, B., Liakata, M., Procter, R.: Political homophily in independence movements: Analysing and classifying social media users by national identity. IEEE Intelligent Systems (2018)

[44] Carreras, X., Marquez, L.S., Salgado, J.G.: Boosting trees for anti-spam email filtering. In: Proceedings of RANLP, Citeseer (2001)

[45] Blanzieri, E., Bryl, A.: A survey of learning-based techniques of email spam filtering. Artificial Intelligence Review **29**(1) (2008) 63–92

[46] Benevenuto, F., Magno, G., Rodrigues, T., Almeida, V.: Detecting spammers on twitter. In: Proceedings of CEAS. (2010)

[47] Lee, K., Eoff, B.D., Caverlee, J.: Seven months with the devils: A long-term study of content polluters on twitter. In: ICWSM. (2011)

[48] Yang, C., Harkreader, R.C., Gu, G.: Die free or live hard? empirical evaluation and new design for fighting evolving twitter spammers. In: Proceedings of RAID. RAID'11, Berlin, Heidelberg, Springer-Verlag (2011) 318–337

[49] Ferrara, E., Varol, O., Davis, C., Menczer, F., Flammini, A.: The rise of social bots. CoRR **abs/1407.5225** (2014)

[50] Miller, Z., Dickinson, B., Deitrick, W., Hu, W., Wang, A.H.: Twitter spammer detection using data stream clustering. Information Sciences **260**(0) (2014) 64 – 73

[51] Santos, I., Miñambres-Marcos, I., Laorden, C., Galán-García, P., Santamaría-Ibirika, A., Bringas, P.G.: Twitter content-based spam filtering. In: International Joint Conference SOCO'13-CISIS'13-ICEUTE'13, Springer (2014) 449–458

[52] Feldman, R.: Techniques and applications for sentiment analysis. Communications of the ACM **56**(4) (2013) 82–89

[53] Asur, S., Huberman, B.A.: Predicting the future with social media. In: Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010

IEEE/WIC/ACM International Conference on. Volume 1., IEEE (2010) 492–499

[54] Liakata, M., Kim, J.H., Saha, S., Hastings, J., Rebholz-Schuhmann, D.: Three hybrid classifiers for the detection of emotions in suicide notes. Biomedical informatics insights **5**(Suppl. 1) (2012) 175

[55] Kumar, A., Sebastian, T.M.: Sentiment analysis: A perspective on its past, present and future. International Journal of Intelligent Systems and Applications (IJISA) **4**(10) (2012) 1

[56] Taboada, M., Brooke, J., Tofiloski, M., Voll, K., Stede, M.: Lexicon-based methods for sentiment analysis. Computational linguistics **37**(2) (2011) 267–307

[57] Zhang, L., Ghosh, R., Dekhil, M., Hsu, M., Liu, B.: Combining lexicon-based and learning-based methods for twitter sentiment analysis. HP Laboratories, Technical Report HPL-2011 **89** (2011)

[58] Mohammad, S., Kiritchenko, S., Zhu, X.: Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. In: Proceedings of the seventh international workshop on Semantic Evaluation Exercises (SemEval-2013), Atlanta, Georgia, USA (June 2013)

[59] Zhu, X., Kiritchenko, S., Mohammad, S.: Nrc-canada-2014: Recent improvements in the sentiment analysis of tweets. In: Proceedings of SemEval, Dublin, Ireland (August 2014) 443–447

[60] Tang, D., Qin, B., Wei, F., Dong, L., Liu, T., Zhou, M.: A joint segmentation and classification framework for sentence level sentiment classification. IEEE/ACM Transactions on Audio, Speech, and Language Processing **23**(11) (2015) 1750–1761

[61] Saif, H., He, Y., Alani, H.: Semantic sentiment analysis of twitter. The Semantic Web–ISWC 2012 (2012) 508–524

[62] Xiang, B., Zhou, L.: Improving twitter sentiment analysis with topic-based mixture modeling and semi-supervised training. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). Volume 2. (2014) 434–439

[63] Calais Guerra, P.H., Veloso, A., Meira Jr, W., Almeida, V.: From bias to opinion: a transfer-learning approach to real-time sentiment analysis. In: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM (2011) 150–158

[64] Li, H., Chen, Y., Ji, H., Muresan, S., Zheng, D.: Combining social cognitive theories with linguistic features for multi-genre sentiment analysis. In: PACLIC. (2012) 127–136

[65] Severyn, A., Moschitti, A.: Twitter sentiment analysis with deep convolutional neural networks. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM (2015) 959–962

[66] Ren, Y., Zhang, Y., Zhang, M., Ji, D.: Context-sensitive twitter sentiment classification using neural network. In: AAAI. (2016) 215–221

[67] Yang, Y., Eisenstein, J.: Overcoming language variation in sentiment analysis with social attention. Transactions of the Association for Computational Linguistics **5** (2017) 295–307

[68] Tang, D., Wei, F., Qin, B., Liu, T., Zhou, M.: Coooolll: A deep learning system for Twitter sentiment classification. SemEval 2014 (2014) 208

[69] Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., Qin, B.: Learning sentiment-specific word embedding for twitter sentiment classification. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Baltimore, Maryland, Association for Computational Linguistics (June 2014) 1555–1565

[70] Cortes, C., Vapnik, V.: Support-vector networks. Machine learning **20**(3) (1995) 273–297

[71] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11) (1998) 2278–2324

[72] Elman, J.L.: Finding structure in time. Cognitive science **14**(2) (1990) 179–211

[73] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. (2014)

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

[75] Collobert, R., Weston, J.: A unified architecture for natural language processing: Deep neural networks with multitask learning. In: Proceedings of the 25th international conference on Machine learning, ACM (2008) 160–167

[76] Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. CoRR **abs/1301.3781** (2013)

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

[78] Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Empirical Methods in Natural Language Processing (EMNLP). (2014) 1532–1543

[79] Blitzer, J., Dredze, M., Pereira, F., et al.: Biographies, bollywood, boomboxes and blenders: Domain adaptation for sentiment classification. In: ACL. Volume 7. (2007) 440–447

[80] Townsend, R., Kalair, A., Kulkarni, O., Procter, R., Liakata, M.: University of warwick: Sentiadaptron-a domain adaptable sentiment analyser for tweets-meets semeval. SemEval 2014 (2014) 768

[81] Pan, S.J., Ni, X., Sun, J.T., Yang, Q., Chen, Z.: Cross-domain sentiment classification via spectral feature alignment. In: WWW, ACM (2010) 751–760

[82] Bollegala, D., Weir, D., Carroll, J.: Using multiple sources to construct a sentiment sensitive thesaurus for cross-domain sentiment classification. In: NAACL HLT, Association for Computational Linguistics (2011) 132–141

[83] Liu, S., Li, F., Li, F., Cheng, X., Shen, H.: Adaptive co-training svm for sentiment classification on tweets. In: CIKM, ACM (2013) 2079–2088

[84] Jiang, J., Zhai, C.: Instance weighting for domain adaptation in nlp. In: ACL, Association for Computational Linguistics (June 2007) 264–271

[85] Xia, R., Yu, J., Xu, F., Wang, S.: Instance-based domain adaptation in nlp via in-target-domain logistic approximation. In: AAAI. (2014)

[86] Mejova, Y., Srinivasan, P.: Crossing media streams with sentiment: Domain adaptation in blogs, reviews and twitter. In: ICWSM. (2012)

[87] Chen, F., Mirisaee, S.H.: Do topic-dependent models improve microblog sentiment estimation? In: ICWSM. (2014)

[88] Ruder, S., Ghaffari, P., Breslin, J.G.: Data selection strategies for multi-domain sentiment analysis. arXiv preprint arXiv:1702.02426 (2017)

[89] Ruder, S., Plank, B.: Learning to select data for transfer learning with bayesian optimization. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics (2017)

[90] Glorot, X., Bordes, A., Bengio, Y.: Domain adaptation for large-scale sentiment classification: A deep learning approach. In: ICML. (2011) 513–520

[91] Li, Z., Zhang, Y., Wei, Y., Wu, Y., Yang, Q.: End-to-end adversarial memory network for cross-domain sentiment classification. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17. (2017) 2237–2243

[92] Yang, W., Lu, W., Zheng, V.: A simple regularization-based algorithm for learning cross-domain word embeddings. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, Association for Computational Linguistics (September 2017) 2898–2904

[93] Felbo, B., Mislove, A., Søgaard, A., Rahwan, I., Lehmann, S.: Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics (2017) 1615–1625

[94] Yang, J., Yan, R., Hauptmann, A.G.: Cross-domain video concept detection using adaptive svms. In: Proceedings of the 15th international conference on Multimedia, ACM (2007) 188–197

[95] Yang, J., Hauptmann, A.G.: A framework for classifier adaptation and its applications in concept detection. In: MIR, ACM (2008) 467–474

[96] Rosenthal, S., Nakov, P., Kiritchenko, S., Mohammad, S., Ritter, A., Stoyanov, V.: Semeval-2015 task 10: Sentiment analysis in twitter. In: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Denver, Colorado, Association for Computational Linguistics (June 2015) 451–463

[97] Boag, W., Potash, P., Rumshisky, A.: Twitterhawk: A feature bucket based approach to sentiment analysis. In: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Denver, Colorado, Association for Computational Linguistics (June 2015) 640–646

[98] Plotnikova, N., Kohl, M., Volkert, K., Evert, S., Lerner, A., Dykes, N., Ermer, H.: Klueless: Polarity classification and association. In: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Denver, Colorado, Association for Computational Linguistics (June 2015) 619–625

[99] Nakov, P., Ritter, A., Rosenthal, S., Sebastiani, F., Stoyanov, V.: Semeval-2016 task 4: Sentiment analysis in twitter. In: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), San Diego, California, Association for Computational Linguistics (June 2016) 1–18

[100] Pontiki, M., Galanis, D., Papageorgiou, H., Manandhar, S., Androutsopoulos, I.: Semeval-2015 task 12: Aspect based sentiment analysis. In: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015),

Denver, Colorado, Association for Computational Linguistics (June 2015) 486–495

[101] Pateria, S., Choubey, P.: AKTSKI at semeval-2016 task 5: Aspect based sentiment analysis for consumer reviews. In: Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016. (2016) 318–324

[102] Socher, R., Lin, C.C., Manning, C., Ng, A.Y.: Parsing natural scenes and natural language with recursive neural networks. In: Proceedings of the 28th international conference on machine learning (ICML-11). (2011) 129–136

[103] Vo, D.T., Zhang, Y.: Target-dependent twitter sentiment classification with rich automatic features. In: Proceedings of the 24th International Conference on Artificial Intelligence. IJCAI'15, AAAI Press (2015) 1347–1353

[104] Tang, D., Qin, B., Feng, X., Liu, T.: Effective lstms for target-dependent sentiment classification. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, Osaka, Japan, The COLING 2016 Organizing Committee (December 2016) 3298–3307

[105] Zhang, M., Zhang, Y., Vo, D.T.: Gated neural networks for targeted sentiment analysis. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence. AAAI'16, AAAI Press (2016) 3087–3093

[106] Tang, D., Qin, B., Liu, T.: Aspect level sentiment classification with deep memory network. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, Texas, Association for Computational Linguistics (November 2016) 214–224

[107] Lakkaraju, H., Socher, R., Manning, C.: Aspect specific sentiment analysis using hierarchical deep learning. In: NIPS Workshop on Deep Learning and Representation Learning. (2014)

[108] Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C.D., Ng, A.Y., Potts, C.: Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Stroudsburg, PA, Association for Computational Linguistics (October 2013) 1631–1642

[109] Nguyen, T.H., Shirai, K.: Phrasernn: Phrase recursive neural network for aspect-based sentiment analysis. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, Association for Computational Linguistics (September 2015) 2509–2514

[110] Mou, L., Peng, H., Li, G., Xu, Y., Zhang, L., Jin, Z.: Discriminative neural sentence modeling by tree-based convolution. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, Association for Computational Linguistics (September 2015) 2315–2325

[111] Kong, L., Schneider, N., Swayamdipta, S., Bhatia, A., Dyer, C., Smith, N.A.: A dependency parser for tweets. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, Association for Computational Linguistics (October 2014) 1001–1012

[112] Le, P., Zuidema, W.: The forest convolutional network: Compositional distributional semantics with a neural chart and without binarization. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, Association for Computational Linguistics (September 2015) 1155–1164

[113] Allan, J.: Topic detection and tracking: event-based information organization. Volume 12. Springer Science & Business Media (2012)

[114] Phuvipadawat, S., Murata, T.: Breaking news detection and tracking in twitter. In: Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on. Volume 3., IEEE (2010) 120–123

[115] Yin, J.: Clustering microtext streams for event identification. In: IJCNLP. (2013) 719–725

[116] Zhou, Y., Kanhabua, N., Cristea, A.I.: Real-time timeline summarisation for high-impact events in twitter. In: Proceedings of the 22nd European Conference on Artificial Intelligence. Volume 285., IOS Press (2016) 1158–1166

[117] Becker, H., Naaman, M., Gravano, L.: Beyond trending topics: Real-world event identification on twitter. ICWSM **11**(2011) (2011) 438–441

[118] Ifrim, G., Shi, B., Brigadir, I.: Event detection in twitter using aggressive filtering and hierarchical tweet clustering. In: Second Workshop on Social News on the Web (SNOW), Seoul, Korea, 8 April 2014, ACM (2014)

[119] Rangrej, A., Kulkarni, S., Tendulkar, A.V.: Comparative study of clustering techniques for short text documents. In: Proceedings of the 20th international conference companion on World wide web, ACM (2011) 111–112

[120] Petrović, S., Osborne, M., Lavrenko, V.: Streaming first story detection with application to twitter. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Association for Computational Linguistics (2010) 181–189

[121] Tsur, O., Littman, A., Rappoport, A.: Scalable multi stage clustering of tagged micro-messages. In: Proceedings of the 21st International Conference on World Wide Web, ACM (2012) 621–622

[122] Ganesh, J., Gupta, M., Varma, V.: Interpreting the syntactic and social elements of the tweet representations via elementary property prediction tasks. NIPS Workshop on Interpretable Machine Learning in Complex Systems (2016)

[123] Kiros, R., Zhu, Y., Salakhutdinov, R.R., Zemel, R., Urtasun, R., Torralba, A., Fidler, S.: Skip-thought vectors. In: Advances in neural information processing systems. (2015) 3294–3302

[124] Vakulenko, S., Nixon, L., Lupu, M.: Character-based neural embeddings for tweet clustering. SocialNLP 2017 (2017) 36

[125] Dhingra, B., Zhou, Z., Fitzpatrick, D., Muehl, M., Cohen, W.W.: Tweet2vec: Character-based distributed representations for social media. In: The 54th Annual Meeting of the Association for Computational Linguistics. (2016) 269

[126] Arora, S., Liang, Y., Ma, T.: A simple but tough-to-beat baseline for sentence embeddings. 5th International Conference on Learning Representations (ICLR) (2017)

[127] Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of machine Learning research **3**(Jan) (2003) 993–1022

[128] Phan, X.H., Nguyen, L.M., Horiguchi, S.: Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In: Proceedings of the 17th International Conference on World Wide Web, ACM (2008) 91–100

[129] Yan, X., Guo, J., Lan, Y., Cheng, X.: A biterm topic model for short texts. In: Proceedings of the 22nd International Conference on World Wide Web, ACM (2013) 1445–1456

[130] Weng, J., Lim, E.P., Jiang, J., He, Q.: Twitterrank: finding topic-sensitive influential twitterers. In: Proceedings of the 3rd ACM International Conference on Web Search and Data Mining, ACM (2010) 261–270

[131] Mehrotra, R., Sanner, S., Buntine, W., Xie, L.: Improving lda topic models for microblogs via tweet pooling and automatic labeling. In: Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval, ACM (2013) 889–892

[132] Alvarez-Melis, D., Saveski, M.: Topic modeling in twitter: Aggregating tweets by conversations. In: ICWSM. (2016) 519–522

[133] Hong, L., Davison, B.D.: Empirical study of topic modeling in twitter. In: Proceedings of the first workshop on social media analytics, ACM (2010) 80–88

[134] Nigam, K., McCallum, A.K., Thrun, S., Mitchell, T.: Text classification from labeled and unlabeled documents using em. Machine learning **39**(2) (2000) 103–134

[135] Zhao, W.X., Jiang, J., Weng, J., He, J., Lim, E.P., Yan, H., Li, X.: Comparing twitter and traditional media using topic models. In: European Conference on Information Retrieval, Springer (2011) 338–349

[136] Quan, X., Kit, C., Ge, Y., Pan, S.J.: Short and sparse text topic modeling via self-aggregation. In: IJCAI. (2015) 2270–2276

[137] Li, C., Wang, H., Zhang, Z., Sun, A., Ma, Z.: Topic modeling for short texts with auxiliary word embeddings. In: Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, ACM (2016) 165–174

[138] Nguyen, D.Q., Billingsley, R., Du, L., Johnson, M.: Improving topic models with latent feature word representations. Transactions of the Association for Computational Linguistics **3** (2015) 299–313

[139] Hu, W., Tsujii, J.: A latent concept topic model for robust topic inference using word embeddings. In: The 54th Annual Meeting of the Association for Computational Linguistics. (2016) 380

[140] McMinn, A.J., Moshfeghi, Y., Jose, J.M.: Building a large-scale corpus for evaluating event detection on twitter. In: Proceedings of the 22nd ACM international conference on Information & Knowledge Management, ACM (2013) 409–418

[141] McLachlan, G.J., Basford, K.E.: Mixture models. inference and applications to clustering. Statistics: Textbooks and Monographs, New York: Dekker, 1988 (1988)

[142] Wallach, H.M., Murray, I., Salakhutdinov, R., Mimno, D.: Evaluation methods for topic models. In: Proceedings of the 26th annual international conference on machine learning, ACM (2009) 1105–1112

[143] Chang, J., Gerrish, S., Wang, C., Boyd-Graber, J.L., Blei, D.M.: Reading tea leaves: How humans interpret topic models. In: Advances in neural information processing systems. (2009) 288–296

[144] Lau, J.H., Newman, D., Baldwin, T.: Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In: EACL. (2014) 530–539

[145] Bouma, G.: Normalized (pointwise) mutual information in collocation extraction. Proceedings of GSCL (2009) 31–40

[146] Joachims, T.: Training linear svms in linear time. In: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM (2006) 217–226

[147] Newman, D., Lau, J.H., Grieser, K., Baldwin, T.: Automatic evaluation of topic coherence. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Association for Computational Linguistics (2010) 100–108

[148] Lau, J.H., Baldwin, T.: The sensitivity of topic coherence evaluation to topic cardinality. In: Proceedings of NAACL-HLT. (2016) 483–487

[149] Mimno, D., Wallach, H.M., Talley, E., Leenders, M., McCallum, A.: Optimizing semantic coherence in topic models. In: Proceedings of the conference on empirical methods in natural language processing, Association for Computational Linguistics (2011) 262–272

[150] Aletras, N., Stevenson, M.: Evaluating topic coherence using distributional semantics. In: IWCS. Volume 13. (2013) 13–22

[151] Fang, A., Macdonald, C., Ounis, I., Habel, P.: Topics in tweets: A user study of topic coherence metrics for twitter data. In: European Conference on Information Retrieval, Springer (2016) 492–504

[152] Fang, A., Macdonald, C., Ounis, I., Habel, P.: Using word embedding to evaluate the coherence of topics from twitter data. In: Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, ACM (2016) 1057–1060

[153] Nenkova, A., McKeown, K., et al.: Automatic summarization. Foundations and Trends® in Information Retrieval **5**(2–3) (2011) 103–233

[154] Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM (2004) 168–177

[155] Zhai, Z., Liu, B., Xu, H., Jia, P.: Clustering product features for opinion mining. In: Proceedings of the fourth ACM international conference on Web search and data mining, ACM (2011) 347–354

[156] Wu, H., Gu, Y., Sun, S., Gu, X.: Aspect-based opinion summarization with convolutional neural networks. In: Neural Networks (IJCNN), 2016 International Joint Conference on, IEEE (2016) 3157–3163

[157] Kim, H.D., Ganesan, K., Sondhi, P., Zhai, C.: Comprehensive review of opinion summarization. Technical report, University of Illinois at UrbanaChampaign (2011)

[158] Luhn, H.P.: The automatic creation of literature abstracts. IBM Journal of research and development **2**(2) (1958) 159–165

[159] Radev, D.R., Jing, H., Styś, M., Tam, D.: Centroid-based summarization of multiple documents. Information Processing & Management **40**(6) (2004) 919–938

[160] Erkan, G., Radev, D.R.: Lexrank: Graph-based lexical centrality as salience in text summarization. Journal of Artificial Intelligence Research **22** (2004) 457–479

[161] Chang, Y.L., Chien, J.T.: Latent dirichlet learning for document summarization. In: Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on, IEEE (2009) 1689–1692

[162] Goldstein, J., Mittal, V., Carbonell, J., Kantrowitz, M.: Multi-document summarization by sentence extraction. In: Proceedings of the 2000 NAACL-

ANLPWorkshop on Automatic summarization-Volume 4, Association for Computational Linguistics (2000) 40–48

[163] Nenkova, A., Vanderwende, L.: The impact of frequency on summarization. Microsoft Research, Redmond, Washington, Tech. Rep. MSR-TR-2005 **101** (2005)

[164] Gillick, D., Favre, B.: A scalable global model for summarization. In: Proceedings of the Workshop on Integer Linear Programming for Natural Langauge Processing, Association for Computational Linguistics (2009) 10–18

[165] Lin, H., Bilmes, J.: Multi-document summarization via budgeted maximization of submodular functions. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Association for Computational Linguistics (2010) 912–920

[166] Kobayashi, H., Noguchi, M., Yatsuka, T.: Summarization based on embedding distributions. In: EMNLP. (2015) 1984–1989

[167] Fuentes, M., Alfonseca, E., Rodríguez, H.: Support vector machines for query-focused summarization trained and evaluated on pyramid data. In: Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, Association for Computational Linguistics (2007) 57–60

[168] Cheng, J., Lapata, M.: Neural summarization by extracting sentences and words. In: Proceedings of The 54th Annual Meeting of the Association for Computational Linguistics (ACL). (2016)

[169] Nallapati, R., Zhai, F., Zhou, B.: Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In: AAAI. (2017) 3075–3081

[170] Isonuma, M., Fujino, T., Mori, J., Matsuo, Y., Sakata, I.: Extractive summarization using multi-task learning with document classification. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. (2017) 2091–2100

[171] Cao, Z., Li, W., Li, S., Wei, F.: Improving multi-document summarization via text classification. In: AAAI. (2017) 3053–3059

[172] Nenkova, A., McKeown, K.: A survey of text summarization techniques. Mining text data (2012) 43–76

[173] Mehta, P.: From extractive to abstractive summarization: A journey. ACL Student Research Workshop (2016) 100

[174] Wang, L.: Summarization and Sentiment Analysis for Understanding Socially-Generated Content. PhD thesis, Cornell University (2016)

[175] Knight, K., Marcu, D.: Statistics-based summarization-step one: Sentence compression. AAAI/IAAI **2000** (2000) 703–710

[176] Knight, K., Marcu, D.: Summarization beyond sentence extraction: A probabilistic approach to sentence compression. Artificial Intelligence **139**(1) (2002) 91–107

[177] Ganesan, K., Zhai, C., Han, J.: Opinosis: a graph-based approach to abstractive summarization of highly redundant opinions. In: Proceedings of the 23rd international conference on computational linguistics, Association for Computational Linguistics (2010) 340–348

[178] Banerjee, S., Mitra, P., Sugiyama, K.: Multi-document abstractive summarization using ilp based multi-sentence compression. In: IJCAI. (2015) 1208–1214

[179] Oya, T., Mehdad, Y., Carenini, G., Ng, R.: A template-based abstractive meeting summarization: Leveraging summary and source text relationships. In: Proceedings of the 8th International Natural Language Generation Conference (INLG). (2014) 45–53

[180] Gerani, S., Mehdad, Y., Carenini, G., Ng, R.T., Nejat, B.: Abstractive summarization of product reviews using discourse structure. In: EMNLP. Volume 14. (2014) 1602–1613

[181] Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: ICLR. (2015)

[182] Rush, A.M., Chopra, S., Weston, J.: A neural attention model for abstractive sentence summarization. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. (2015) 379–389

[183] Chopra, S., Auli, M., Rush, A.M.: Abstractive sentence summarization with attentive recurrent neural networks. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. (2016) 93–98

[184] Nallapati, R., Zhou, B., Gulcehre, C., Xiang, B., et al.: Abstractive text summarization using sequence-to-sequence rnns and beyond. In: Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL). (2016) 280–290

[185] See, A., Liu, P.J., Manning, C.D.: Get to the point: Summarization with pointer-generator networks. In: Proceedings of The 55th Annual Meeting of the Association for Computational Linguistics (ACL). (2017)

[186] Paulus, R., Xiong, C., Socher, R.: A deep reinforced model for abstractive summarization. arXiv preprint arXiv:1705.04304 (2017)

[187] Wang, L., Ling, W.: Neural network-based abstract generation for opinions and arguments. In: Proceedings of NAACL-HLT. (2016) 47–57

[188] Sandhaus, E.: The new york times annotated corpus. Linguistic Data Consortium, Philadelphia **6**(12) (2008) e26752

[189] Dai, A.M., Le, Q.V.: Semi-supervised sequence learning. In: Advances in Neural Information Processing Systems. (2015) 3079–3087

[190] Miyato, T., Dai, A.M., Goodfellow, I.: Adversarial training methods for semi-supervised text classification. In: ICLR. (2017)

[191] Ramachandran, P., Liu, P.J., Le, Q.V.: Unsupervised pretraining for sequence to sequence learning. arXiv preprint arXiv:1611.02683 (2016)

[192] Tilk, O., Alumäe, T.: Low-resource neural headline generation. In: Proceedings of the Workshop on New Frontiers in Summarization (EMNLP). (2017)

[193] Zoph, B., Yuret, D., May, J., Knight, K.: Transfer learning for low-resource neural machine translation. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. (2016) 1568–1575

[194] Nguyen, T.Q., Chiang, D.: Transfer learning across low-resource, related languages for neural machine translation. In: Proceedings of the 8th International Joint Conference on Natural Language Processing (IJCNLP). (2017)

[195] Hua, X., Wang, L.: A pilot study of domain adaptation effect for neural abstractive summarization. In: Proceedings of the Workshop on New Frontiers in Summarization (EMNLP). (2017)

[196] O'Connor, B., Balasubramanyan, R., Routledge, B.R., Smith, N.A.: From tweets to polls: Linking text sentiment to public opinion time series. ICWSM **11**(122-129) (2010) 1–2

[197] Lerman, K., Blair-Goldensohn, S., McDonald, R.: Sentiment summarization: evaluating and learning user preferences. In: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics, Association for Computational Linguistics (2009) 514–522

[198] LOUIS, A., NEWMAN, T.: Summarization of business-related tweets: A concept-based approach. In: 24th International Conference on Computational Linguistics. (2012) 765

[199] Rudra, K., Banerjee, S., Ganguly, N., Goyal, P., Imran, M., Mitra, P.: Summarizing situational tweets in crisis scenario. In: Proceedings of the 27th ACM Conference on Hypertext and Social Media, ACM (2016) 137–147

[200] Ghosh, S., Ghosh, K., Ganguly, D., Chakraborty, T., Jones, G.J., Moens, M.F.: Ecir 2017 workshop on exploitation of social media for emergency relief and preparedness (smerp 2017). In: ACM SIGIR Forum. Volume 51., ACM (2017) 36–41

[201] Takamura, H., Yokono, H., Okumura, M.: Summarizing a document stream. In: ECIR, Springer (2011) 177–188

[202] Chakrabarti, D., Punera, K.: Event summarization using tweets. ICWSM **11** (2011) 66–73

[203] Sharifi, B., Hutton, M.A., Kalita, J.K.: Experiments in microblog summarization. In: Social Computing (SocialCom), 2010 IEEE Second International Conference on, IEEE (2010) 49–56

[204] Liu, F., Liu, Y., Weng, F.: Why is sxsw trending? exploring multiple text sources for twitter topic summarization. In: Proceedings of the Workshop on Languages in Social Media, Association for Computational Linguistics (2011) 66–75

[205] Lin, J., Efron, M., Wang, Y., Sherman, G., Voorhees, E.: Overview of the trec-2015 microblog track. In: Proceedings of the 24th Text REtrieval Conference, TREC. (2015)

[206] Aslam, J., Diaz, F., Ekstrand-Abueg, M., McCreadie, R., Pavlu, V., Sakai, T.: Trec 2015 temporal summarization track overview. In: Proceedings of the 24th Text REtrieval Conference, TREC. (2015)

[207] Lin, J., Roegiest, A., Tan, L., McCreadie, R., Voorhees, E., Diaz, F.: Overview of the trec 2016 real-time summarization track. In: Proceedings of the 25th Text REtrieval Conference, TREC. Volume 16. (2016)

[208] O'Connor, B., Krieger, M., Ahn, D.: Tweetmotif: Exploratory search and topic summarization for twitter. In: ICWSM. (2010) 384–385

[209] Xu, W., Grishman, R., Meyers, A., Ritter, A.: A preliminary study of tweet summarization using information extraction. NAACL 2013 (2013) 20

[210] Sharifi, B.: Automatic microblog classification and summarization. Unpublished masters thesis, University of Colorado, Colorado Springs, CO, USA (2010)

[211] Vanderwende, L., Suzuki, H., Brockett, C., Nenkova, A.: Beyond sumbasic: Task-focused summarization with sentence simplification and lexical expansion. Information Processing & Management **43**(6) (2007) 1606–1618

[212] Cano Basave, A.E., He, Y., Xu, R.: Automatic labelling of topic models learned from twitter by summarisation. In: Proceedings of The 52nd Annual Meeting of the Association for Computational Linguistics (ACL). (2014)

[213] Nichols, J., Mahmud, J., Drews, C.: Summarizing sporting events using twitter. In: Proceedings of the 2012 ACM international conference on Intelligent User Interfaces, ACM (2012) 189–198

[214] Shou, L., Wang, Z., Chen, K., Chen, G.: Sumblr: continuous summarization of evolving tweet streams. In: Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval, ACM (2013) 533–542

[215] Ren, Z., Liang, S., Meij, E., de Rijke, M.: Personalized time-aware tweets summarization. In: Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval, ACM (2013) 513–522

[216] Duan, Y., Chen, Z., Wei, F., Zhou, M., Shum, H.Y.: Twitter topic summarization by ranking tweets using social influence and content quality. Proceedings of COLING 2012 (2012) 763–780

[217] Liu, X., Li, Y., Wei, F., Zhou, M.: Graph-based multi-tweet summarization using social signals. Proceedings of COLING 2012 (2012) 1699–1714

[218] He, R., Liu, Y., Yu, G., Tang, J., Hu, Q., Dang, J.: Twitter summarization with social-temporal context. World Wide Web **20**(2) (2017) 267–290

[219] Swapna, G., JIANG, J.: Finding thoughtful comments from social media. In: COLING. (2012)

[220] Yang, Z., Cai, K., Tang, J., Zhang, L., Su, Z., Li, J.: Social context summarization. In: Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval, ACM (2011) 255–264

[221] Olariu, A.: Efficient online summarization of microblogging streams. In: EACL. (2014) 236–240

[222] Rudra, K., Ghosh, S., Ganguly, N., Goyal, P., Ghosh, S.: Extracting situational information from microblogs during disaster events: a classification-summarization approach. In: Proceedings of the 24th ACM International on

Conference on Information and Knowledge Management, ACM (2015) 583–592

[223] Nguyen, H.: Research report: 2013 state of social media spam. http://nexgate.com/wp-content/uploads/2013/09/Nexgate-2013-State-of-Social-Media-Spam-Research-Report.pdf (2013)

[224] Aiello, L.M., Deplano, M., Schifanella, R., Ruffo, G.: People are strange when you're a stranger: Impact and influence of bots on social networks. CoRR **abs/1407.8134** (2014)

[225] Kalbitzer, J., Mell, T., Bermpohl, F., Rapp, M.A., Heinz, A.: Twitter psychosis: a rare variation or a distinct syndrome. Journal of Nervous and Mental Disease **202**(8) (August 2014) 623

[226] Tan, C., Lee, L., Pang, B.: The effect of wording on message propagation: Topic- and author-controlled natural experiments on twitter. CoRR **abs/1405.1438** (2014)

[227] Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J., Smith, N.A.: Part-of-speech tagging for twitter: Annotation, features, and experiments. In: Proceedings of ACL. HLT '11, Stroudsburg, PA, USA (2011) 42–47

[228] Nielsen, F.Å.: A new anew: Evaluation of a word list for sentiment analysis in microblogs. In: Workshop on'Making Sense of Microposts: Big things come in small packages. (2011) 93–98

[229] Liu, B.: Sentiment analysis: a multifaceted problem. IEEE Intelligent Systems **25**(3) (2010) 76–80

[230] Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing contextual polarity in phrase-level sentiment analysis. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing. HLT '05, Stroudsburg, PA, USA, Association for Computational Linguistics (2005) 347–354

[231] Pang, B., Lee, L.: Opinion mining and sentiment analysis. Foundations and trends in information retrieval **2**(1-2) (2008) 1–135

[232] Pan, S.J., Yang, Q.: A survey on transfer learning. Knowledge and Data Engineering, IEEE Transactions on **22**(10) (2010) 1345–1359

[233] Xia, R., Zong, C., Hu, X., Cambria, E.: Feature ensemble plus sample selection: domain adaptation for sentiment classification. Intelligent Systems, IEEE **28**(3) (2013) 10–18

[234] Dai, W., Xue, G.R., Yang, Q., Yu, Y.: Co-clustering based classification for out-of-domain documents. In: SIGKDD, ACM (2007) 210–219

[235] Bigi, B.: Using Kullback-Leibler distance for text categorization. Springer (2003)

[236] Hsu, C.W., Lin, C.J.: A comparison of methods for multiclass support vector machines. Neural Networks, IEEE Transactions on **13**(2) (2002) 415–425

[237] Yang, J.: A general framework for classifier adaptation and its applications in multimedia. PhD thesis, Columbia University (2009)

[238] Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology **2** (2011) 27:1–27:27

[239] Pak, A., Paroubek, P.: Twitter as a corpus for sentiment analysis and opinion mining. In: Proceedings of the Seventh International Conference on Language

Resources and Evaluation (LREC'10), Valletta, Malta, European Language Resources Association (ELRA) (may 2010)

[240] Burnap, P., Gibson, R., Sloan, L., Southern, R., Williams, M.: 140 characters to victory?: Using twitter to predict the uk 2015 general election. Electoral Studies **41** (2016) 230–233

[241] Pontiki, M., Papageorgiou, H., Galanis, D., Androutsopoulos, I., Pavlopoulos, J., Manandhar, S.: Semeval-2014 task 4: Aspect based sentiment analysis. In: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014). (2014) 27–35

[242] Klein, D., Manning, C.D.: Accurate unlexicalized parsing. In: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1. ACL '03, Stroudsburg, PA, USA (2003) 423–430

[243] Li, P., Zhu, Q., Zhang, W.: A dependency tree based approach for sentence-level sentiment classification. In: Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2011 12th ACIS International Conference on, IEEE (2011) 166–171

[244] Fouss, F., Yen, L., Pirotte, A., Saerens, M.: An experimental investigation of graph kernels on a collaborative recommendation task. In: Data Mining, 2006. ICDM'06. Sixth International Conference on, IEEE (2006) 863–868

[245] Mihalcea, R.: Graph-based ranking algorithms for sentence extraction, applied to text summarization. In: The Companion Volume to the Proceedings of 42st Annual Meeting of the Association for Computational Linguistics, Barcelona, Spain, Association for Computational Linguistics (July 2004) 170–173

[246] Finkel, J.R., Grenager, T., Manning, C.: Incorporating non-local information into information extraction systems by gibbs sampling. In: Proceedings of

the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05), Ann Arbor, Michigan, Association for Computational Linguistics (June 2005) 363–370

[247] Bird, S.: Nltk: the natural language toolkit. In: Proceedings of the COLING/ACL on Interactive presentation sessions. COLING-ACL '06, Stroudsburg, PA, USA, Association for Computational Linguistics (2006) 69–72

[248] Ritter, A., Clark, S., Mausam, Etzioni, O.: Named entity recognition in tweets: An experimental study. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. EMNLP '11, Stroudsburg, PA, USA, Association for Computational Linguistics (2011) 1524–1534

[249] Landis, J.R., Koch, G.G.: The measurement of observer agreement for categorical data. Biometrics (1977) 159–174

[250] Palogiannidi, E., Kolovou, A., Christopoulou, F., Kokkinos, F., Iosif, E., Malandrakis, N., Papageorgiou, H., Narayanan, S., Potamianos, A.: Tweester at semeval-2016 task 4: Sentiment analysis in twitter using semantic-affective model adaptation. In: SemEval@ NAACL-HLT. (2016) 155–163

[251] Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: Liblinear: A library for large linear classification. Journal of machine learning research **9**(Aug) (2008) 1871–1874

[252] Li, J., Luong, T., Jurafsky, D., Hovy, E.: When are tree structures necessary for deep learning of representations? In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, Association for Computational Linguistics (September 2015) 2304–2314

[253] Newman, N.: The rise of social media and its impact on mainstream journalism. (2009)

[254] Jordaan, M.: Poke me, i'm a journalist: The impact of facebook and twitter on newsroom routines and cultures at two south african weeklies. Ecquid Novi: African Journalism Studies **34**(1) (2013) 21–35

[255] Berkhin, P.: A survey of clustering data mining techniques. In: Grouping multidimensional data. Springer (2006) 25–71

[256] Yin, J., Wang, J.: A text clustering algorithm using an online clustering scheme for initialization. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM (2016) 1995–2004

[257] Aggarwal, C.C., Subbian, K.: Event detection in social streams. In: Proceedings of the 2012 SIAM international conference on data mining, SIAM (2012) 624–635

[258] Hoffman, M., Bach, F.R., Blei, D.M.: Online learning for latent dirichlet allocation. In: advances in neural information processing systems. (2010) 856–864

[259] Lau, J.H., Collier, N., Baldwin, T.: On-line trend analysis with topic models:\# twitter trends detection topic model online. In: COLING. (2012) 1519–1534

[260] Frey, B.J., Dueck, D.: Clustering by passing messages between data points. science **315**(5814) (2007) 972–976

[261] Müllner, D., et al.: fastcluster: Fast hierarchical, agglomerative clustering routines for r and python. Journal of Statistical Software **53**(9) (2013) 1–18

[262] Sokal, R.R., Rohlf, F.J.: The comparison of dendrograms by objective methods. Taxon (1962) 33–40

[263] Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of computational and applied mathematics **20** (1987) 53–65

[264] Arbelaitz, O., Gurrutxaga, I., Muguerza, J., Pérez, J.M., Perona, I.: An extensive comparative study of cluster validity indices. Pattern Recognition **46**(1) (2013) 243–256

[265] Rosenberg, A., Hirschberg, J.: V-measure: A conditional entropy-based external cluster evaluation measure. In: EMNLP-CoNLL. Volume 7. (2007) 410–420

[266] Vinh, N.X., Epps, J., Bailey, J.: Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. Journal of Machine Learning Research **11**(Oct) (2010) 2837–2854

[267] Srijith, P., Hepple, M., Bontcheva, K., Preotiuc-Pietro, D.: Sub-story detection in twitter with hierarchical dirichlet processes. Information Processing & Management **53**(4) (2017) 989–1003

[268] Fang, A., Macdonald, C., Ounis, I., Habel, P., Yang, X.: Exploring time-sensitive variational bayesian inference lda for social media data. In: European Conference on Information Retrieval, Springer (2017) 252–265

[269] Tsakalidis, A., Papadopoulos, S., Cristea, A.I., Kompatsiaris, Y.: Predicting elections for multiple countries using twitter and polls. IEEE Intelligent Systems **30**(2) (2015) 10–17

[270] Hermann, K.M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., Blunsom, P.: Teaching machines to read and comprehend. In: Advances in Neural Information Processing Systems. (2015) 1693–1701

[271] McKeown, K., Passonneau, R.J., Elson, D.K., Nenkova, A., Hirschberg, J.: Do summaries help? In: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, ACM (2005) 210–217

[272] Soderland, J.C.S., Mausam, G.B.: Hierarchical summarization: Scaling up multi-document summarization. In: Proceedings of the 52nd Annual Meeting of the Association for Computlational Linguistics. (2014) 902–912

[273] Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: Text summarization branches out: Proceedings of the ACL-04 workshop. Volume 8., Barcelona, Spain (2004)

[274] Denkowski, M., Lavie, A.: Meteor universal: Language specific translation evaluation for any target language. In: Proceedings of the ninth workshop on statistical machine translation. (2014) 376–380

[275] Nenkova, A., Passonneau, R., McKeown, K.: The pyramid method: Incorporating human content selection variation in summarization evaluation. ACM Transactions on Speech and Language Processing (TSLP) **4**(2) (2007) 4

[276] Louis, A., Nenkova, A.: Automatically assessing machine summary content without a gold standard. Computational Linguistics **39**(2) (2013) 267–300

[277] Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research **12**(Jul) (2011) 2121–2159

[278] Jozefowicz, R., Vinyals, O., Schuster, M., Shazeer, N., Wu, Y.: Exploring the limits of language modeling. arXiv preprint arXiv:1602.02410 (2016)

[279] Mihalcea, R., Tarau, P.: Textrank: Bringing order into text. In: EMNLP. Volume 4. (2004) 404–411

[280] Boudin, F., Mougard, H., Favre, B.: Concept-based summarization using integer linear programming: From concept pruning to multiple optimal solutions. In: Conference on Empirical Methods in Natural Language Processing (EMNLP) 2015. (2015)

[281] Sobhani, P., Inkpen, D., Zhu, X.: A dataset for multi-target stance detection. EACL 2017 (2017) 551

[282] Lau, J.H., Baldwin, T., Cohn, T.: Topically driven neural language model. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL). (2017) 355–365

[283] Cer, D., Yang, Y., Kong, S.y., Hua, N., Limtiaco, N., John, R.S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., et al.: Universal sentence encoder. arXiv preprint arXiv:1803.11175 (2018)

[284] Kusner, M., Sun, Y., Kolkin, N., Weinberger, K.: From word embeddings to document distances. In: International Conference on Machine Learning. (2015) 957–966

[285] Xu, W., Callison-Burch, C., Dolan, W.B.: Semeval-2015 task 1: Paraphrase and semantic similarity in twitter (pit). Proceedings of SemEval (2015)

[286] Hu, B., Chen, Q., Zhu, F.: Lcsts: A large scale chinese short text summarization dataset. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP). (2015)

[287] Ng, J.P., Abrecht, V.: Better summarization evaluation with word embeddings for rouge. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP). (2015)

[288] Conover, M.D., Gonçalves, B., Ratkiewicz, J., Flammini, A., Menczer, F.: Predicting the political alignment of twitter users. In: Privacy, Security, Risk

and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on, IEEE (2011) 192–199

[289] Agrawal, R., Srikant, R., et al.: Fast algorithms for mining association rules. In: Proc. 20th int. conf. very large data bases, VLDB. Volume 1215. (1994) 487–499