JAMES COOK UNIVERSITY

DOCTORAL THESIS

Document-level Sentiment Analysis of Email Data

Author: Sisi LIU

A thesis submitted in fulfilment of the requirements
for the degree of Doctor of Philosophy

in the

College of Science and Engineering

July 20, 2020

Declaration of Authorship

I, Sisi LIU, hereby declare that I have full ownership of this thesis titled, "Document-level Sentiment Analysis of Email Data" and the content written in it. I confirm that:

- This thesis has not been submitted in any form for another degree or diploma at any university or other institution of tertiary education.
- Information derived from the published or unpublished work of others has been acknowledged in the text and a list of references is given.
- This thesis contains no material which infringes the copyright of any other person.
- Any permission statements from copyright owners are in an appendix to both the print and electronic copies of the thesis.

Signed:

____

Date:

____

Acknowledgements

This thesis represents a long, yet enjoyable journey of four years' research through which I could never have persisted without the guidance and support of my supervisors, family and friends. Herein, I would first like to express my sincere gratitude and affection to my primary supervisor Professor Ickjai (Jai) Lee, who has been a great mentor in my research candidature and inspired me in the pursuit of a professional career. As an international student with a non-IT background and limited research experience, it was Jai who offered me enormous help in settling into a new environment and building solid research and technical skills. I would also like to extend my gratitude to my secondary supervisors, Associate Professor Laurie Murphy and Dr Kyungmi (Joanne) Lee, for their special insights and domain-related knowledge, which assisted me in consummating my research and this thesis.

Moreover, I would like to give special thanks to James Cook University for providing financial support and offering a leading research and academic environment. I am also grateful for all the fellow researchers and friends I met at the university who devoted their time and kindness to me.

Last, but not least, I would like to dedicate my deepest thanks to my parents, Zhihong Wang and Feng Liu, for their everlasting love and support throughout my studies. It was their continuous encouragement and understanding that helped get me through every difficult time and reach my goals. I would also like to give special thanks to my long-time friends and close relatives in my home country for their psychological and emotional support.

Statement of the Contribution of Others

Nature of Assistance	Contribution	Names
Supervision support	Primary supervision Secondary supervision Secondary supervision	Professor Ickjai Lee Associate Professor Laurie Murphy Dr Kyungmi Lee
Research support	Conceptual guidance, analytical support, paper/thesis revision, editing	Professor Ickjai Lee
Proofreading support	Grammar and spelling checks, syntactic structure and language usage corrections	Professor Ickjai Lee Dr John Gibbens (paid external proofreader)
Financial support	International Research Training Program Scholarship (IRTPS)	James Cook University

Abstract

Sisi LIU

Document-level Sentiment Analysis of Email Data

With the increasing prevalence of electronic devices and advances in network technology, large volumes of textual data are being produced during the daily operations of various online media platforms. Sentiment analysis is a field of text mining that aims to automatically identify the sentiments or opinions contained in a piece of text. Through the implementation of statistical models and machine learning algorithms, sentiment analysis identifies and quantifies opinionated patterns extracted from subjective expressions in massive text datasets to support decision-making processes.

Despite the fact that Email is a widely adopted contemporary means of communication in business settings, Email sentiment analysis is a field that has not been studied thoroughly. Document-level sentiment analysis is the basic form and is crucial, as it can extract opinions or sentiments from an entire document. As Emails are organised by subject lines and threads, studying each Email message as a whole piece of textual data aids in better understanding of how Emails are written and communicated. Hence, it is reasonable to undertake document-level sentiment analysis for Email data that delivers more meaningful insights. Nevertheless, Email has several unique features that are influential to sentiment classification performance, including noisy and unstructured content, sentiment sequences and multiple topics. To develop a model suitable for Email document sentiment analysis, these features must be taken into consideration.

This thesis designs and develops a systematic framework for document-level sentiment analysis of Email data. To effectively analyse and classify the sentiments contained in Email data, a framework is explored that has four major phases: 1) preprocessing, 2) feature generation, 3) document vectorisation and 4) sentiment analysis. The study aims to test the hypothesis that algorithms that incorporate sentiment sequences and multi-topic features outperform conventional methods of Email sentiment classification. To achieve this, three sub-studies were conducted, focusing on 1) sentiment sequence clustering, 2) sequence-encoded neural sentiment classification and 3) multi-topic neural sentiment classification. In brief, a novel method of sequence-based document sentiment analysis is introduced for discovering sentiment sequences contained in Email data and clustering the sentiments. Once the presence of sentiment sequences within Email documents is confirmed, a robust sequence-encoded neural network model with a dependency graph-based position-encoding technique enhanced with weighted sentiment features is proposed to further utilise sentiment sequences for sentiment classification. And finally, a neural network model with topic embeddings and topic weighting vectors is designed and developed to better model Email documents and capture complex sentiment structures within them.

In addition to sentiment sequences and multi-topic features, which are investigated in the three main studies, the proposed framework is further evaluated by implementing a preprocessing phase that handles noise and data scarcity issues in Email data. Experiments comparing analytical performance using raw and cleaned datasets, and using original and augmented datasets, demonstrate the effectiveness of the preprocessing phase, which comprises Email cleaning, text normalisation and data augmentation.

Overall, a comprehensive and systematic framework for document-level Email sentiment analysis is developed through the exploration of sentiment sequence clustering, sequence-encoded neural sentiment classification and multi-topic neural sentiment classification. The methods described in this thesis will aid in more accurately and efficiently determining the sentiments contained in massive amounts of Email data. With the assistance of the analytical results obtained from the framework, document-level Email sentiment analysis will contribute to the better understanding of Email communication and utilisation of Emails as a tool for insightful decision making.

Contents

List of Figures

List of Tables

List of Abbreviations

BoWs	Bag-of-Words
BiLSTM	Bidirectional Long Short-term Memory
CNN	Convolutional Neural Network
CRF	Conditional Random Field
CRM	Customer Relationship Management
CTSS	Compatible Time-sharing Systems
DBSCAN	Density-based Spatial Clustering of Applications with Noise
DG	Dependency Graph
GRU	Gated Recurrent Unit
HTML	Hypertext Mark-up Language
k-NN	k-Nearest-Neighbor
LDA	Latent Dirichlet Allocation
LR	Logistic Regression
LSA	Latent Semantic Analysis
LSTM	Long Short-term Memory
MAE	Mean Absolute Error
Macro-F	Macro F-measure
MLP	Multi-Layer Perceptron
MT-BiLSTM	Multi-Topic Bidirectional Long Short-term Memory
NB	Naïve Bayes
NLP	Natural Language Processing
NMF	Non-negative Matrix Factorization
POS	Part-of-Speech
PT	Plain Text
RBFN	Radial Basis Function Neural
ReLU	Rectified Linear Unit
RF	Random Forest
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
SGD	Stochastic Gradient Descent
SMS	Short Message Service
SSE	Squared Sum Error
SVM	Support Vector Machine
SWN	SentiWordNet
TF-IDF	Term Frequency-inverse Document Frequency
TRACLUS	TRAjectory CLUStering
URL	Uniform Resource Locator
WCSS	Within Cluster Sum of Squared
WEKA	Waikato Environment for Knowledge Analysis
WN	WordNet

1 Introduction

In this chapter, I introduce the highlights of this thesis, which describes a thorough study of *document-level Email sentiment analysis*. The study aimed to effectively analyse sentiment sequences and classify sentiment polarity in Email data. In Section 1.2, the research motivations are elaborated in accordance with a brief review of the background knowledge and theoretical foundations of sentiment analysis. Sections 1.3 and 1.4 describe the overall research framework and problem, which are used to derive the research aims and questions. Finally, Section 1.5 outlines the main contents of the remaining thesis chapters.

1.1 Background

> *"It is rare that the public sentiment decides immorally or unwisely, and the individual who differs from it ought to distrust and examine well his own opinion."*
>
> — Thomas Jefferson, *Letter to William Findley*

With the diffusion of social networks and Web 2.0 technology, increasingly massive volumes of user-generated content are being produced from various sources; for example, communication services such as Short Message Service(SMS) and Email, social media platforms such as Facebook[1] and Twitter[2], and websites such as IMDB[3] and TripAdvisor[4]. While social media platforms are more popular in casual communications, "Email continues to be an essential part of daily business and consumer communication (p. 2)." due to its advantages of being cost-efficient

[1] https://www.facebook.com/

[2] https://twitter.com/

[3] https://www.imdb.com/

[4] https://www.tripadvisor.com/

and highly compatible, as stated by the *Email Statistics Report, 2019-2023*[5]. Table 1.1 and Table 1.2 present some statistics that highlight current and projected rates of Email use.

TABLE 1.1: Worldwide daily Email t raffic (B), 2019-2023

Daily Email Traffic	**2019**	**2020**	**2021**	**2022**	**2023**
Total Worldwide Emails Sent/Received Per Day (B)	293.6	306.4	319.6	333.2	347.3
% Growth	-	4.4%	4.3%	4.3%	4.3%

TABLE 1.2: Worldwide Email user forecast (M), 2019-2023

	2019	**2020**	**2021**	**2022**	**2023**
Worldwide Email Users (M)	3,930	4,037	4,147	4,258	4,371
% Growth	-	3%	3%	3%	3%

To avoid the issue of data overloading and extract useful information and patterns from large amounts of textual data, it is essential to develop algorithms that perform such extraction automatically. This process forms a burgeoning field of study known as text mining and Natural Language Processing(NLP). Sentiment analysis is one of the most attractive areas of text mining, which analyses opinionated textual information to inform decision-making processes. The term *sentiment*, as in sentiment analysis, as defined in the *Cambridge Dictionary*, is either "a) IDEA: a thought, opinion, or idea based on a feeling about a situation, or a way of thinking about something" or "b) FEELINGS: gentle feelings such as sympathy, love, etc., especially when considered to be silly or not suitable." Sentiment analysis can be conducted at three broad levels according to whether the sentiments are associated with a document, sentence or aspect. This thesis focuses on developing a framework for document-level Email sentiment analysis that mainly examines sentiments associated with an entire Email document (according to the first definition of sentiment: ′thought, opinion, or idea′).

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

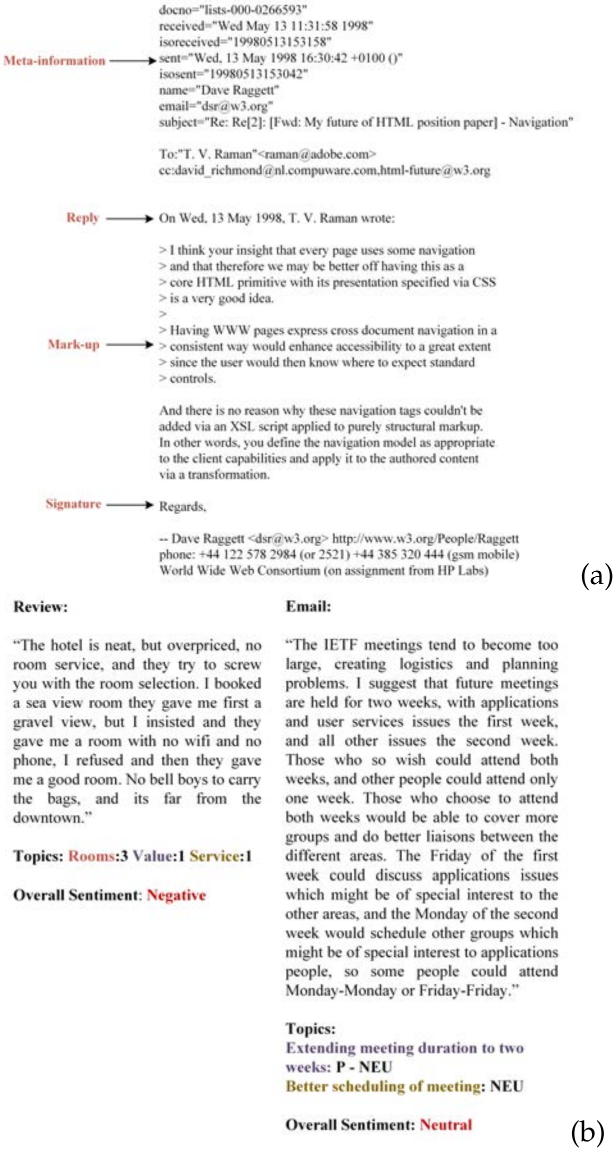
Sentiment analysis has been successfully applied to a wide range of industrial practices, especially to fields like business decision-making (Tang et al., 2009; Wu et al., 2014; Wu et al., 2010), Customer Relationship Management(CRM) (Oelke et al., 2009), emergency response (Caragea et al., 2014), risk management (Coletto et al., 2016), political campaigns (Wanner et al., 2009) and social psychology (Pestian et al., 2012; Yu et al., 2013a). Techniques for sentiment analysis have advanced and matured over the past two decades. Most sentiment analysis studies are based on data sources involving social media posts (Caragea et al., 2014; Coletto et al., 2016; Wu et al., 2014), reviews (Oelke et al., 2009; Tang et al., 2009; Wu et al., 2010), and news feeds (Wanner et al., 2009). Nevertheless, Email data is not a common target of sentiment analysis and effective techniques for Email sentiment analysis are yet to be explored.

1.2 Motivations

According to the statistics in Table 1.1, 293.6 billion Emails were sent or received worldwide per day in 2019, and this number is predicted to increase by 4.3% annually. Hence, there is an urgent need for Email data mining, mainly due to the extreme information overload issues associated with the large volumes of Email data generated by various applications, such as business operations, personal communications and commercial activities.

The study of Email mining involves many tasks and applications, of which summarisation and visualisation (Dredze et al., 2008; Li et al., 2004a), spam detection (Blanzieri and Bryl, 2008) and thread identification (Sharaff and Nagwani, 2016) have attracted attention from the research community. Hangal et al. (2011) proposed an interactive visual analytic system (abbreviated as MUSE) for Email archiving and stated that "while sentiments are among the noisiest cues provided by MUSE, they are also often the most engaging (p. 6)." With the wide application of Email communication to various personal, social and commercial activities, sentiment analysis of Email data is beneficial to several real-life practices. On the one hand, the implementation of sentiment analysis in a personal Email system contributes to work prioritisation by orgainsing the importance of Emails based on their levels of subjectivity. On the other hand, the implementation of sentiment analysis in a business Email system assists in customer relationship management by dealing with customer complaints through emphasising on Emails with negative sentiments.

Email sentiment analysis is an intriguing yet still-developing area of study owing to the distinctive features of Email data. These features differ from those of other common data sources used in sentiment analysis, which intensifies the difficulty of directly applying existing techniques to Email sentiment analysis.


Meta-information
Reply
Mark-up
Signature
(a)
(b)

```
docno="lists-000-0266593"
received="Wed May 13 11:31:58 1998"
isoreceived="19980513153158"
sent="Wed, 13 May 1998 16:30:42 +0100 ()"
isosent="19980513153042"
name="Dave Raggett"
email="dsr@w3.org"
subject="Re: Re[2]: [Fwd: My future of HTML position paper] - Navigation"

To:"T. V. Raman"<raman@adobe.com>
cc:david_richmond@nl.compuware.com,html-future@w3.org

On Wed, 13 May 1998, T. V. Raman wrote:

> I think your insight that every page uses some navigation
> and that therefore we may be better off having this as a
> core HTML primitive with its presentation specified via CSS
> is a very good idea.
>
> Having WWW pages express cross document navigation in a
> consistent way would enhance accessibility to a great extent
> since the user would then know where to expect standard
> controls.

And there is no reason why these navigation tags couldn't be
added via an XSL script applied to purely structural markup.
In other words, you define the navigation model as appropriate
to the client capabilities and apply it to the authored content
via a transformation.

Regards,

-- Dave Raggett <dsr@w3.org> http://www.w3.org/People/Raggett
phone: +44 122 578 2984 (or 2521) +44 385 320 444 (gsm mobile)
World Wide Web Consortium (on assignment from HP Labs)
```

Review:

"The hotel is neat, but overpriced, no room service, and they try to screw you with the room selection. I booked a sea view room they gave me first a gravel view, but I insisted and they gave me a room with no wifi and no phone, I refused and then they gave me a good room. No bell boys to carry the bags, and its far from the downtown."

Topics: Rooms:3 Value:1 Service:1

Overall Sentiment: Negative

Email:

"The IETF meetings tend to become too large, creating logistics and planning problems. I suggest that future meetings are held for two weeks, with applications and user services issues the first week, and all other issues the second week. Those who so wish could attend both weeks, and other people could attend only one week. Those who choose to attend both weeks would be able to cover more groups and do better liaisons between the different areas. The Friday of the first week could discuss applications issues which might be of special interest to the other areas, and the Monday of the second week would schedule other groups which might be of special interest to applications people, so some people could attend Monday-Monday or Friday-Friday."

Topics:
Extending meeting duration to two weeks: P - NEU
Better scheduling of meeting: NEU

Overall Sentiment: Neutral

FIGURE 1.1: Email data with three distinctive sentiment-relevant features: a) sample raw Email with noise and unstructured content and b) sample labelled Email with sentiment sequence and multi-topic features compared with a sample labelled review.

As implied by the two sample Email datasets shown in Figure 1.1, three distinctive features can be identified that set the challenges for document-level Email sentiment analysis.

Noise and unstructured content. The complete structure of a piece of raw Email data may be composed of two parts: a header and a body (Tang et al., 2014). The Email fragment shown in Figure 1.1 (a) contains quite a few lines of meta-information (e.g., ′subject′, ′To′ or ′cc′) in the header part, and unstructured and noise content (e.g., reply lines, mark-ups or signature blocks) in the body part. As obvious features observed directly in Email data, the issues of noise and unstructured content have been addressed in many existing studies on Email mining. More details regarding these features will be discussed in Section 2.3.1.

Sentiment sequence feature. The sample Email presented in the second example (Figure 1.1 b) is annotated with three sets of labels: topic labels, polarity labels associated with topics and an overall sentiment label. To have a better understanding of the concept of sentiment sequence features, all polarity labels are to be highlighted. It can be observed from the four polarity labels that this piece of Email data embeds a sentiment flow of $positive(P) \rightarrow neutral(NEU) \rightarrow neutral(NEU)$ within the content (without consideration of topics at this point) and is finally classified as *neutral*. Sentiment sequence features appear in Email data mainly due to their lengthy and complex relational and syntactical structures. As such features have not been comprehensively explored in any existing research, they comprise a relatively novel and unique focus in this study.

Multi-topic feature. A sample review with the same sets of labels (as described in the previous paragraph) is also presented in the second example (Figure 1.1 b) to illustrate multi-topic features in Email data. In comparison to the topics annotated to the Email message, which are closely connected and represented by short phrases, the topics in the review exhibit clear boundaries in meanings, as each topic

(’room’, ’value’ or ’service’) is relevant to the domain yet is an independent term describing an aspect. Some techniques developed for aspect-level sentiment analysis are utilised to treat data with topics in single terms or as a list of seed words (Poria et al., 2016; Ruder et al., 2016). It is observed that multi-topic features in Email data have fewer boundaries in meanings and more concrete descriptions than those found in other types of textual data. To be more specific, in the example, topics annotated to the Email message are short phases compared to single topic terms annotated to the review and the keyword ’meeting’ exists in both Email topics. Therefore, existing techniques are inadequate for performing sentiment analysis on Emails.

To sum up, the above three features increase the difficulty of Email sentiment analysis due to the lack of labelled Email data that is readily available for comprehensive quantitative evaluation. This forms another core challenge of the task. Motivated by the fact that existing techniques are insufficient for handling all four difficulties, this study aimed to develop improved techniques that effectively analyse and classify sentiments from Email data by considering the four factors of 1) noise and unstructured content, 2) sentiment sequence features, 3) multi-topic features and 4) a lack of labelled data.

1.3 Research problem

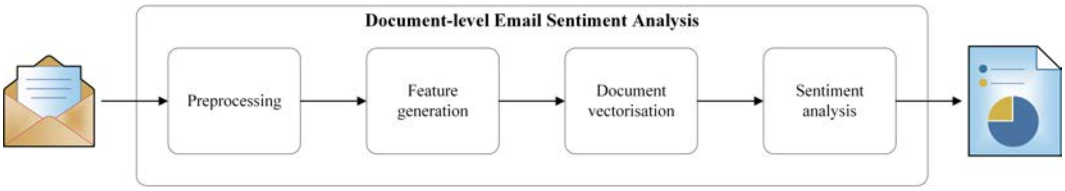


FIGURE 1.2: An overview of the document-level Email sentiment analysis framework.

Inspired by the challenges discussed in the previous section, the overall research problem in this study is to design and develop a systematic and comprehensive framework for document-level Email sentiment analysis. The aim is to effectively analyse and classify sentiments from Email data according to the framework shown in Figure 1.2. The framework consists of four major phases—preprocessing, feature generation, document vectorisation and sentiment analysis—and contains four main functions:

- Noise handling
- Sentiment sequence
- Sentiment classification
- Quantitative evaluation

The above four functions are associated with unique features identified in Email data and the general framework formulated for document-level Email sentiment analysis. In brief, a noise handling function is implemented in the preprocessing phase that aims to solve the issue of noise and unstructured content through proper Email cleaning and text normalisation methods. Sentiment sequence features and multi-topic features are addressed in the feature generation phase as part of the sentiment sequencing and sentiment classification functions. A quantitative evaluation function is implemented in the sentiment analysis phase that aims to obtain reliable classification results from an adequate amount of data through appropriate data augmentation methods.

1.4 Research aims and questions

To break down the aforementioned framework into more specific tasks, four research aims are defined according to the main components of the framework:

- **Preprocessing**: To investigate preprocessing methods that reduce the impact of unstructured and noisy data, and data scarcity.
- **Feature generation**: To investigate the effectiveness of sentiment sequence and multi-topic features on Email sentiment determination and effective feature generation methods.
- **Document vectorisation**: To investigate document vectorisation methods that capture sentiment sequence and multi-topic features that can be used to effectively model Email documents and represent them as numeric vectors.

- **Sentiment analysis**: To investigate effective sentiment sequence discovery and sentiment classification methods.

The high-level research question derived from the main research problem is formulated as: how to incorporate the special characteristics of Email, including noise, sentiment sequence and multi-topic, into the sentiment analysis process and build a robust and effective framework for Email sentiment classification? Several sub-questions are identified that should lead to concrete technical approaches to achieving each aim:

1. What preprocessing methods are essential in addressing unstructured and noisy contents in Email data and can solve the issues of data scarcity and imbalanced class distributions in labelled Emails?
2. How to effectively capture sentiment sequence features and discover sentiment sequence patterns within Email data?
3. How to encode sentiment sequence features in a neural network model for robust and accurate sentiment polarity classification?
4. How to capture multi-topic features and model documents with multi-topic segments for effective sentiment polarity classification?

Briefly, Research Question 2 is addressed through a study on sentiment sequence clustering, with a more detailed discussion given in Chapter 4. Research Question 3 is addressed through a study on sequence-encoded neural sentiment classification, with a more detailed discussion provided in Chapter 5. Research Question 4 is addressed by a study on multi-topic neural sentiment classification (Chapter 6). Research Question 1 is addressed by conducting experiments that compare the preprocessed and original data obtained in the second and third studies (Chapter 5 & 6). Research hypotheses associated with the research aims and questions are discussed in Section 2.5 following a thorough review of the literature and a summary of existing research gaps.

1.5 Thesis significance

The main significance of the research is the design and development of a systematic and comprehensive framework for document-level sentiment analysis of Email data. The framework fulfills four tasks, including noise handling, sentiment sequence discovery, sentiment polarity classification and quantitative evaluation, through three studies on 1) sentiment sequence clustering, 2) sequence-encoded neural sentiment classification and 3) multi-topic neural sentiment classification. an investigation on the . This research further contributes to the literature of Email sentiment analysis by investigating the effectiveness of Email data preprocessing and augmentation methods on solving the issues of data scarcity and imbalanced class distributions.

The following list presents the main contributions and significance of the research summarised by the main thesis chapters.

- **Chapter 3.** Email data preprocessing and augmentation. This research is a novel investigation into Email sentiment analysis with benchmarking datasets and results. A thorough set of preprocessing and data augmentation methods is introduced to address the issues of unstructured and noisy contents in Email data, data scarcity and imbalanced class distributions in labelled Emails. The proposed methods are effective in reducing the negative influence of the above mentioned issues on the classification performance.

- **Chapter 4.** Sentiment sequence clustering. A three-phase trajectory representation approach is designed to model Email documents into sentiment sequence representations. The proposed method proves the existence of sentiment sequence within Email documents and explores the possibility of implementing sentiment sequence features into the process of sentiment classification with improved performance.

- **Chapter 5.** Sequence-encoded neural sentiment classification. A position encoding method with dependency graph-based position features encoded by discourse depth weighting is developed to capture sentiment sequence features from Email documents and incorporate them into a neural model for sentiment classification. The proposed method properly models the sentiment sequence features in Emails and obtains more robust and accurate classification results compared to other baseline methods.

- **Chapter 6.** Multi-topic neural sentiment classification. A document segmentation method based on topic modelling and semantic text segmentation is explored to capture multi-topic features from Email documents and incorporate them into a neural model for sentiment classification. The proposed method manages to effectively detects multi-topic features in Emails and achieves improved sentiment classification results compared to other state-of-the-art algorithms.

1.6 Thesis outline

Overall, seven chapters, including this introductory chapter, are involved in the thesis. A brief outline of each of the following chapters is provided below.

- **Chapter 2** critically reviews the literature on sentiment analysis and Email mining to summarise the existing research gaps and inform our hypotheses used in the study of document-level sentiment analysis of Email data. A broad analysis of sentiment analysis was undertaken by reviewing studies grouped according to tasks and granularities. An in-depth analysis of document-level sentiment classification was then undertaken, which gained insights into the features and techniques involved in relevant tasks. Studies related to Email mining were then reviewed, gaining insights into the characteristics of Email data and the techniques implemented in sentiment- and non-sentiment-related tasks.
- **Chapter 3** describes the overall structure of the Email document sentiment analysis method, with detailed coverage of the four major phases of preprocessing, feature generation, document vectorization, and sentiment analysis. An elaboration is provided for the data collection and label

conversion process based on the use of three benchmark Email datasets. The components of the preprocessing phase, which involve data augmentation, Email cleaning and text normalisation, are described in this chapter. Technical details to address Research Question 1 are mainly covered in this chapter.

- **Chapter 4** presents the study of an unsupervised sequence-based clustering approach to the identification and visualisation of sentiment sequence patterns within Email data. A revised TRAjectory CLUStering(TRACLUS) algorithm is implemented with documents represented by sentiment trajectories to perform sentiment sequence clustering. A three-stage trajectory representation approach composed of sentiment feature generation, pseudo-longitude and –latitude transformation, and pixel conversion is developed to transform Email documents into sentiment trajectories. Technical details and empirical results for Research Question 2 are covered in this chapter.

- **Chapter 5** describes the study of a sequence-encoded Convolutional Neural Network(CNN) model for Email document sentiment classification. A dependency graph and discourse weighting method is proposed to capture position features. And then a sentiment sequence encoding method using an Long Short-term Memory(LSTM) model of combined sentiment lexical features and position features is developed to vectorise documents as inputs for a revised CNN model for classification. Technical details for Research Question 3 and empirical results for Research Question 1 & 3 are covered in this chapter.

- **Chapter 6** describes the study on document-level multi-topic sentiment classification for Email data using a topic-weighted Bidirectional Long Short-term Memory(BiLSTM) model. An improved semantic text segmentation method with Latent Dirichlet Allocation(LDA) topic modelling is adopted to model documents into multiple topic segments. A multi-topic BiLSTM model is built on the original BiLSTM with additional layers of topic embeddings and topic weighting vectors. Technical details for Research

Question 4 and empirical results for Research Question 1 & 4 are covered in this chapter.

- **Chapter 7** concludes the thesis by highlighting the main contributions and key findings of each chapter. Finally, some limitations of the thesis and further research directions are discussed.

Table 1.3 presents a summary of the intellectual contributions and publications associated with this thesis.

TABLE 1.3: Summary of intellectual contributions of publications involved in the thesis.

Chapter	Publication	Intellectual Contribution
3	Content derived from papers listed in Chapter 5 and 6.	Liu collected data and coded the algorithms. Lee advised on the design of the overall structure and label conversion approaches.
4	[**Published**]Liu, S., & Lee, I. (2018). Discovering sentiment sequence within email data through trajectory representation. *Expert Systems with Applications*, 99, 1-11.	Liu developed the research problems, coded the algorithms and performed evaluations. Lee guided the paper structure and definitions. Liu drafted the paper and Lee provided revision and editorial support.
5	[**Submitted**]Liu, S., & Lee, I. Sequence encoding incorporated CNN model for Email document sentiment classification. *Applied Soft Computing*.	Liu developed the proposed approach, coded the algorithms and performed the major evaluations. Lee refined the experiments with a model stopping criterion and statistical testing. Liu drafted the paper and Lee provided revision and editorial support.
6	[**Accepted with a minor revision**]Liu, S., Lee, K., & Lee, I. Document-level multi-topic sentiment classification of Email data with BiLSTM and data augmentation. *Knowledge-based Systems.*	Liu developed the proposed approach, coded the algorithms and performed major evaluations. Lee refined the experiments with a model stopping criterion and statistical testing. Liu drafted the paper and Lee provided revision and editorial support.

2 Literature review

In this chapter, I present a comprehensive literature review that covers the theoretical background and technical foundations of sentiment analysis and Email mining. Representative publications relevant to these two areas are critically analysed and summarised. Section 2.1 provides a broad overview of sentiment analysis, with a general overview and categorisation of its tasks and granularities. Section 2.2 provides an in-depth analysis of document-level sentiment classification involving different types of features and techniques. Section 2.3 reviews the tasks relevant to Email mining in detail and summarises the major concepts related to the characteristics of Email data and the techniques used in these studies. In Section 2.4, I identify research gaps related to document-level sentiment analysis and Email sentiment analysis to justify the necessity of my research. Finally, I describe several research hypotheses that are associated with the research aims and objectives and the research gaps identified.

2.1 An overview of sentiment analysis

Sentiment analysis is a specialized area of study that provides insight into textual information. It is also an indispensable part of NLP as, presumptively, any textual information expresses either facts or opinions (Liu, 2012). The review of the literature covers a broad range of definitions related to sentiment analysis. For example, "sentiment analysis is the field of study that analyses people's opinions, sentiments, evaluations, appraisals, attitudes, and emotions towards entities such as products, services, organizations, individuals, issues, events, topics, and their attributes" (Liu, 2012, p. 7), which is a relatively well-adopted definition in this field of research.

The history of sentiment analysis can be traced back to the late-20th to the early-21st century, when the term *sentiment* appeared in published articles in reference to predictive judgments of text for the purposes of financial market analysis (Barberis et al., 1998; Das and Chen, 2001). Later in 2002, Turney (2002) explored the possibility of applying the semantic orientation of adjectives to the classification of the overall opinion of a document. This marked the beginning of research into the sentiments or opinions contained in textual information, which became a significant focus of NLP. The term *sentiment analysis* allegedly first appeared as a key concept in a paper entitled "Sentiment analysis: Capturing favorability using natural language processing" written by Nasukawa and Yi (2003). In the same year, Yi et al. (2003) published the paper "Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques" on the topic of sentiment analysis.

Broadly, sentiment analysis studies can be functionally categorised into various task orientations, such as subjectivity classification (Maas et al., 2011; Wilson et al., 2005) and emotion recognition (Alm et al., 2005; Kumar and Minz, 2013). Nevertheless, similar problem-solving structures and knowledge-discovery processes were observed in these studies. *Sentiment analysis* or *opinion mining* is, hence, more frequently employed in review articles in reference to high-level summarization of concepts and techniques. Though it is common in academia to use *sentiment analysis* and *opinion mining* interchangeably (Liu, 2012), I use the term *sentiment analysis* throughout the thesis in reference to my main research objective.

To structurally define the problem of sentiment analysis, it is necessary to understand the relationship between *opinion* and *sentiment*. Kim and Hovy (2004) proposed a quadruple of ($topic$, $holder$, $claim$, $sentiment$) to represent an opinion. Years later, Liu et al. (2010) formally defined an *opinion* as any "subjective expression" that forms a quintuple of ($entity$, $aspect$, $sentiment$, $holder$, $time$), with sentiments being one kind of attribute of the expression. To be more specific, the former associates opinions with claims, holders, topics and sentiments, while the latter integrates opinions with more specific targets, including entities, aspects and times, apart from expressions, holders and sentiments.

However, considering practical needs and computational complexity, many existing studies approach sentiment analysis as a classification problem that aims to determine *sentiment polarity* (defined by Liu et al., 2010, as the orientation of a sentiment from an opinionated piece of text containing claims or expressions; Kumar and Minz, 2013; Kundi et al., 2014; Maas et al., 2011; Majumder et al., 2017; Nakagawa et al., 2010). From this perspective, sentiment analysis studies can be further categorized as document-level, sentence-level or aspect-level based on the granularity of the pieces of text. Additionally, as a classification task, sentiment analysis can either be binary or multi-class depending on the number of polarity labels. Binary polarity (containing positive and negative) is the most common way of classifying a text document; whereas in real-life situations, sentiments are more complex and diversified. Therefore, in the review summary, I use *multi-class* as a representative term for tasks other than binary polarity ones, such as fine-grained (5-classes) or multi-scaled ratings.

A review of the history of sentiment analysis indicates that the year 2001 was when awareness of the indispensability of sentiment analysis increased. The number of opinion-mining-related studies also increased, as did the prevalence of Web 2.0 applications, which allow more user-generated content to be made available to the public (Pang, Lee, et al., 2008). Subsequently, a large number of studies have been undertaken and a remarkable number of papers have been published over the past two decades. Techniques developed for solving sentiment analysis problems have evolved significantly with the advancement of machine learning and parallel processing. Once every few years, a review paper on sentiment analysis is published that evaluates the proposed techniques and progress made during that period of time. The articles reviewed in this section were referenced by such review papers and are grouped according to the tasks and granularities defined in Figure 2.1.

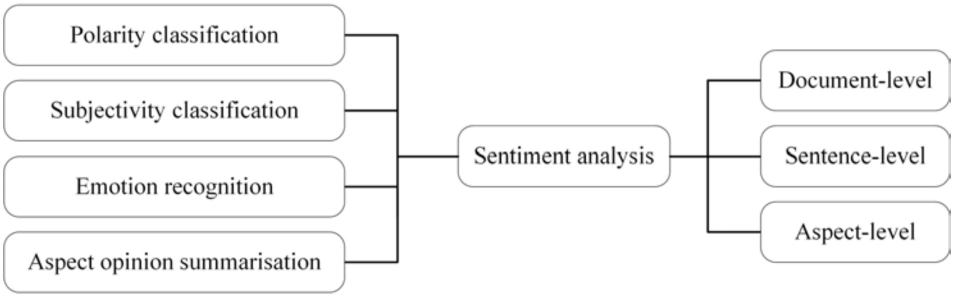


FIGURE 2.1: Sentiment analysis grouped in different tasks and granularities.

2.1.1 Tasks in sentiment analysis

Considering the popularity of tasks and technical relevance to my research, I review studies on sentiment analysis that can be sorted into the following four tasks: 1) polarity classification, 2) subjectivity classification, 3) emotion recognition and 3) aspect opinion summarisation.

2.1.1.1 Polarity classification

Many early studies on sentiment analysis focused on *polarity classification*, a task that aims to classify an opinionated piece of textual data (e.g., a document, sentence or aspect) into one of a set of polarity labels, either binary (e.g., positive or negative) or multi-class (e.g., fine-grained or scaled ratings; Dave et al., 2003; Nasukawa and Yi, 2003; Pang et al., 2002; Turney, 2002). As a primary and fundamental task in sentiment analysis, polarity classification has continuously attracted plenty of interest from the research community over recent years.

Polarity classification techniques are broadly categorised into lexicon-based and machine learning-based approaches. Pure lexicon-based approaches were dominant in early studies when the main focus was to simply identify sentiment orientation from words and phrases, whereas more recent studies tend to utilise pre-developed sentiment lexicons as features rather than as the sole determinant of the sentiment polarity of a target (Kundi et al., 2014; Rao et al., 2018). Compared to lexicon-based approaches that involve excessive human effort during the lexicon generation process, machine learning approaches are increasingly adopted by researchers due to their automated implementation and efficiency in detecting sentiment polarities (Bespalov et al., 2012; Li et al., 2015; Matsumoto et al., 2005; Onan et al., 2016; Pang et al., 2002).

With the continuous improvements in computational speed and power, more recent studies have shifted focus from supervised learning approaches to deep neural network models (Chen et al., 2016; Rao et al., 2018; Tang, 2015; Yang et al., 2016b). A more detailed review of techniques for document-level sentiment polarity classification is given in Section 2.2.3.

2.1.1.2 Subjectivity classification

As distinct from polarity classification, *subjectivity classification* deals with judging whether a piece of textual data (e.g., a document, sentence or phrase) is opinionated or not. The term *subjective* was raised by Wiebe et al. (1999) in a paper entitled "Development and use of a gold-standard data set for subjectivity classifications". Unlike *objective*, *subjective* is determined by whether the primary intention of a sentence is to be factual or not.

Liu et al. (2010) defined subjectivity classification as a task of determining whether a sentence is subjective or objective. In other words, subjectivity classification is, to some extent, equivalent to a sentence-level binary polarity classification task. For instance, Maas et al. (2011) utilised an unsupervised probabilistic model with a supervised sentiment component computed by a logistic regression predictor to perform sentence-level subjectivity detection for movie review data. Nakagawa et al. (2010) implemented an unsupervised probabilistic model with Conditional Random Fields(CRFs) and hidden variables for classifying subjectivity at the phrase- and sentence-levels and for detecting polarity reversals in data from different domains. These studies observed that subjectivity classification is more commonly regarded as a prerequisite or a filtering phase for further sentiment analysis rather than as an independent task.

2.1.1.3 Emotion recognition

An extension of sentiment polarity classification that analyses more fine-grained emotional states is known as *emotion recognition*. In real-life applications, the sentiments in some types of opinionated textual data may not well fit into dichotomous categories (e.g., positive or negative). Instead, some studies utilise more human-like sets of affective or emotive labels for analysis, such as the six emotional states of Eckman (1972): anger, fear, disgust, surprise, sadness and happiness; or the "Big Five" personality traits of openness, conscientiousness, extraversion, agreeableness and neuroticism (David and Suls, 1999). Such studies involve detecting moods in lyrics (Kumar and Minz, 2013), analysing mental state in diaries (Tai et al., 2015) or detecting personalities in essays (Majumder et al., 2017). Most of these studies consider linguistic and psychological factors in the analytical process and make contributions that are more applicable than technical.

2.1.1.4 Aspect opinion summarisation

The task of *aspect opinion summarisation* focuses on the summarisation of sentiments associated with aspects of a set of opinionated textual data. An aspect is defined as a feature or topic embedded in an opinionated document (Liu et al., 2010). It is more commonly observed in reviews or comments that have multiple aspects, with each associated with a different sentiment polarity. Considering this factor, opinion summarisation based on aspects provides a more meaningful interpretation than an overall sentiment polarity classification. For example, the hotel review presented in Figure 1.1 is partially written as "The hotel is neat, but overpriced, no room service, and they try to screw you with the room selection", in which three aspects (room, value and service) can be identified, with each associated with a sentiment polarity (positive, negative, negative).

Among the various studies relevant to aspect opinion summarisation, some focus on the aspect extraction part of the task to expand the coverage of aspect terms and phrases (Mukherjee and Liu, 2012; Yin et al., 2017), while others concentrate and others concentrate on the aspect-level polarity classification part of the task to improve the classification accuracy of aspect-associated sentiment (Poria et al., 2016; Ruder et al., 2016).

2.1.2 Granularities in sentiment analysis

The other common way of categorising sentiment analysis is based on granularity. The three commonly adopted granularities are *document-level*, *sentence-level*, and *aspect-level*. According to a review of the literature, different features and techniques may be implemented for sentiment analysis depending on whether the source of a sentiment is a document, a sentence or an aspect.

2.1.2.1 Document-level analysis

Sentiment analysis at the document level treats any short or long opinionated document as a whole and regards a single sentiment polarity as sufficient to summarise it. To acquire satisfactory performance in document-level sentiment analysis, studies indicate that the focus should be on different features in short and long documents. For short documents, such as Tweets with a word limit, the focus is more on the identification of expressions that embed sentiments or opinions, due to an assumption that the document only discusses a single topic and all sentiments are associated with it (Kundi et al., 2014; Tang, 2015). For long documents or ones with various lengths, such as lyrics (Kumar and Minz, 2013), diaries (Tai et al., 2015) or essays (Majumder et al., 2017), determination of the overall sentiment polarity of a document is more dependent on an exploration of factors (e.g., topics, Tai et al., 2015; or writers, Majumder et al., 2017) and the weighted contributions of the sentiments associated with these factors.

2.1.2.2 Sentence-level analysis

Sentiment analysis at the sentence level detects sentiments or opinions from sentences, which are generally recognised by punctuation such as full stops, question marks, exclamation marks, etc. As sentence-level sentiment analysis is insufficient for providing summarised information, it is more commonly conducted concurrently with other levels of analysis. Sentence-level sentiment analysis is specifically useful in dealing with two issues: noise filtering and polarity shifts (e.g., the sentence "Fairly good acting, but overall a disappointing movie" contains a polarity shift from positive to negative; Maas et al., 2011; Nakagawa et al., 2010). To be more specific, the first issue is addressed through subjectivity classification (as discussed in Section 2.1.1.2) of sentences to remove objective ones that are regarded as noise, as they have no influence on the overall sentiment of the complex content. The second issue is addressed by capturing relational and syntactical structures among the phases in a sentence using techniques like CRF (Nakagawa et al., 2010).

2.1.2.3 Aspect-level analysis

As some researchers argue that it is rather primitive for document-level sentiment analysis to assume that a piece of textual data only has one single sentiment (Mukherjee and Liu, 2012; Poria et al., 2016; Ruder et al., 2016; Wang et al., 2010), sentiment analysis at the aspect level that identifies aspects expressed in a piece of textual data and classifies the sentiments or opinions associated with each aspect is introduced. This level of study has attracted a lot of interest in the last decade as techniques for other levels of granularity have gradually matured following years of development. An aspect-level sentiment analysis task generally involves two parts: aspect extraction using probabilistic models, e.g., LDA (Poria et al., 2016), or regression analysis (Wang et al., 2010) and sentiment classification using neural network models (Ruder et al., 2016). Considering that an aspect-level sentiment analysis task typically involves aspects and sentiments, most existing studies have focused on the review domain, as review data typically involves multiple aspects (e.g., 'price', 'brand' and 'colour' for a review of a smartphone) and a rating for each aspect that can be used as a ground truth label for evaluations (Poria et al., 2016; Ruder et al., 2016; Wang et al., 2010).

2.2 Document-level sentiment polarity classification

Despite the fact that document-level sentiment polarity classification has been studied for years and derived many advanced techniques with promising outcomes, it remains to be a popular area in the study of sentiment analysis. A recent survey on opinion mining and sentiment analysis conducted by Ravi and Ravi (2015) illustrates statistically that 73 articles of a total of 159 reviewed (46%) are undertaken at document level. The literature also indicates that document-level sentiment polarity classification is ideal for textual data that has the following two factors: a) lengthy contents (e.g., essays or diaries; Majumder et al., 2017; Tai et al., 2015); b) rare data sources with limited evaluation on developed techniques (e.g., lyrics; Kumar and Minz, 2013). Hence, document-level sentiment analysis of Email data is reasonable, considering that it typically has both factors.

In recent years, methods of document-level sentiment polarity classification have evolved significantly, from using manually-annotated sentiment lexicons as guides to automated machine learning algorithms and complex deep neural network models. Thus, a collection of twelve representative papers with features or techniques relevant to the research questions and that have contributed to the development of theoretical fundamentals and concrete techniques were critically reviewed. This will help to better understand and identify existing research gaps and the framework to be developed in this research.

A review of these twelve studies was undertaken based on the taxonomy of characteristics of sentiment classification proposed by Abbasi et al. (2008), which involves tasks, domains, features and techniques. Obviously, the characteristic of tasks, in this section, is related to document-level sentiment polarity classification. Apart from that, a brief summary of the revision indicates that the data domains involved in these studies cover reviews and microblogs, as 11 out of the 12 studies investigated different kinds of review data. Herein, a revised taxonomy of the characteristics involved in the document-level sentiment polarity classification studies reviewed in this research is illustrated in Figure 2.2. The taxonomy is composed of two main parts: features (involving sentiment, sequence and supplementary features) and techniques (involving lexicon-based and machine learning approaches, with the latter further divided into supervised learning and deep learning approaches).

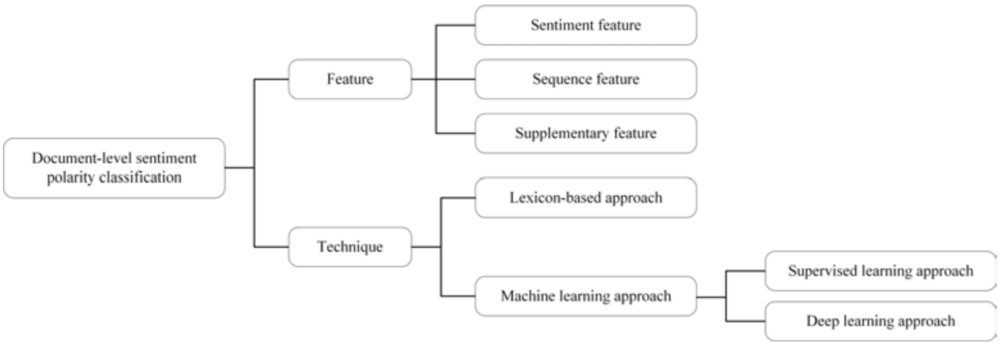


FIGURE 2.2: A taxonomy of the characteristics of the document-level sentiment polarity classification studies reviewed in this study, with regard to features and techniques.

2.2.1 Features

Like text mining and NLP, sentiment polarity classification also relies on *features*, which are the parts of a piece of text that contain the most salient sentiment information for analysis (Pang, Lee, et al., 2008) and help deal with the issue of language complexity in textual data. As suggested by Abbasi et al. (2008), features for sentiment analysis can be broadly categorised as syntactic, semantic, link-based and stylistic features. However, considering the relevance of the three features of Email data identified in Section 1.2, a categorisation of features as *sentiment*, *sequence* and *supplementary* features was adopted, as described below.

2.2.1.1 Sentiment features

Features that capture syntactical and semantic meanings from words and phrases, such as n-grams, word embeddings and sentiment lexicons, are basic word-level features suitable for use in polarity classification. The aforementioned three features are among those most widely adopted in a variety of polarity classification studies (Bespalov et al., 2012; Kundi et al., 2014; Li et al., 2015; Rao et al., 2018; Tang et al., 2015a; Yang et al., 2016b; Yin et al., 2017). An n-gram model measures the similarity between two words based on a predefined n-sequence of characters.

Some studies have proposed revised n-gram models by calculating the probability of the next word in a document via the formation of words into n-gram representations (Bespalov et al., 2012; Li et al., 2015). For instance, Bespalov et al. (2012) incorporated a latent n-gram model as a base feature to generate contiguous sequential representations of documents in vector space. In terms of computational complexity, an $\mathcal{O}(n+1)$ notion is applied to each n-gram variant, meaning that more time and space are consumed accordingly. It was also indicated by Li et al. (2015)'s experimental results that an improvement in accuracy rate occurs with an increase in n. Hence, an n-gram model has to be implemented with great caution, as it is not easy to find a balance between accuracy and computational time.

Word embedding is a technique that models a document as a set of continuous numeric vectors. It was initially introduced as an improvement of the traditional one-hot encoding technique (Harris and Harris, 2015), which is rather computationally inefficient as it converts documents into highly-dimensional representations. Many recent works have intended to use word embeddings as features, as the technique has been observed to be remarkably effective with neural network models used to capture semantic similarities from terms mapped into vectors (Chen et al., 2016; Tang et al., 2015a; Yang et al., 2016b). For instance, Tang et al. (2015a) and Yang et al. (2016b) utilised word embeddings for both sentence- and document-level representations. Additionally, some variant forms of embedding have been developed and implemented based on word embedding techniques such as document embedding (Li et al., 2015) and aspect embedding (Yin et al., 2017).

A *sentiment lexicon* is "a set of words (or phrases) each of which is assigned with a sentiment polarity score" (Wang and Xia, 2017, p. 1). Though the number of studies on lexicon-based polarity classification is low due to issues of unsatisfactory accuracy and time-consumption in generating lexicons, the utilisation of pre-annotated universal sentiment lexicons, such as SentiWordNet(SWN) (Kundi et al., 2014; Rao et al., 2018) and Opinion Lexicon (Kundi et al., 2014), as features are still popular as they are straightforward and easily implemented. Interestingly, it has been observed that sentiment lexicons are more effective as a supplementary feature in addition to other features, e.g., word embeddings. For instance, while Kundi et al. (2014) implemented a set of sentiment lexicons as the only feature and obtained an average precision of 85.5%, which was slightly higher than that of the comparative method (84.4%), Rao et al. (2018) experimented on models with and without a SWN lexicon feature and achieved a much higher accuracy rate with the SWN lexicon feature (46.3%) than without it (43.2%).

2.2.1.2 Sequence features

Features relevant to sequences were brought to attention for document-level sentiment polarity classification, mainly due to the inadequacy of modelling whole documents to determine sentiment polarity (e.g., less than 90% accuracy for binary classification; Kundi et al., 2014; Pang et al., 2002). Therefore, some researchers have approached this issue by making a more reasonable assumption: that a document can have different sentiments expressed in phrases or sentences, and not all of them contribute equally to the overall sentiment polarity of the document. They also believe that effectively capturing weighted sentiments among different building blocks (e.g., words, phrases or sentences) in a document can lead to better classification accuracy in document-level sentiment polarity.

Some studies incorporate sequence features in word- or phrase-level sequences (Bespalov et al., 2012; Matsumoto et al., 2005) or sentence-level sequence (Bhatia et al., 2015; Matsumoto et al., 2005) that capture the relational and syntactical structures among words, phrases or sentences and other lexical or semantic features. For instance, Bespalov et al. (2012) utilised a latent space to represent a phrase positional weighting feature into n-gram sequence embeddings Bhatia et al. (2015) trained a dependency-based discourse tree parser on a sentiment lexicon to construct a rhetorical recursive structure of sentences. Matsumoto et al. (2005) captured frequent word subsequence and dependency subtree patterns from terms in Part-of-Speech(POS) tags and n-gram modelled structures.

To highlight, Mao and Lebanon (2007) proposed a study that addressed the issue of sentiments within documents, which was relatively novel at that time. They

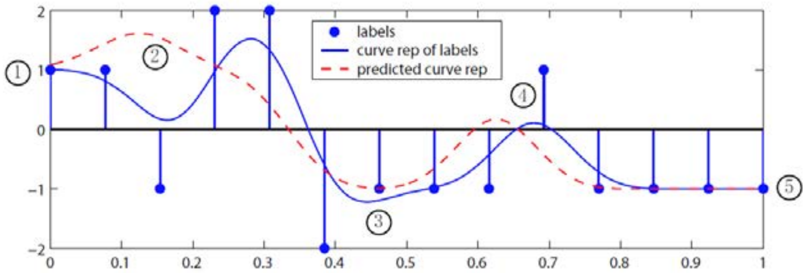


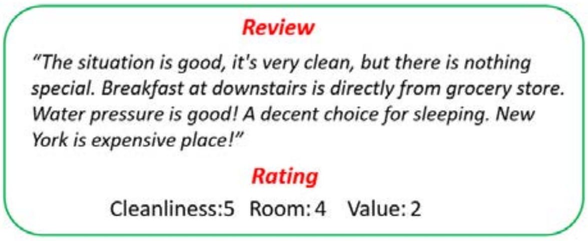
FIGURE 2.3: Sentiment flow and its smoothed curve representation. The blue circles indicate the labeled sentiment of each sentence. The blue solid curve and red dashed curve are smoothed representations of the labeled and predicted sentiment flows. Only non-objective labels are kept in generating the two curves (Mao and Lebanon, 2007).

developed an isotonic CRF model to capture local sentiment flow features (as defined in Figure 2.3) and incorporated them into the final classification process to predict global sentiment polarity. The study undertook a qualitative evaluation of a sample of movie reviews to justify the existence of local sentiment flow and the possibility of applying sentiment flow features to text summarisation. However, this study obtained a relatively low sentiment classification accuracy of 36%, indicating that there is a need to improve techniques of local sentiment discovery.

2.2.1.3 Supplementary features

While some studies attempt to improve the performance of document-level sentiment polarity classification by using sequence features, others approach the problem by using the advantages of supplementary features, such as multi-aspect (Yin et al., 2017), topic distribution (Onan et al., 2016), or user and product features (Chen et al., 2016). Some data domains, such as hotel and product reviews, naturally contain features additional to the review content, such as date, name, rating, etc. Figure 2.4 shows an example of a hotel review containing aspects and ratings.

As an alternative to aspect-level sentiment polarity classification, Yin et al. (2017) proposed a document-level multi-aspect sentiment classification system. Their study adopted a pre-defined list of aspect seed words and converted aspect terms into aspect-specific word embeddings as inputs for the classification model.



Review

"The situation is good, it's very clean, but there is nothing special. Breakfast at downstairs is directly from grocery store. Water pressure is good! A decent choice for sleeping. New York is expensive place!"

Rating

Cleanliness:5 Room:4 Value:2

FIGURE 2.4: Example: hotel review with aspects (Yin et al., 2017).

They also undertook an experiment investigating the influence of the number of aspect keywords. Peak performance was obtained with a small number of keywords. Some researchers have adopted unsupervised learning algorithms, such as LDA topic modelling (Onan et al., 2016), to generate aspect-like features, as some data domains may not fit well with the description of an aspect seed list. For instance, Onan et al. (2016) extracted document-level topic representation using an LDA generative probabilistic model, considering its ability to handle long documents. Moreover, a study conducted by Chen et al. (2016) involved two sets of supplementary features—user and product—and proved the semantic usefulness of incorporating user preferences and product characteristics with quantitative evaluations in the classification process.

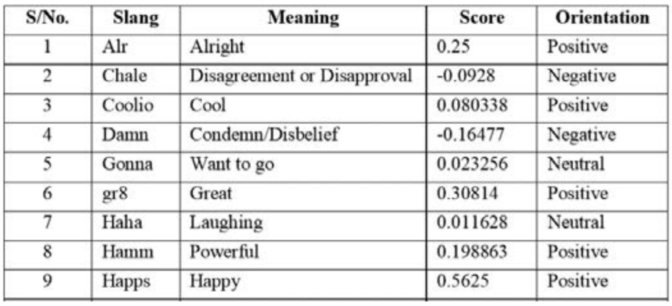
Though improved classification performance due to the use of supplementary features has been reported by these studies, the implementation of supplementary features must be done with caution by attending to two associated issues. On one hand, some of these supplementary features are domain-specific. For instance, a pre-defined list of aspect seed words from hotel reviews does not suit movie reviews. On the other hand, extracting and processing supplementary features requires additional time and space for computation. It is essential to consider a more efficient way of handling supplementary features while improving the performance at a higher level.

2.2.2 Techniques

Briefly, the *techniques* described in the twelve reviewed articles are categorised into lexicon-based, supervised learning and deep learning approaches. Each approach is evaluated according to its advantages and disadvantages. The lexicon-based approach is interpretable with direct observations of sentiment phases and their polarities, yet it is less accurate than other approaches and requires prior knowledge to develop sentiment lexicons. The supervised learning approach is efficient and has high scalability, yet it is less stable in performance due to its dependency on features and domains. The deep learning approach is effective and has high classification accuracy, yet it is less interpretable as the working mechanisms of many deep neural network models are hidden in a "black box". The characteristics of each category are elaborated in the following subsections.

2.2.2.1 Lexicon-based approach

One of the decisive factors in the lexicon-based approach is the quality of the sentiment dictionaries and lexicons involved in the classification process. Some studies focus on the construction of sentiment lexicons to deal with issues such as domain adaptivity (Wang and Xia, 2017), and others concentrate on the discovery of appropriate universal sentiment lexicons and the development of a proper scoring system based on the lexicons. For instance, Kundi et al. (2014) constructed a slang dictionary (as shown in Figure 2.5) containing a set of slangs annotated with scores and orientations using a weighting threshold value computed based on SWN lexicon and developed a scoring algorithm for sentiment prediction.



S/No.	Slang	Meaning	Score	Orientation
1	Alr	Alright	0.25	Positive
2	Chale	Disagreement or Disapproval	-0.0928	Negative
3	Coolio	Cool	0.080338	Positive
4	Damn	Condemn/Disbelief	-0.16477	Negative
5	Gonna	Want to go	0.023256	Neutral
6	gr8	Great	0.30814	Positive
7	Haha	Laughing	0.011628	Neutral
8	Hamm	Powerful	0.198863	Positive
9	Happs	Happy	0.5625	Positive

FIGURE 2.5: Polarity of slang words (Kundi et al., 2014).

Although sentiment lexicons are still widely adopted as features in document-level sentiment polarity classification studies (Bhatia et al., 2015; Rao et al., 2018), the lexicon-based approach is much less popular than machine learning or hybrid approaches owing to its lower classification accuracy. Bhatia et al. (2015) found that their lexicon-based algorithm only obtained accuracy rates of 68.3% and 74.9% on two movie review datasets, which is worse performance than that of a supervised learning classifier, which obtained accuracies of 82.4% and 81.5%.

2.2.2.2 Machine learning approach

Machine learning is a subfield of artificial intelligence involving computational algorithms and statistical models that learn patterns from labelled data (known as *training data*). They then apply rules learned from the preview process to predict unlabelled data (known as *test data*). A more formal definition of the algorithms researched in machine learning is given by Tom M. Mitchell in his book *Machine Learning* (Mitchell et al., 1997): "A computer program is said to learn from experience $\mathcal{E}$ with respect to some class of tasks $\mathcal{T}$ and performance measure $\mathcal{P}$ if its performance at tasks in $\mathcal{T}$, as measured by $\mathcal{P}$, improves with experience $\mathcal{E}$."

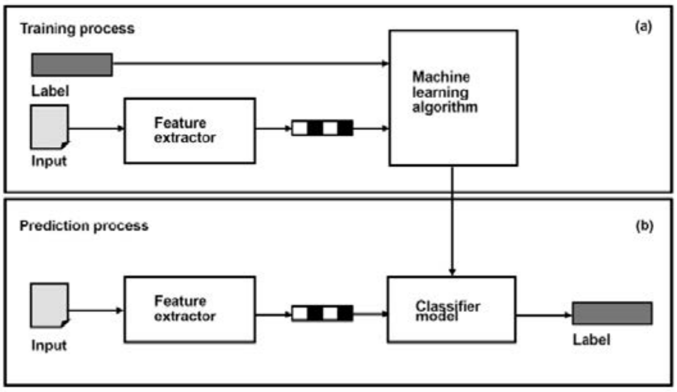


FIGURE 2.6: Process of sentiment analysis (SA). (a) Training process of an SA algorithm with a feature extractor and machine learning algorithm and (b) Prediction process of an SA algorithm with a feature extractor and classifier (Saura and Bennett, 2019).

Figure 2.6 illustrates how a machine learning algorithm is involved in a complete sentiment classification process. Briefly, a training process is required for the algorithm to learn features (as discussed in the previous section) generated by a feature extractor. It then develops a classifier model to perform classification in a prediction process. Among the various types of machine learning algorithms, *supervised learning* and *deep learning* approaches are preferred options for predictive tasks, including document-level sentiment polarity classification. The following few paragraphs provide a more comprehensive discussion of the characteristics of the above two types of approaches.

Supervised learning approach. Algorithms, such as probabilistic models (e.g., Naïve Bayes(NB)), discriminative models (e.g., Support Vector Machine(SVM)), or statistical models (e.g., Logistic Regression(LR)), are different types of supervised learning approaches. Some of these algorithms have been developed for decades and are commonly adopted in a wide range of text mining tasks, including document-level sentiment polarity classification (Bespalov et al., 2012; Li et al., 2015; Mao and Lebanon, 2007; Matsumoto et al., 2005; Onan et al., 2016).

$$\mathcal{L}(\mathcal{X}) = -\sum_{i\in\{1..|\mathcal{X}|\}} \log \frac{\exp(\boldsymbol{\beta}_{y_i}^{\top} \times \mathbf{d}_{\mathbf{x}_i})}{1+\sum_{j\in\{1..C\}}\exp(\boldsymbol{\beta}_{j}^{\top} \times \mathbf{d}_{\mathbf{x}_i})} \qquad \text{(a)}$$

$$\sum_{i}\sum_{j}\left[f\left(x_{w_{i,j}}^{\top}x_{d_i}\right)+\sum_{k=1}^{K} f\left(-x_{w_{\text{random}}}^{\top}x_{d_i}\right)\right] \qquad \text{(b)}$$

FIGURE 2.7: Formulas used in supervised learning approaches. (a) Negative log likelihood function (Bespalov et al., 2012); (b) Negative sampling equation (Li et al., 2015).

As most of the supervised learning algorithms are well-developed and mature, it is common to apply these methods directly for classification purposes. However, as the performance and computational complexity of supervised learning algorithms depend significantly on the training process, some studies have explored techniques to improve training speed and minimise training error (Bespalov et al., 2012; Li et al., 2015). For instance, Bespalov et al. (2012) refined the multinomial LR classifier with the negative log-likelihood function mathematically defined in Figure 2.7 (a), while Li et al. (2015) the negative sampling technique formulated in Figure 2.7 (b) accelerate the speed of training LR classifiers.

In terms of classification performance, a review of these studies indicates that, for binary classification tasks, over 90% accuracy can be obtained with supervised learning methods on movie review datasets (Bespalov et al., 2012; Li et al., 2015). However, some studies also observed a drop in classification accuracy when attempting to involve novel features or experiment on multi-domain datasets (Mao and Lebanon, 2007; Onan et al., 2016).

Deep learning approach. Zhang et al. (2018) defined *deep learning* as "the application of artificial neural networks (neural networks for short) to learning tasks using networks of multiple layers" (p. 2). The history of deep learning can be traced back to the 1990s, when "shallow" models with one or two hidden layers, e.g., artificial neural network and multilayer perception, were introduced (Zhang et al., 2018). Nevertheless, as computing power at that time was inadequate for handling such complex and computationally expansive models, not many applications were involved in early research. With the advancement of hardware and parallel processing techniques, the deep learning approach became increasingly appealing in the area of NLP over these years. Among the twelve reviewed studies, half implement a deep learning approach to perform document-level sentiment polarity classification (Bhatia et al., 2015; Chen et al., 2016; Rao et al., 2018; Tang et al., 2015a; Yang et al., 2016b; Yin et al., 2017).

Neural network models, such as CNN and Recurrent Neural Network(RNN), are typical options for text mining and NLP tasks. Specifically, CNN-based feedforward neural network models that capture features using convolutional filters with weights and biases are implemented for tasks such as word and sentence composition. RNN-based cyclic neural network models that capture sequences using hidden states and time steps have been implemented for tasks such as encoding and decoding. Considering the structural and sentimental complexity in many documents, LSTM has been more frequently adopted as a base model in document-level sentiment polarity classification studies. For instance, Tang et al. (2015a) developed a gated RNN model for document sentiment classification with an LSTM component for sentence composition, while Rao et al. (2018) implemented an LSTM layer for both word- and sentence-level representations.

LSTM is a classic variant of RNN that captures long-term dependencies in sequentially-structured data. The structure of LSTM, with building blocks and operational functions, is described in Figure 2.8 and a more detailed discussion of its working mechanism can be found in Chapter 5.

The *attention mechanism* is a recently proposed technique that further assists

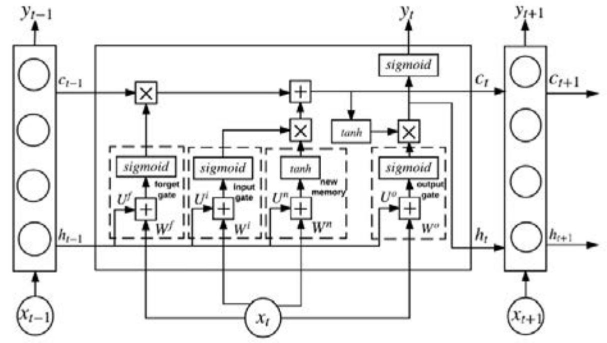


FIGURE 2.8: Long short-term memory network (Zhang et al., 2018).

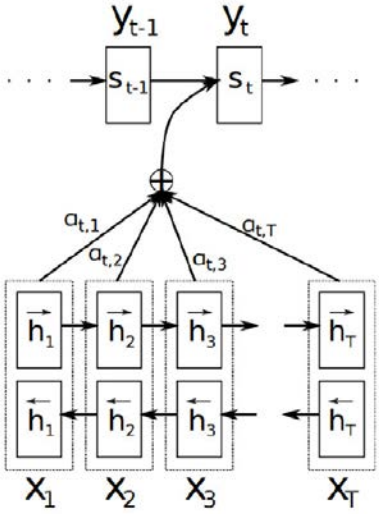


FIGURE 2.9: Attention mechanism in a bidirectional recurrent neural network (Zhang et al., 2018).

LSTMs in handling long-term dependency problems in extremely complex textual structures. It has been implemented in several studies with relatively promising outcomes (Chen et al., 2016; Yang et al., 2016b; Yin et al., 2017). So far, among a list of neural network models experimented on with the same set of benchmarking datasets (Chen et al., 2016; Rao et al., 2018; Tang et al., 2015a; Yang et al., 2016b; Yin et al., 2017), a hierarchical attention network built on a bidirectional gated RNN and attention mechanism at both word- and sentence-level achieved the highest accuracy rate, of around 70%, on multi-class classification of product review datasets (Yang et al., 2016b). A hierarchical LSTM with user and product attention models achieved the highest accuracy rate of 53.3% on multi-class classification of movie review datasets. The workings of the attention mechanism in a bidirectional RNN are presented in Figure 2.9 and readers can refer to Zhang et al. (2018) for more technical details.

2.2.3 Summary of the literature on document-level sentiment polarity classification

A summary of the characteristics, in terms of polarity, domains, features, techniques and performance, obtained from the studies critically reviewed in the previous few sections is presented in Table 2.1. A further discussion of some of the main findings regarding features and techniques is made below.

- Features:
  - The effectiveness of sentiment lexicon features is justified by the results of Kundi et al. (2014) and Rao et al. (2018), who found that their methods performed better with sentiment lexicon features. Additionally, experimental results further imply that a sentiment lexicon is more effective when implemented with other features (Rao et al., 2018).
  - Word embedding is an essential and basic feature for deep neural network models. Pre-trained word embeddings, such as GloVec (Pennington et al., 2014), perform quite well in sentiment classification, but results may vary based on the dimensionality of the embeddings (Rao et al., 2018).
  - Sequence features contribute to better classification performance. The results of Bhatia et al. (2015)'s study indicated an improvement in the classification performance of a lexicon-based method (72.6% and 78.9% with sequence features over 68.3% and 74.9% without) and a supervised classifier (82.9% and 82% with sequence features over 82.4% and 81.5% without) with sequence features on two movie review datasets.
- Techniques:
  - Deep learning approaches tend to be more widely adopted for multi-class sentiment polarity classification and have proven to be more effective for such tasks than other types of approaches.

- For a certain set of benchmarking datasets, slight improvements in classification accuracy have been observed with hierarchical attention networks (Yang et al., 2016b) and hierarchical LSTMs with user and product attention models (Chen et al., 2016).

TABLE 2.1: Summary of the characteristics of document-level sentiment polarity classification studies.

Reference	Polarity	Domain	Feature	Technique	Performance
Matsumoto et al. (2005)	Binary	Movie review	Word subsequence Dependency structure	Supervised learning approach	88.3% - 93.7% accuracy
Mao and Lebanon (2007)	Binary Multi-class	Movie review	Local sentiment flow	Supervised learning approach	Around 38% accuracy Around 36% accuracy
Bespalov et al. (2012)	Binary Multi-class	Product review Hotel review Product review Hotel review	Phrase positional weighting feature n-gram	Supervised learning approach	94.4% accuracy 93.1% accuracy 78.0% accuracy 68.6% accuracy
Kundi et al. (2014)	Binary	Tweets	Sentiment lexicon	Lexicon-based approach	Average 85.8% precision
Bhatia et al. (2015)	Binary	Movie review	Dependency-based discourse tree	Deep learning approach	84.1% - 85.6% accuracy

Continued on next page

Table 2.1 – *Continued from previous page*

Reference	**Polarity**	**Dataset**	**Feature**	**Technique**	**Performance**
Li et al. (2015)	Binary	Movie review	n-gram Document embeddings	Supervised learning approach	92.14% accuracy
Tang et al. (2015a)	Multi-class	Product review Movie review	Word embeddings	Deep learning approach	Over 65% accuracy 45.3% accuracy
Chen et al. (2016)	Multi-class	Product review Movie review	Word embeddings User & product feature	Deep learning approach	Over 65% accuracy 53.3% accuracy
Yang et al. (2016b)	Multi-class	Product review Movie review	Word & sentence embeddings	Deep learning approach	Around 70% accuracy 49.4% accuracy
Onan et al. (2016)	Binary Multi-class	Multi-domain Review	Topic modelling & representation feature	Supervised learning approach	73.4% accuracy 77.21%
Yin et al. (2017)	Multi-class	Hotel review Product review	Aspect feature Word & aspect embeddings	Deep learning approach	46.7% accuracy 38.3% accuracy

Continued on next page

Table 2.1 – *Continued from previous page*

Reference	**Polarity**	**Dataset**	**Feature**	**Technique**	**Performance**
Rao et al. (2018)	Multi-class	Product review Movie review	Word embeddings Sentiment lexicon	Deep learning approach	Over 65% accuracy 46.3% accuracy

2.3 Email data and Email mining

In the history of Email, or Email systems to be more specific, there are two remarkable cases that cannot be ignored. One was the capability to remotely access, store and share files via a central system provided by MIT's Compatible Time-sharing Systems(CTSS) developed in the early 1960s (Nightingale et al., 2008; Tang et al., 2014). The other is the implementation of the symbol '@' as a separator of the name and address domains, which was done in the first Email sent by Ray Tomlinson through ARPANET in 1971 (Bogawar and Bhoyar, 2012; Spicer, 2016; Tomlinson, 2009). It is undeniable that the introduction of Email has significantly influenced how people communicate. Unlike conventional telephone or telegraph, Email allows instant and asynchronous communication at the same time. Due to its low cost, high flexibility and ease of usage, Email is mostly adopted in business settings as a formal means of communication.

While Email communication has a history of nearly 40 years, the study of Email mining only received attention from scholars in the early 20^{th} century when the number of Emails began accumulating enormously. Many Email mining studies highlight the importance of considering the unique characteristics of Email data before further analytical processing (Bogawar and Bhoyar, 2012; Das et al., 2019; Dehiya and Mueller, 2016; Shen et al., 2013; Tang et al., 2005). Owing to these characteristics, preprocessing is an indispensable component in many Email mining tasks. Hence, apart from features and techniques, preprocessing is another important factor to be reviewed in various Email mining tasks.

2.3.1 Characteristics of Email data

Previous studies on Email mining reveal that the main difficulty in applying standard text mining techniques to Email data is due to the unique characteristics that differentiate Email data from other types of textual data. Tang et al. (2005) focused on Email data cleaning methods that are specifically associated with the characteristics of Email data; that is, their noisy contents and complex structures. Moreover, Bogawar and Bhoyar (2012) highlighted that "a distinctive separating line (p. 3)" is set between Email and text mining, and that "some text mining techniques might be inefficient in email data (p. 3)" due to its specific characteristics. The example given in Figure 1.1 (a), Chapter 1, serves as visual evidence of the different kinds of information that a raw Email might contain. In detail, the main characteristics of Email data include:

- **Noise content.** Generally, noisy content in Email data can be broadly categorised as *duplication* and *linguistic errors* (Bogawar and Bhoyar, 2012; Dehiya and Mueller, 2016; Tang et al., 2005). In terms of duplications, as a one-to-many communication tool, Email has the options of replying to and forwarding messages. Owing to these features, it is highly likely that some replies or forwarded messages contain duplicated text. Additionally, as a formal means of communication, Email data, especially business Emails, are structured with greeting and signature blocks, which are additional duplicated components. In terms of linguistic errors, Tang et al. (2005) suggested that the types of linguist errors in Email data can vary from mistakenly removed or placed punctuation, unnecessary spaces, badly-cased words, misspelled words, etc. Bogawar and Bhoyar (2012) further highlighted that, in some extreme cases (spam Emails for instance), excessive noise is intentionally inserted in the form of unusual words and phrases.

- **Unstructured content.** Unlike noisy content that is embedded in the original Email message, unstructured content is mainly derived from online systems during the data collection process. Mark-ups, Hypertext Mark-up Language(HTML) tags and attachments are common unstructured features of Email data (Blanzieri and Bryl, 2008; Bogawar and Bhoyar, 2012; Dehiya and Mueller, 2016; Tang et al., 2014).While any noisy content is assumed to negatively impact the analytical results, some unstructured contents, such as Uniform Resource Locator(URL) links and attachments, might be useful for some Email mining tasks (Tang et al., 2014).

- **Meta-information.** Blanzieri and Bryl (2008) stated that a whole Email can be separated into a body and header, with the meta-information generally contained in the header part. Information such as date, address, sender, receiver, etc. are common types of meta-information. Many Email mining studies report that, depending on the task requirements, some meta-information might be useful (Blanzieri and Bryl, 2008; Bogawar and Bhoyar, 2012; Shen et al., 2013; Tang et al., 2005). For instance, Email data begins with a subject field that provides an overview of the focus of the main message. Hence, subject information is often treated as a part of the message body and can be helpful in better understanding the Email content.

- **Lengthy.** As observed from the discussion on general document-level sentiment polarity classification, the length of a document is one of the factors that determine which features and techniques should be implemented. Email data can be extremely variant in length. For instance, a statistical summary of the characteristics of Enron's Email corpus shows that the first quartile of length is 46 words and the third quartile reaches 466 words, indicating that there is a wide range of lengths in the corpus (Das et al., 2019).

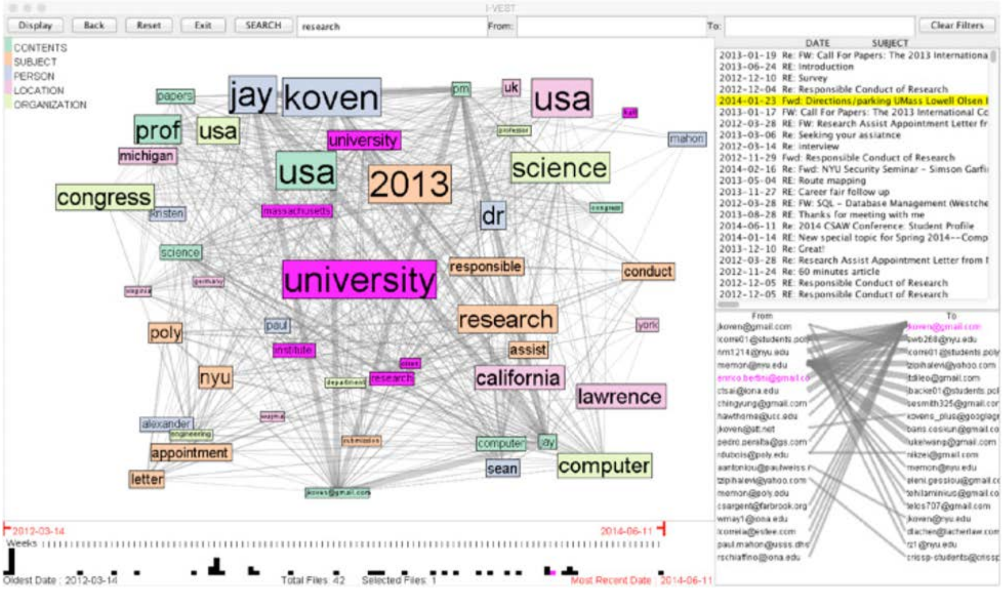
2.3.2 Email mining tasks and techniques

Email mining covers a wide range of tasks involving summarisation and visualisation (Dredze et al., 2008; Koven et al., 2016), thread detection (Sharaff and Nagwani, 2016; Ulrich et al., 2008), spam classification (Blanzieri and Bryl, 2008; Ezpeleta et al., 2016) and sentiment related tasks such as emotion visualisation (Hangal et al., 2011; Mohammad and Yang, 2011), personality detection (Shen et al., 2013), etc. As previous studies on Email sentiment analysis are limited, a collection of sixteen publications on Email mining tasks were grouped into non-sentiment-related and sentiment-related tasks in a review of the preprocessing, features and techniques involved in these studies.

2.3.2.1 Non-sentiment related tasks

As previously discussed, Email data suffers from serious data overloading problems. To investigate possible solutions to this problem, three types of Email mining tasks are performed by the research community: *Email summarisation and* *visualisation*, *thread identification*, and *spam detection*. The first two types approach the issue by categorising Email into groups and understanding Email communication and structures through visual supports. The latter one approaches the issue by filtering out useless Email messages.

Email summarisation and visualisation. The main purpose of this task is to assist in better understanding the associations among different Emails through summarising and grouping a large set of Email data into certain categories, e.g., topic keywords (Dredze et al., 2008) or time stamps.



DATE	SUBJECT
2013-01-19	Re: FW: Call For Papers: The 2013 Internationa
2013-06-24	RE: Introduction
2012-12-10	RE: Survey
2012-12-04	Re: Responsible Conduct of Research
2014-01-23	Fwd: Directions/parking UMass Lowell Olsen I
2013-01-17	FW: Call For Papers: The 2013 International Cc
2012-03-28	RE: FW: Research Assist Appointment Letter fr
2013-03-06	Re: Seeking your assiatnce
2012-03-14	Re: interview
2012-11-29	Fwd: Responsible Conduct of Research
2014-02-16	Re: Fwd: NYU Security Seminar - Simson Garfi
2013-05-04	RE: Route mapping
2013-11-27	RE: Career fair follow up
2012-03-28	RE: FW: SQL - Database Management (Westche
2013-08-28	RE: Thanks for meeting with me
2014-06-11	Re: 2014 CSAW Conference: Student Profile
2014-01-14	RE: New special topic for Spring 2014--Comp
2013-12-10	Re: Great!
2012-03-28	Re: Research Assist Appointment Letter from I
2012-11-24	Re: 60 minutes article
2012-12-05	RE: Responsible Conduct of Research
2012-12-05	RE: Responsible Conduct of Research

FIGURE 2.10: InVEST display showing relationships to a selected file (Koven et al., 2016).

Some studies aim at developing interactive visual analytic systems for better Email management (Koven et al., 2016; Li et al., 2004a), and others investigate methods that perform summarisation of Email data for further application, such as personal Email prioritisation (Yoo et al., 2009) and template induction (Proskurnia et al., 2017). For instance, Koven et al. (2016) developed a visual analytic system known as InVEST (see Figure 2.10), which has filtering, expansion and organisation functionalities. With Email preprocessed by duplication and junk removal and indexing, the system can output query results based on keyword and entity rankings. A study conducted by Dredze et al. (2008) also addressed the problem of keywords. Rather than visualisation, this study explored methods of classifying Email based on topic keywords. The study indicated that LDA-based methods performed better than Latent Semantic Analysis(LSA)-based methods in terms of an automated folder categorisation task. Furthermore, use of a combination of keywords and subjects significantly improved the discovery of useful information.

Thread identification. Sharaff and Nagwani (2016) define *Email thread identification* as "a process of identifying the chronological chain of email messages based on their content (p. 1)." Based on this definition, it is obvious that topic and temporal features are crucial in the process of thread identification (Sharaff and Nagwani, 2016; Ulrich et al., 2008). Sharaff and Nagwani (2016) utilised a clustering approach, as defined in Figure 2.11, using LDA topic modelling and non-negative matrix factorization with people and subject similarity features that are compared with the *k*-Means algorithm. As statistical models and clustering approaches are sensitive to noisy data, a preprocessing phase with stop word removal and stemming was implemented. The results obtained from this study indicate that LDA topic modelling was effective in discovering threads and Email clusters, with subject similarity features contributing more to the classification accuracy than the people similarity and combined similarity features.

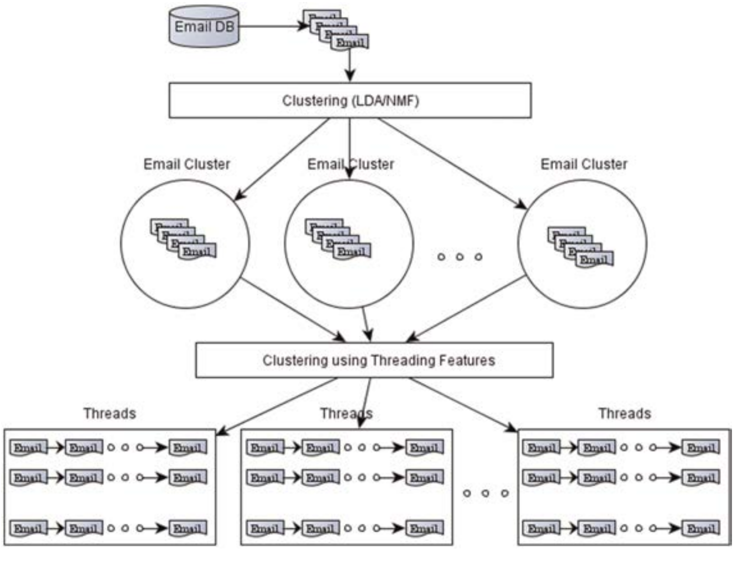
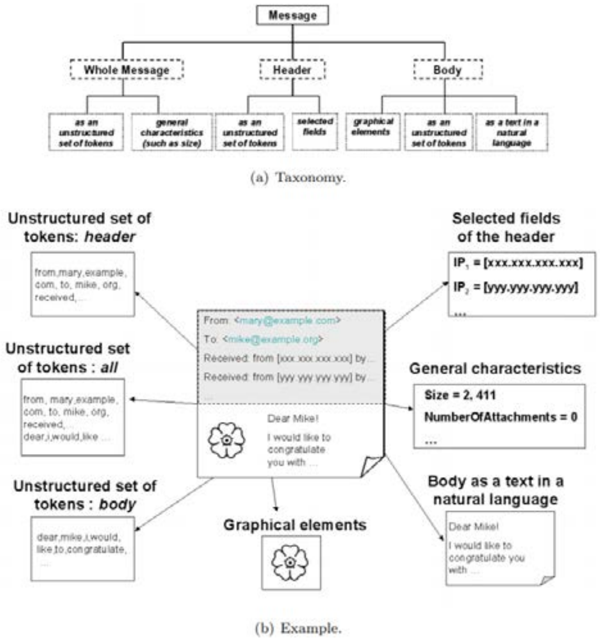


FIGURE 2.11: The process of email thread identification (Sharaff and Nagwani, 2016).

Spam detection. Spam detection is a binary classification task that determines whether a piece of Email data is a legitimate mail (known as *ham*) or a junk mail (known as *spam*; Blanzieri and Bryl, 2008). Spam detection is an appealing field of study in Email mining that has produced many research and review papers in recent years. As it is not the main focus of the present research, a review paper that summarises popular methods of spam detection is evaluated.


Message
Whole Message
Header
Body
as an unstructured set of tokens
general characteristics (such as size)
as an unstructured set of tokens
selected fields
graphical elements
as an unstructured set of tokens
as a text in a natural language
Unstructured set of tokens: header
from,mary,example, com, to, mike, org, received,...
Selected fields of the header
IP1 = [xxx.xxx.xxx.xxx]
IP2 = [yyy.yyy.yyy.yyy]
...
From: <mary@example.com>
To: <mike@example.org>
Received: from [xxx.xxx.xxx.xxx] by...
Received: from [yyy.yyy.yyy.yyy] by...
Dear Mike!
I would like to congratulate you with ...
Unstructured set of tokens : all
from, mary,example, com, to, mike, org, received,... dear,i,would,like ...
General characteristics
Size = 2, 411
NumberOfAttachments = 0
...
Unstructured set of tokens : body
dear,mike,i,would, like,to,congratulate, ...
Graphical elements
Body as a text in a natural language
Dear Mike!
I would like to congratulate you with ...

(a) Taxonomy.

(b) Example.

FIGURE 2.12: What to analyse? Message structure from the point of view of feature selection (Blanzieri and Bryl, 2008).

In their study, Blanzieri and Bryl (2008) proposed a taxonomy of features to be analysed in the spam detection process (Figure 2.12). An Email message was analysed at the levels of header, body and whole message. While some features, such as an unstructured set of tokens, were observed in all three parts, certain features were relevant to spam detection, including graphical elements and attachments. Blanzieri and Bryl (2008) also gathered and analysed common methods implemented in various spam detection studies involving statistical models (e.g., Term Frequency-inverse Document Frequency(TF-IDF), Bag-of-Words(BoWs) and n-grams), supervised learning approaches (e.g., k-Nearest-Neighbor(k-NN), SVM and NB), and hybrid approaches (e.g., combined statistical feature modelling with supervised learning classifiers). Evaluation of their performance indicated that supervised learning and hybrid approaches obtained better classification results than keyword-based or statistical-based models.

It was further summarised from Blanzieri and Bryl (2008)'s review paper that although spam detection sentiment analysis are both classification tasks, there were differences in the types of features to be used and the algorithm evaluation criteria.

2.3.2.2 Sentiment related tasks

To review sentiment-related tasks, I make a straightforward grouping of tasks into *lexicon-based*, *machine learning* and *hybrid* approaches to present a concise understanding of the technical gaps existing between Email sentiment analysis and the proposed framework.

Lexicon-based approach. The literature indicates that lexicon-based approaches comprise the most popular method of Email sentiment analysis, with half of the reviewed articles utilising them (Das et al., 2019; Dehiya and Mueller, 2016; Hangal et al., 2011; Mohammad and Yang, 2011). However, it must be highlighted that most of these studies only incorporated sentiment analysis as a functional part of overall Email analysis rather than as pure Email sentiment analysis.

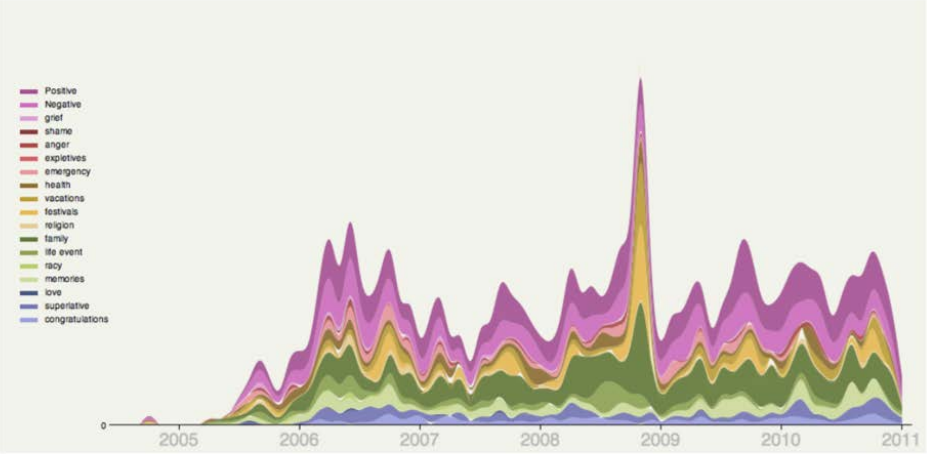


FIGURE 2.13: A MUSE visualization of email sentiment. A stacked graph shows the number of email messages reflecting a particular sentiment category over time (Hangal et al., 2011).

In detail, Hangal et al. (2011) developed an interactive visual analytic system named MUSE (as presented in Figure 2.13) for analysing and visualising patterns in Email archive contents. Email data was preprocessed with stop word removal, word factoring and stemming using the Stanford NLP toolkit before further analytical processing. As one of the functionalities in the system, sentiment analysis was performed with a lexicon-based approach using a self-generated English lexicon with terms covering 20 categories, such as emotions, family, health, etc. The results were visualised through stacked graphs over time series. Similarly, Dehiya and Mueller (2016) implemented a stacked plot to visualise sentiment terms associated with *country* and *year* using a sentiment lexicon in an auxiliary analysis of Hillary Clinton's Emails. Das et al. (2019) proposed a RegTech solution for systematic and effective risk and malaise detection in Emails. Sentiment analysis was involved as part of the solution by performing word classification with sentiment word lists to assist in various risk detection tasks. In contrast, Mohammad and Yang (2011) proposed a study on more fine-grained emotion analysis through building a word-emotion association lexicon through crowdsourced annotation of n-gram corpus and English thesaurus. Emotion detection and identification of the gender difference in using emotional words in love letters, hate mails, and suicide notes were conducted using the emotion lexicon with a relative frequency formulated on ratio difference. A bar graph and tag cloud were adopted to visualise the summary of emotional terms and their priorities.

Though lexicon-based approaches were implemented in all four studies, it is assumed that since the main focus of the first three studies was not sentiment analysis, lexicon-based approaches were adopted simply due to their ease of implementation. Hence, these studies have issues such as a lack of comparative evaluation and scalability. Additionally, though Mohammad and Yang (2011) performed sentiment analysis of Email data, generation of the word-emotion association lexicon through crowdsourcing was relatively time-consuming and involved excessive human intervention.

Machine learning approach. Both unsupervised clustering methods and supervised classifiers are common techniques utilised in sentiment-related Email studies (Chhaya et al., 2018; Liu et al., 2016). Liu et al. (2016) implemented a clustering approach using density-based spatial clustering of applications with a noise algorithm on the Enron Email corpus with consideration of BoWs and sentiment lexicon features. A further topic and temporal classification process was undertaken on the Email clusters for visualisation purposes. However, this study had the same problem as the previous four in terms of a lack of quantitative evaluation of classification performance. Focusing on this issue, the study conducted by Chhaya et al. (2018), though not solely on sentiment polarity classification, was reviewed as it involved empirical experiments on classification performance. Chhaya et al. (2018) described a detailed annotation process of a subset of the Enron corpus and utilised this labelled dataset for quantitative evaluations. This dataset was also utilised as a benchmark dataset in this study. Though the results of Chhaya et al. (2018) indicate a relatively good outcome, with an accuracy of 86% with a random forest classifier, the study performed a binary classification task and only experimented on different supervised learning classifiers for comparative evaluations.

Hybrid approach. (Shen et al., 2013) and (Liu and Lee, 2015) used hybrid approaches involving combinations of 1) statistical models with supervised learning approaches and 2) unsupervised learning algorithms with supervised learning approaches. Interestingly, both studies implemented a preprocessing phase with common steps including duplication removal and POS tagging.

In detail, Shen et al. (2013) undertook a personality prediction task with Email messages. A set of features involved in personality prediction included BoWs features, meta-features, POS-tagged contents and a sentiment polarity dictionary. BoWs features were trained on a self-generated word list composed of 20,000 common words obtained from TV and movie scripts. Meta-features included several items, such as counts of replied/forwarded Emails, numbers, attachments, punctuation symbols, etc. Classification was performed with a hybrid approach using various combinations of a generative/statistical model (ensemble model, labelled LDA model or Bayes probabilistic model) and a supervised learning classifier (NB, SVM or Random Forest(RF)). The experiments did not produce consistent performance with any one algorithm, as SVM performed best on some tasks while random forest performed best on others. However, an experiment of accuracy sensitivity with various sizes of training and testing datasets implied that classification accuracy increased with dataset size. A hybrid approach combining a k-Means algorithm with supervised learning algorithms was proposed in Liu and Lee (2015) as an attempt to perform sentiment classification on unlabelled data through a pseudo-labelling process. Though high accuracy was obtained with k-Means labelling and an SVM classifier, the quality of the pseudo-labels was doubtful. Nevertheless, the study did prove the effectiveness of the SWN lexicon, which produced the second-highest accuracy after k-Means labelling.

2.3.3 Summary of the literature on Email data and Email mining

A summary of relevant Email mining tasks, in terms of preprocessing, features and techniques, is presented in Tables 2.2 and 2.3. A further discussion of some of the main findings of this review, based on the aforementioned three factors, is provided below.

- Preprocessing:
  - Both non-sentiment-related and sentiment-related studies highlight the importance of text normalisation and Email cleaning in the preprocessing phase.
  - Removal of unnecessary contents (e.g., duplication and stop words) and meta-information handling, are two steps to be addressed in multiple sentiment-related tasks.
- Features:
  - Email data is rich in various types of meta-information. Good use of this feature has been made in many non-sentiment-related Email mining studies. It is assumed that some of this meta-information might also be useful in sentiment analysis.
  - Only conventional features, e.g., sentiment lexicons and syntactic features, have been addressed in existing Email sentiment analyses. However, it is emphasized that the unique features (sentiment sequences

and multi-topics) that influence the performance of sentiment classification should be considered.

- Techniques:
  - Many visual analytical tools have been developed for Email data and for use in Email sentiment analysis. Lexicon-based approaches with graph visualisation are more commonly implemented in existing sentiment-related studies. However, Email sentiment analysis is a classification task that requires proper predictive techniques.
  - Some supervised learning approaches, such as SVM (Shen et al., 2013) and RF (Chhaya et al., 2018), have been demonstrated to be effective in terms of classification performance compared with other supervised learning classifiers. Nevertheless, the review of document-level sentiment polarity classification indicates that there is a wide range of algorithms of different categories, e.g., neural network models, that have been developed for this task. Hence, a more comprehensive evaluation of the performance of these methods for Email sentiment analysis is needed.

TABLE 2.2: Summary of relevant non-sentiment-related Email mining studies.

Non-sentiment related study	Task	Preprocessing	Feature	Technique
Li et al. (2004a)	Email summarisation Email visualisation	-	n-gram	Statistical approach Graph visualisation
Blanzieri and Bryl (2008)	Spam detection	Stop words removal Stemming	Header features Body features	Statistical modelling Supervised learning approach Hybrid approach
Dredze et al. (2008)	Email summarisation	-	Query-document similarity word association	Unsupervised learning approach Probabilistic topic models
Ulrich et al. (2008)	Thread identification	-	Meta features	Speech act theory
Yoo et al. (2009)	Email summarisation	Address canonicalisation	Meta features Social importance features	Newman clustering method SVM
Koven et al. (2016)	Email summarisation Email visualisation	Duplication and junk removal Indexing	Name entity recognition Keyword and term ranking	Unsupervised learning approach Interactive visual analytic
Sharaff and Nagwani (2016)	Thread identification	Stop words removal Stemming	Thread features Subject similarity	Unsupervised learning approach: LDA and NMF
Proskurnia et al. (2017)	Email summarisation	Filtering Lemmatisation	Term count vectors Phrase frequency	Unsupervised learning approach Greedy algorithm

TABLE 2.3: Summary of relevant sentiment-related Email mining studies.

Sentiment related study	**Task**	**Preprocessing**	**Feature**	**Technique**
Hangal et al. (2011)	Email summarisation Emotion visualisation	Stop words removal Word factoring Stemming	Emotion lexicon	Lexicon-based approach Web interface Graph visualization
Mohammad and Yang (2011)	Emotion visualisation	-	English thesaurus Word n-gram	Lexicon-based approach Graph visualization Tag cloud
Shen et al. (2013)	Personality detection	Thread separation Signature blocks and reply lines removal POS tagging	Sentiment polarity lexicon Pronouns Negations	Hybrid approach: Statistical model Supervised learning classifier
Liu and Lee (2015)	Sentiment classification	Duplication and stop words removal Stemming POS tagging	Bag-of-words Sentiment lexicon	Hybrid approach: k-Means clustering Supervised learning classifiers
Liu et al. (2016)	Sentiment clustering Email summarisation	Duplication and stop words removal	Topic and temporal features Bag-of-words Sentiment lexicon	Temporal classification DBSCAN clustering Tag cloud
Dehiya and Mueller (2016)	Sentiment visualisation	Meta-information handling	Sentiment terms Temporal features	Lexicon-based approach Graph visualisation
Chhaya et al. (2018)	Emotion classification Tone detection	-	Lexical and syntactic features Affect-based and derived features Word embeddings	Supervised learning approach: Random forest
Das et al. (2019)	Sentiment visualisation Risk detection	Meta-information handling POS tagging	Sentiment word list Temporal features	Lexicon-based approach Graph visualisation

2.4 Research gaps

After comprehensively and critically reviewing studies on document-level sentiment polarity classification and Email mining, several research gaps were identified that are relevant to this study on document-level Email sentiment analysis. Described below are research gaps related to the five functions of the proposed framework, involving Email data, noise handling, sentiment sequence, sentiment classification and quantitative evaluation (as discussed in Section 1.3).

- **Email data.** The literature review suggests that existing sentiment analysis techniques are inadequate for addressing the sentiment sequences and multi-topic features identified in Email data. As for the techniques developed for Email sentiment analysis, none of them is capable of handling sentiment sequences or has been fully quantitatively evaluated.

- **Noise handling.** Email-specific cleaning is covered in many non-sentiment-related studies (Blanzieri and Bryl, 2008; Das et al., 2019; Koven et al., 2016; Proskurnia et al., 2017; Sharaff and Nagwani, 2016; Yoo et al., 2009), which provide detailed descriptions of the implementation process. It is also covered in some sentiment-related studies (Dehiya and Mueller, 2016; Hangal et al., 2011; Liu and Lee, 2015; Liu et al., 2016; Shen et al., 2013), which can serve as useful references. However, none of these studies has conducted any quantitative evaluations of the influence of the noise handling process.

- **Sentiment sequence.** There is a distinct gap in this area. Sequence-related features are incorporated in some sentiment classification studies (Bespalov et al., 2012; Mao and Lebanon, 2007; Matsumoto et al., 2005), whereas the sentiment sequence feature is only addressed in Mao and Lebanon (2007)'s study. Although Mao and Lebanon (2007) proved the existence of sentiment sequences within documents, the overall classification performance was unsatisfactory.

- **Sentiment classification.** Apart from the gaps mentioned in Email data functions, sentiment classification for Email data also lacks benchmarked results for many state-of-the-art algorithms. It was observed that most existing Email sentiment analysis studies have implemented lexicon-based and machine learning approaches, whereas more advanced deep neural network models have been developed for other types of data.

- **Quantitative evaluation.** Some Email sentiment analysis studies only focus on visual analytics without any quantitative evaluation of classification performance (Dehiya and Mueller, 2016; Hangal et al., 2011; Mohammad and Yang, 2011). It is argued that these studies lack detailed methods of sentiment visualisation and that their analyses are narrow and limited to specific datasets. In those studies that did perform a certain level of quantitative evaluation, issues such as a lack of comparative algorithms (Shen et al., 2013) and data scarcity (Chhaya et al., 2018) were observed.

In this research, a document-level Email sentiment analysis framework is proposed to cover all five functions. To address the Email data function, three benchmark Email datasets are prepared as the fundamentals for empirical experiments. Proper data preprocessing and augmentation methods are implemented to address the noise handling function. Furthermore, comprehensive quantitative evaluations on the comparison between raw data, preprocessed data and augmented data are undertaken to justify the essentiality of preprocessing and augmentation in obtaining satisfied Email sentiment classification results. A novel clustering-based trajectory representation approach is introduced to address the sentiment sequence function as trajectory clustering methods are capable of capturing sequence features in data and providing interpretable visual outputs. As literature indicates a superiority of the classification performance of deep neural network models over lexicon-based and supervised learning-based algorithms, deep neural network models are utilised as the base algorithms to incorporate sentiment sequence and multi-topic features to address the sentiment classification function. And finally, to address the quantitative evaluation function, benchmarking results with both baseline and state-of-the-art algorithms are obtained through experimenting on the three Email datasets.

A summary of the above discussion is presented in Table 2.4 which lists the research gaps identified in this critical review of document-level sentiment classification and Email mining research.

TABLE 2.4: Summary of research gaps. ○ represents task fulfilled; △ represents task partially fulfilled; and ✗represents task unfulfilled.

Reference	Email data	Noise handling	Sentiment sequence	Sentiment classification	Quantitative evaluation
Li et al. (2004a) Ulrich et al. (2008)	○	✗	✗	✗	✗
Blanzieri and Bryl (2008) Koven et al. (2016) Sharaff and Nagwani (2016) Proskurnia et al. (2017)	○	○	✗	✗	✗
Dredze et al. (2008)	○	✗	✗	✗	△
Yoo et al. (2009) Das et al. (2019)	○	○	✗	✗	△
Hangal et al. (2011) Liu et al. (2016) Dehiya and Mueller (2016)	○	○	✗	△	✗
Mohammad and Yang (2011)	○	✗	✗	○	△
Chhaya et al. (2018)	○	✗	✗	○	△
Shen et al. (2013) Liu and Lee (2015)	○	○	✗	○	△
Blanzieri and Bryl (2008)	○	○	✗	✗	△
Mao and Lebanon (2007)	✗	✗	○	△	△
Matsumoto et al. (2005) Bespalov et al. (2012)	✗	✗	△	○	○
Kundi et al. (2014) Li et al. (2015) Tang et al. (2015a) Bhatia et al. (2015) Chen et al. (2016) Onan et al. (2016) Yang et al. (2016b) Yin et al. (2017) Rao et al. (2018)	✗	✗	✗	○	○
Proposed framework	○	○	○	○	○

2.5 Research hypotheses

On the basis of the research gaps identified from the critical review of the literature, hypotheses associated with each research question (covered in Section 1.3) were formulated and are summarised in Table 2.5.

TABLE 2.5: Summary of research aims, objectives and hypotheses.

Question	Hypothesis
1. What preprocessing methods are essential in addressing unstructured and noisy contents in Email data and can solve the issues of data scarcity and imbalanced class distributions in labelled Emails?	1.1 Email cleaning with text normalisation will reduce the impact of noise and unstructured content and positively influence classification performance.
	1.2 Data augmentation will solve the scarcity and imbalanced class distribution issues that are common to labelled Email data.
	1.3 Supervised learning techniques and neural network models will provide better classification performance with augmented datasets than non-augmented ones.
2. How to effectively capture sentiment sequence features and discover sentiment sequence patterns within Email data?	2.1 Sentiment sequence features can be embedded in sentiment trajectories built from Email documents and captured through sentiment trajectory representation.
	2.2 Sentiment sequence features will contribute positively to classification performance and can be discovered through a trajectory clustering approach.
3. How to encode sentiment sequence features in a neural network model for robust and accurate sentiment polarity classification?	3.1 Sentiment sequences can be encoded through position and sentiment lexical features.
	3.2 Sentiment sequence-encoded CNN models will provide better classification performance than baseline, unsupervised learning and supervised learning approaches.
	3.3 Algorithms with sentiment sequence features will provide better classification performance than algorithms without them.
4. How to capture multi-topic features and model documents with multi-topic segments for effective sentiment polarity classification?	4.1 LDA topic modelling and semantic text segmentation techniques can effectively model the multi-topic features of Email documents.
	4.2 Topic weighting and topic features generated by the LDA topic modelling will improve the performance of polarity classification.
	4.3 Multi-topic features will positively contribute to classification performance and MT-BiLSTM will outperform other sentence- or document-level neural network models.

3 Overall structure of document-level Email sentiment analysis

In this chapter, I demonstrate the overall structure of a document-level sentiment analysis method for Email data that is built upon a high-level summarisation of the topics covered in the following three data chapters. The general flow of work consists of four major steps: preprocessing, feature generation, document vectorization and sentiment analysis. Section 3.1 summarises the overall structure and topics associated with each data chapter. Section 3.2 describes the benchmark Email datasets used for quantitative evaluation in this thesis. And Section 3.3 describes the complete Email data preprocessing method.

3.1 Document-level sentiment analysis of Email data

The general framework for document-level sentiment analysis of Email data is composed of four major steps: preprocessing, feature generation, document vectorisation and sentiment analysis (as demonstrated in Chapter 1). As discussed in Chapter 1, the overall research problem is approached by conducting three studies, on sentiment sequence clustering, sequence-encoded neural sentiment classification and multi-topic neural sentiment classification. Each study covers a combination of topics defined in the overall structure presented in Figure 3.1. In detail, the task of sentiment sequence clustering is explored through sentiment trajectory, trajectory representation, sentiment trajectory clustering and categorical and temporal classification. The task of sequence-encoded neural sentiment classification is explored through sentiment sequencing, word embedding, sentiment sequence encoding and polarity classification. The task of multi-topic neural sentiment classification is explored through document segmentation, word embedding and polarity classification.

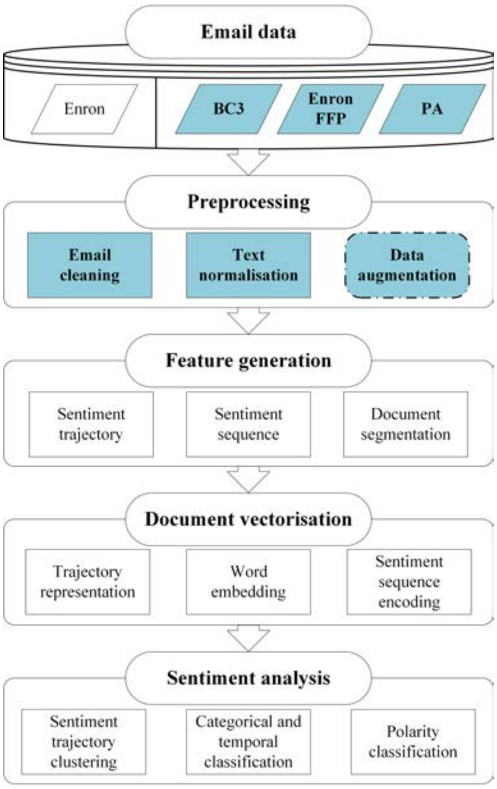


FIGURE 3.1: Overall structure of the Email document sentiment analysis framework, with the specific topics covered in this chapter highlighted in blue and bold.

All three studies utilise the same sets of labelled Email data for empirical evaluation and implement a preprocessing phase as standard sentiment analysis practice. A thorough discussion of the data collection and label conversion processes used with three benchmark Email datasets and the detailed steps of the preprocessing phase are presented in this chapter.

3.2 Email datasets

To perform quantitative evaluations of the feasibility and effectiveness of each proposed approach in the following chapters (e.g., benchmark analysis), three medium-sized Email datasets were collected, including two publicly-available datasets and one private Email archive. Considering the limitation of publicly labelled Email datasets for classification evaluations, it is expected that to generate larger genuine Email datasets for further experiments is a potential research direction and stated in Chapter 7 as one of the major limitations in this study. Nevertheless, to investigate the possible solutions to the data scarcity issue, a data augmentation method was proposed and discussed in detail in Section 3.3.2. Initially, the three Email datasets were labelled based on different standards and numbers of classes. The class labels of the two public datasets were adjusted through straightforward statistical modelling methods to minimise the influence of labelling bias due to adjustment and ensure the reliability and authenticity of the classification performance. A brief justification of the sources of the three Email datasets and the corresponding label conversion process is presented. Table 3.1 summarises the overall class distribution of the three datasets.

TABLE 3.1: Summary of class distributions of three datasets (NoE: Number of Emails).

Dataset	**NoE**	**Labelled NoE**
BC3	255	Positive - 147 Negative - 29 Neutral - 79
EnronFFP	960	Strongly Negative - 172 Negative - 214 Neutral - 574
PA	600	Positive - 150 Negative - 128 Neutral - 322

3.2.1 BC3 dataset

Abbreviated as BC3, the British Columbia Conversation Corpora (Ulrich et al., 2008) is an Email corpus that contains 255 messages generated from 40 email threads. In the original dataset, all subject sentences in each Email message are assigned a class label of $positive$ (P), $negative$ (N), $both$ (PN), or $neither$ (X). Hence, to acquire a document-level three-class label, the majority voting technique (Scott and Matwin, 1999), which is a straightforward discriminative modelling technique that has been well accepted for text classification tasks, was applied to the Email corpus for the purpose of concatenating document-level sentiment labels. In detail, whether an Email document is labelled as positive, negative or neutral is determined by the total number of subject sentences in a document. A document is assigned with a class label of positive if it contains more positive sentences than negative ones; negative otherwise. If a document contains no subject sentences or only X labels, it is assigned as neutral. A final three-class distribution of 147 positive, 29 negative and 79 neutral Emails was obtained.

3.2.2 EnronFFP dataset

Abbreviated as EnronFFP, the second dataset is a subset of the original large Enron Email corpus with frustration, formality and politeness annotations. Derived from the study conducted by Chhaya et al. (2018), which focused on quantifying feelings and tones in Emails, the EnronFFP dataset is pre-labelled with three sets of tags: frustration, within an interval of $[-2,-1,0]$, formality of $[-2,-1,0,1,2]$, and politeness of $[-2,-1,0,1,2]$. Previous psychological studies suggest that *frustration* is a standard measurement of a negative feeling set by seven visual analogue scales (Dill and Anderson, 1995; Wade et al., 1990). Thus, it is justifiable to use frustration labels as a reference to identify negative Emails in the dataset. To obtain a three-class sentiment-oriented label that is consistent with other datasets, the labels were converted into $Strongly\ Negative$ (SN), $Negative$ (N), and $Neutral$ (Neu), using frustration as a referential criterion and converting frustration scores into sentiment labels based on the mean value of 10 annotators' scores. The mathematical formula for the above-mentioned label conversion process is depicted as follows:

$$f(s) = \begin{cases} -2, & \text{if } \frac{\sum_{i=1}^{10} s_i}{10} < \mu, \\ -1, & \text{else if } \mu < \frac{\sum_{i=1}^{10} s_i}{10} < 0, \\ 0, & \text{otherwise.} \end{cases} \tag{3.1}$$

In Equation 3.1, the value of μ is computed as -0.7 that is, the mean frustration score of the entire Email corpus. The final three-class distribution ends up with 172 strongly negative, 214 negative and 574 neutral Emails.

3.2.3 PA dataset

Abbreviated as PA, the Personal Archive dataset is a manually labelled dataset containing 600 Email messages originating from the author's personal Outlook Email account. In general, the main annotation process was manually conducted by a sender and the corresponding recipient of each Email message in the dataset. Then, a random third independent annotator from the authors' list was assigned for validation. Each Email was assigned to a score set of three with an interval of $[-1, 1]$ based on its sentiment granularity, or assigned as 0 if it was neutral. The overall score of an Email was computed by the weighted average of its score set, in which 50% came from the sender, 30% from the recipient and 20% from the independent annotator. The scores were converted into labels based on the same criterion, with a total score greater than 0 labelled as positive, less than 0 as negative, and equal to 0 as neutral. The labelled Email distribution contained 150 positive, 128 negative and 322 neutral Emails.

3.3 Preprocessing

The literature review (Chapter 2) highlighted the necessity of data preprocessing in sentiment analysis and Email mining-related tasks. Proper preprocessing methods contribute to better performance and higher efficiency. Considering the quality and quantity of the experimental Email datasets used in this research, the preprocessing phase was further divided into three parts: data augmentation, Email cleaning and text normalisation. Details of each part are discussed in the following sections.

3.3.1 Email cleaning and text normalisation

To acquire high-quality and effective analytical results, data quality must be ensured by using proper cleaning and normalisation methods. For the purposes of Email cleaning and unification, a pre-developed package named $EmailParser$[6] is applied to identify and remove greetings (e.g., 'Hi xx' or 'Dear xx'), and signature blocks (e.g., 'Regards xx' and 'Sincerely xx'). Moreover, the regular expression functions ($Pattern$; Berk and Ananian, 2005) package in Java and (re; Goyvaerts and Levithan, 2012) module in Python are utilised to perform duplication removal by scanning and filtering out content from Emails that begin with or contain the keywords 'original', 're:' or 'reply', and 'fw:' or 'forward' in either the Email subject or body, as well as unstructured expressions and mark-ups such as '&amp;', 'quot;', '&gt', etc. Text normalisation tasks, including tokenisation, lowercase, stop word removal, stemming or lemmatisation and POS tagging, are implemented using various NLP toolkits (Stanford Core NLP toolkit[7] and Apache Lucene OpenNLP toolkit[8] in Java and $nltk$ toolkit[9] in Python). Additionally, a $NEGATION\_WORD\_LIST$ derived from (Wilson et al., 2005) is adopted to perform negation handling, and a short word removal step ($len()$) is implemented using the Python generic function.

Among the aforementioned normalisation steps, POS tagging, a process used to tag words according to their lexical categories in a sentence, is an indispensable component of SWN lexicon-involved subtasks. Derived from the WordNet(WN), the SWN lexicon is a publicly available lexical reference widely adopted for sentiment classification and opinion extraction tasks. Some recent research indicates wide use of the SWN lexicon for feature extraction in sentiment analysis and related tasks. For instance, Kundi et al. (2014) proposed a score-based approach for discovering sentiments from slang words using the SWN lexicon in order to generate semantic values. Kumar and Minz (2013) utilised the SWN lexicon to extract sentiment features from song lyrics for mood detection. Tai et al. (2015)

[6] https://github.com/mynameisvinn/EmailParser
[7] https://stanfordnlp.github.io/CoreNLP/
[8] https://opennlp.apache.org/
[9] https://www.nltk.org

implemented LDA in combination with the SWN lexicon for calculating emotion scores from users' diaries to detect mental disorders. POS tagging is an essential step in the normalisation method because SWN 3.0 (Baccianella et al., 2010) assigns a sentiment score to a word based on its word class. For example, for an input word 'good', the SWN lexicon returns 0.55 for the noun form or 0.63 for the adjective form. Consistent with the criteria of the SWN lexicon (which will be discussed in the next section), Apache's scheme of POS tagging is converted into wider categories. For instance, phrase-type nouns, including NN (noun) and NNP (proper noun), are categorised into n; phrase-type adjectives, including JJ (adjective), JJR (adjective, comparative) and JJS (adjective, superlative) are categorised into a.

A pseudocode for the complete scheme of Email cleaning and text normalisation ($EmailCN$) is presented in Pseudocode 1. Denote $\mathcal{ED}$ as a collection of Email documents consists of messages $\{ed_1, ed_2, \ldots, ed_n\}$, and $\mathcal{T}$ as a list of tokens $\{t_1, t_2, \ldots, t_m\}$ in each Email message $ed_i \in \mathcal{ED}$. Noted that steps involving length handling (*len*()), stop word removal and POS tagging are marked as optional and their implementation is to be discussed in each data chapter in detail.

3.3.2 Data augmentation with random word replacement

A review of deep learning indicates that, in addition to data quality, data quantity also has a significant impact on the performance of neural network models (Wu et al., 2019). Moreover, a relatively balanced class distribution of datasets is essential to ensure moderate constraint of model training and stable variance of model estimation (Wang et al., 2016). However, due to the lack of a large volume of publicly available labelled Email datasets, and the fact that manually labelled data requires huge human effort and amounts of time, the use of machine learning techniques to automatically generate synthetic data was considered. Inspired by past studies that utilise data augmentation methods to enlarge the scale of the dataset (Wang and Yang, 2015; Wei and Zou, 2019; Zhang et al., 2015), a hybrid

Pseudocode 1 EmailCN

```
Input: A set of raw Email documents $\mathcal{ED}$;
Output: Each refined Email document $ed_i \in \mathcal{ED}$ represented with a collection of
  post-processed tokens $t_m$ in a word list $\mathcal{T}$;
for each Email document $ed_i \in \mathcal{ED}$ do
    Convert each document $ed_i$ into $Email()$ object;
    Apply $EmailParser()$ and regular expression functions;
    Return a revised Email document $ed_i$;
    Apply tokenisation to $ed_i$;
    for each token $t_j \in \mathcal{T}$ do
        Convert $t_j$ to lowercase;
        /* Optional step*/;
        if $len(t_j)$ is less than 3 then
            Remove $t_j$ from $\mathcal{T}$;
        end if
        /* Optional step*/;
        if $t_j \in STOP\_WORD\_LIST$ then
            Remove $t_j$ from $\mathcal{T}$;
        end if
        if $t_j \in NEGATION\_WORD\_LIST$ then
            Replace $n't$ with the word $not$;
        end if
        Check $t_j$ spelling with $SpellChecker()$ function;
        Apply stemming or lemmatisation to $t_j$;
        /* Optional step*/;
        Apply POS tagging to $t_j$ and convert tags based on rules defined;
    end for
    Return each refined Email document $ed_i$ with a list of post-processed words
  $\mathcal{T}$;
end for
```

method with a combination of a k-NN classifier with word embeddings and a WN lexicon (Miller, 1995) is implemented to handle unique Email data.

To minimise the influence of noisy data and increase processing speed, a similar Email cleaning and text normalisation process to that shown in Pseudocode 1 is applied to the initial raw Email data to generate clean and reliable vocabularies. Specifically, the Python generic function len is implemented for short word removal and $nltk$ toolkit and re (Goyvaerts and Levithan, 2012) module are used to perform functions including tokenisation ($tokenize$()), lowercase (*lowercase*()), spell check ($SpellChecker$()), stop word removal ($STOP\_WORD\_LIST$) and lemmatisation ($lemmatize$(); Perkins, 2014).

After proper cleaning and normalisation, a word replacement dictionary containing vocabulary words and their synonyms or terms of similar usage using word embeddings with the k-NN classifier is generated as the first step. A post-processed Email corpus is tokenised into a list of vocabularies and each document is transferred into a collection of numeric vectors using pre-trained Glovec word embeddings (Pennington et al., 2014). The k-NN classifier is applied to the vectorised corpus to identify the first five nearest-neighbouring words for each term in the vocabulary, which are then stored in a dictionary. Then, the WN lexicon and its synonym thesaurus are utilised to filter out improper replacement terms, such as acronyms generated by word embeddings, and to expand the coverage of the existing dictionary. If a keyword in the dictionary is indexed in the WN lexicon, then any of its values that do not get returned as synonyms by WN are removed, and additional synonyms that do not exist as values are appended. Examples of vocabulary words and their replacement terms in the dictionary are presented in Table 3.2.

TABLE 3.2: Sample words and their replacement terms.

Word in Emails	Word in Dictionary
accused	['accuse', 'impeach', 'incriminate', 'criminate', 'charge']
broadly	['loosely', 'generally']
email	['mail', 'twitter', 'facebook', 'message']
grateful	['thankful', 'thank', 'glad', 'happy', 'wish']
educational	['education', 'academic', 'learning', 'teaching', 'community']
unfortunately	['unluckily', 'regrettably', 'alas']
enroll	['inscribe', 'enter', 'enrol', 'recruit']
plot	['game', 'patch', 'diagram', 'plat']

Finally, synthetic documents are constructed using the above-described dictionary. To determine the probability of a word being replaced, a threshold value δ of 0.5 is defined. As suggested by Wei and Zou (2019), using a random synonym with a replacement rate of 20% or less for each sentence yields better performance. In the present study, different probabilities were tested and a final threshold value of δ equal to 0.5 was selected, which resulted in a 5–20% replacement rate for each document. Apart from stop words, each word in a document is assigned a random value that is compared with δ. If a word has a random number greater than δ and existed as a keyword in the dictionary, then it is replaced by one of its random values. Table 3.3 lists some example sentences with their synthetic ones.

TABLE 3.3: Example sentences and their synthetic equivalents.

Original	Synthetic
I don't get any unusual code.	I don't get any strange code. I don't have any strange cipher.
Actually I think there are some potentially interesting effectual ramifications.	Actually I recall there are some potentially interesting legal ramifications. Actually I suppose there are some potentially interesting sound ramifications.
It's a good piece of work.	It's a effective part of work. It's a good nibble of work.

To evaluate the influence of data quantity on sentiment classification performance, three sets of augmented data scaled with different ratios and balanced ratios were generated, which are summarised in Table 3.4 with details of their class distributions. To be more specific, the first half of the table lists the class distributions of three augmented datasets based on the ratios to their original. For instance, a ratio of @10 for the BC3 dataset resulted in a distribution of 1470, 290 and 790 for positive, negative and neutral Emails respectively, which was ten times of the original number of each class. While the second half of the table lists the class distribution of three augmented datasets based on the balanced ratios to their original. The augmented numbers were obtained based on the ratio of one class. For instance, a balanced ratio of #10 for the BC3 dataset resulted in a distribution of 290, 290 and 290 for positive, negative and neutral Emails respectively, which was ten times of the original number of the negative class used as a reference.

3.4 Conclusions

In this chapter, a general framework for document-level sentiment analysis of Email data was presented to lead a brief overview of the three studies to be discussed in the following chapters. In addition to the general framework, three labelled Email datasets were introduced with detailed generation and label conversion processes as a high-level summarison of the classification evaluations of the feasibility and effectiveness of each proposed method in this thesis.

TABLE 3.4: Summary of distributions over three labels of three augmented datasets in different ratios (NoE: Number of Emails; @: Ratio to its original; #: Balanced ratio to its original).

Ratio	Labelled NoE@10	Labelled NoE@20	Labelled NoE@30	Labelled NoE@50	Labelled NoE@100
BC3	Positive - 1470 Negative - 290 Neutral - 790 Total - 2550	Positive - 2940 Negative - 580 Neutral - 1580 Total - 5100	Positive - 4410 Negative - 870 Neutral - 2370 Total - 7650	Positive - 7350 Negative - 1450 Neutral - 3950 Total - 12750	Positive - 14700 Negative - 2900 Neutral - 7900 Total - 25500
EnronFFP	Strongly Negative - 1720 Negative - 2140 Neutral - 5740 Total - 9600	Strongly Negative - 3440 Negative - 4280 Neutral - 11480 Total - 19200	Strongly Negative - 5160 Negative - 6460 Neutral - 17220 Total - 28800	Strongly Negative - 8600 Negative - 10700 Neutral - 28700 Total - 48000	Strongly Negative - 17200 Negative - 21400 Neutral - 57400 Total - 96000
PA	Positive - 1500 Negative - 1280 Neutral - 3220 Total - 6000	Positive - 3000 Negative - 2560 Neutral - 6440 Total - 12000	Positive - 4500 Negative - 5120 Neutral - 9660 Total - 18000	Positive - 7500 Negative - 6400 Neutral - 16100 Total - 30000	Positive - 15000 Negative - 12800 Neutral - 32200 Total - 60000
Balanced ratio	**Labelled NoE#10**	**Labelled NoE#20**	**Labelled NoE#30**	**Labelled NoE#50**	**Labelled NoE#100**
BC3	Positive - 290 Negative - 290 Neutral - 290 Total - 870	Positive - 580 Negative - 580 Neutral - 580 Total - 1740	Positive - 870 Negative - 870 Neutral - 870 Total - 2610	Positive - 1450 Negative - 1450 Neutral - 1450 Total - 4350	Positive - 2900 Negative - 2900 Neutral - 2900 Total - 8700
EnronFFP	Strongly Negative - 1720 Negative - 1720 Neutral - 1720 Total - 5160	Strongly Negative - 3440 Negative - 3440 Neutral - 3440 Total - 10320	Strongly Negative - 5160 Negative - 5160 Neutral - 5160 Total - 15480	Strongly Negative - 8600 Negative - 8600 Neutral - 8600 Total - 25800	Strongly Negative - 17200 Negative - 17200 Neutral - 17200 Total - 51600
PA	Positive - 1500 Negative - 1500 Neutral - 1500 Total - 4500	Positive - 3000 Negative - 3000 Neutral - 3000 Total - 9000	Positive - 4500 Negative - 4500 Neutral - 4500 Total - 13500	Positive - 7500 Negative - 7500 Neutral - 7500 Total - 22500	Positive - 15000 Negative - 15000 Neutral - 15000 Total - 45000

To address the first research question and the corresponding hypotheses as discussed in Chapter 2, a preprocessing stage that involved Email cleaning, text normalisation and data augmentation was developed as part of the framework. Email cleaning and text normalisation was implemented using pre-developed NLP functions and packages. An additional Email parser was applied to raw Email documents to handle special greetings and signature blocks. Data augmentation with a random word replacement technique was implemented using a k-NN classifier trained on word embeddings and a WN lexicon. The method was applied to the post-processed data to generate two sets of augmented datasets, one based on ratios and the other based on balanced ratios, for the evaluation on the essentiality of the preprocessing methods to Email sentiment classification.

4 Sentiment sequence discovery with trajectory representation for Email data[10]

In this chapter, I describe a novel type of sequence-based sentiment analysis that uses trajectory representation to discover sentiment sequences and clusters in Email data. Section 4.2 reviews the literature in the field of sentiment analysis with sequence- and trajectory clustering-related techniques. Sections 4.3 and 4.4 discuss the formulation of the problem, in which documents are transformed into sentiment trajectories through a three-stage trajectory representation approach. I evaluate the proposed method by conducting empirical experiments on real Email datasets. Section 4.5 summarises the sentiment trajectory patterns with temporal categories, while Section 4.6 summarises the empirical results. Figure 4.1 illustrates the topics in the overall Email sentiment analysis structure that will be covered in-depth in this chapter.

4.1 Introduction

Document sentiment analysis is generally considered as a multitudinous problem composed of several sub-problems such as aspect extraction and grouping, feature extraction, and sentiment classification (Liu, 2012). For years, document-level sentiment analysis has focused on the refinement and development of feature

[10] This chapter is written based on the following publication 'Liu, S., & Lee, I. (2018). Discovering sentiment sequence within email data through trajectory representation. Expert Systems with Applications, 99, 1-11.'

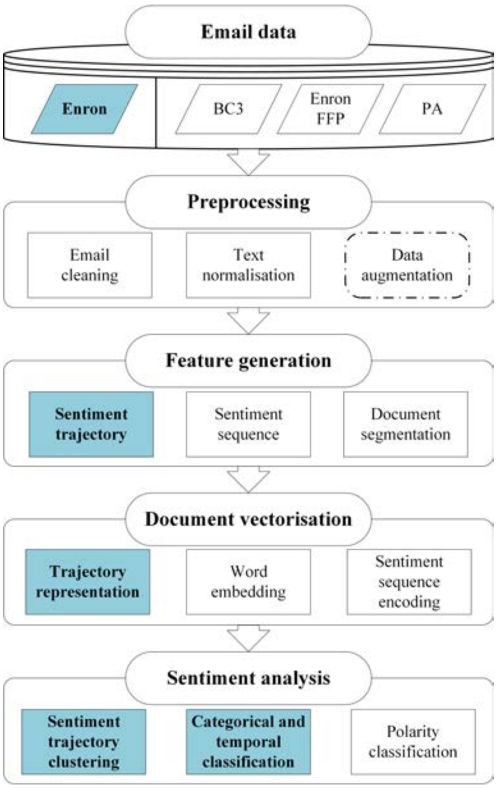


FIGURE 4.1: Overall structure of the Email document sentiment analysis framework, with the specific topics covered in this chapter highlighted in blue and bold.

extraction and sentiment classification techniques (Bhatia et al., 2015; Li et al., 2015; Liu et al., 2016; Moraes et al., 2013; Tang, 2015; Tang et al., 2015b). For text mining problems involving feature identification or extraction processes, the sequence of words or phrases is a prominent concept applied to various term weighting schemes such as n-gram models and CRF (Bao et al., 2004; García-Hernández et al., 2006; Mao and Lebanon, 2007; Matsumoto et al., 2005).

As a crucial factor for correctly identifying the sentiments of a document, sequence features, such as phrase or word sequences, within documents is to be recognised. Mao and Lebanon (2007) introduced the concept of local sentiment flow for the first time and used a modified CRF to analyse the sentiment flow in sentences within a document. Deep learning techniques, such as word embedding that incorporates sequencing in the feature selection process, are increasingly appealing and have been applied to document sentiment analysis (Tang et al., 2015a,b). Nevertheless, to the best of my knowledge, no study has been conducted on discovering sentiment sequences within documents as part of sentiment clustering or classification processes.

As highlighted in Chapter 1, the sentiment sequence is one of the distinctive features of Email that is investigated in this research. Considering that the lengths of Emails can be extremely variant (depending on whether they are an original, replied-to or forwarded message), conventional sentiment analysis techniques may be inadequate for identifying sentiment sequence patterns within them (Blanzieri and Bryl, 2008; Bogawar and Bhoyar, 2012; Hangal et al., 2011). Traditional feature-based classification algorithms take vectorised documents as inputs and generate a single polarity label for each document as outputs without a proper visual support of the sentiments within the documents. Specifically, traditional techniques used for other social media data, which mainly focus on enhancing emoticon and irregular expression detection, are inadequate. Therefore, it is necessary to explore a novel approach to sentiment sequence clustering in Email data and to the discovery of sentiment sequence patterns within Email data. It is hypothesised that the sentiment sequence features could contribute to the improvement of Email sentiment classification. With the assistance of the sentiment sequence clustering results, a better understanding on how sentiments are expressed in Emails and Emails are communicated can be obtained.

The main aim of this study was to propose a sequence-based sentiment clustering technique for improving document-level sentiment analysis. The essence of sequence discovery within documents for sentiment clustering lies in the way of extracting feature words. Feature-based document sentiment classification extracts the frequency or weighting of features in a document. For example, the following two review fragments convey positive sentiments at document level;

> "Overall, I like this hotel. The room is clean and service is good. But the food in the hotel café is awful."
>
> "I would stay here again. The location more than made up for any problems we had with the room. The staff is excellent and very friendly."

however, they are not identical. Conventional document sentiment classification rules generally treat features in a static way without considering the interaction among documents, whereas two documents classified as positive may express different sentiments based on the position of features within documents, as shown in the given example ($positive \rightarrow positive \rightarrow positive \rightarrow negative$ for the first review while $positive \rightarrow positive \rightarrow negative \rightarrow positive$ for the second review). On the contrary, this novel sequence-based sentiment analysis introduces the concept of sequence within documents in sentiment analysis considering the chronological presence of features, which minimises the opportunity of clustering sentences conveying the same sentiments into different categories.

As a result of incorporating spatial information from the text into the feature generation process, trajectory clustering (by means of a clustering algorithm particularly developed for spatial datasets) is used in comparison with other traditional sentiment classification algorithms. With the unique characteristics of Email data discussed above, the trajectory clustering algorithm is capable of handling instances with various attribute lengths and assigning them a set of sentiments instead of a single polarity.

This chapter provides an elaboration of the proposed unsupervised learning-based approach to clustering sentiment sequences and classifying sentiments in Email data. In accordance with the sequence features in a trajectory representation, a revised trajectory clustering algorithm is defined and developed. The five major contributions of this chapter are as follows:

- introducing a novel way to solve sentiment analysis tasks based on sentiment sequence features within documents by using trajectory representation for Email sentiment pattern recognition;
- proposing a technique for transforming features into a 2-dimensional trajectory representation;
- discovering sentiment sequences within documents in temporal categories and clustering sentiment polarities using trajectory clusters;
- visualising sentiment sequences aligning with original Email messages represented as sentiment features as well as Email messages in categorical

and temporal distributions; and

- evaluating the efficiency and effectiveness of the proposed method with real-life Email datasets.

4.2 Related work

The concept of sequences is not unique in sentiment analysis, whereas using trajectory representation for sentiment analysis is relatively novel. Hence, I review previous studies on sentiment analysis with sequences and relevant trajectory clustering techniques to assess the feasibility of utilising sentiment trajectory representation for sequence clustering.

4.2.1 Sentiment analysis with sequence

Studies relevant to sentiment sequences involve document sequences and temporal sentiment analysis. In the previous few decades, some techniques have been proposed and developed for studying document sequences. Most studies conducted on document sequences focus on linguistic comparisons and grammatical relationships. For instance, Wei and Chang (2007) developed a technique for discovering evolutionary patterns in sequential documents based on temporal relationships. Bao et al. (2004) applied semantic sequence kin and word sequence kernels to document copy detection. Furthermore, Jindal and Liu (2006) proposed an approach known as *class sequential rule mining* that uses machine learning techniques to identify comparative sentences.

Apart from studies on document sequences, Matsumoto et al. (2005) proposed a novel feature selection technique using syntactic relations for the extraction of word sub-sequences. Mao and Lebanon (2007) developed a revised CRF for the prediction of ordinal sequences in word sets. However, traditional studies share problems such as a lack of temporal information and limitations in discovering sentiment sequences. The temporal sequence is considered to be another form of sentiment sequence. An increasing quantity of studies has been undertaken on temporal sentiment analysis in the past few years, as the incorporation of temporal features with sentiment analysis has become increasingly appealing to researchers. For example, Fukuhara et al. (2007) implemented a coefficient model for displaying patterns and relationships among categories, timestamps and sentiments using graphs. Diakopoulos et al. (2010) extracted temporal trends in categories and keywords from news data generated from social media using an automated visualisation tool called Vox Civitas.

However, a review of past studies indicates that sentiment sequences within documents have not yet been studied, while the usage of Email data for sentiment analysis and linking temporal clustering and sentiment sequence identification is rare. Therefore, a method for discovering sentiment sequence patterns within documents is needed.

4.2.2 Trajectory clustering

A *trajectory* is a representation of the movement of a mobile object. Yao (2003) stated that "spatiality and temporality are two unique dimensions in geography (p. 2)." As mentioned earlier, this research conducted sentiment analysis from a sequence-based perspective to discover sentiment sequence patterns within documents. To achieve this, traditional methods of transforming documents into features represented by vectors is inadequate. Since sentiment variation within documents is denoted by the positions of features in combination with their sentiment values, a trajectory space that models the movement of spatio-temporal datasets is an ideal option for representation. Therefore, a trajectory clustering algorithm was utilised in the proposed framework for clustering document sentiments that are represented as trajectories. By transforming text features into spatio-temporal features, sentiment sequence detection in spatiotemporally-represented documents differs from general sentiment classification tasks. Therefore, conventional sentiment analysis algorithms are unsuitable for solving the problem, as most adoptable classifiers, such as SVM and NB, are only able to handle points rather than sequences.

Clustering is a process of assigning a set of randomly generated objects into groups based on a similarity measurement. Trajectory clustering was specifically developed for grouping moving objects, or spatial-temporal data, and for discovering patterns in the representative trajectories of each cluster. As an evolving field of study, quite a few trajectory clustering algorithms have been proposed. For instance, Li et al. (2010) proposed a trajectory clustering method using a micro- and macro-clustering framework called TCMM for incremental clustering of micro-clusters and viewing of current clustering results. Li et al. (2004b) extended the micro-trajectory clustering algorithm with time interval embedded to moving micro-clustering denoted as MMC. Yu et al. (2013b) developed a density-based trajectory clustering algorithm with a tree structure called CTraStream for clustering real-time incremental and high-scale trajectory data streams. Bermingham and Lee (2015) extended TRACLUS (Lee et al., 2007) to higher dimensions in order to handle n-dimensional trajectories. TRACLUS is a sub-trajectory clustering approach based on a partition-and-group framework. Although algorithms for clustering trajectories have been refined and diversified, TRACLUS was used in this research for two main reasons. First, TRACLUS, that utilise a partition-and-grouping framework to identify and cluster trajectories based on similar sub-trajectories, is the fundamental algorithm in the field of trajectory mining and has more widely applications than its variations. Second, TRACLUS can handle highly-dimensional data, which is a major feature of trajectory-represented documents. Details of the TRACLUS algorithm will be discussed in the following sections.

4.3 Problem statement

As stated in the previous section, this study introduces the concept of sequences in features and conducts sentiment analysis from a different viewpoint. The major problem to be solved in this research is to discover the sentiment sequence within a document and to assign sentiment polarities to trajectory clusters. To achieve this aim, the concept of *sentiment trajectory* and its relevant feature transformation and trajectory representations must be defined.

Sentiment trajectory A trajectory, generally referred to as a spatial trajectory, is represented by a chain of geographical points generated from moving objects, such as vehicles and people (Zheng, 2015). A trajectory is a set $\mathcal{T} = \{(x_1, y_i, t_1), \ldots, (x_n, y_n, t_n)\}$ of points where (x_i, y_i) represents a spatial position and t_i denotes a corresponding temporal context for $0 \leq i \leq n$. Since the principle of the movement of an object has a close connection to the sentiment flow of a document, the concept of trajectory mining can be reasonably applied to sentiment analysis. To be more specific, each sentiment feature in a document is equivalent to a location on a map and is uniquely represented by a set of attributes similar to geographical coordinates. Herein, a sentiment trajectory is formed by considering each Email message as a representation of chronologically ordered sentiment points. Herein, a sentiment trajectory is defined as a set $\mathcal{T_S} = \{(p_1, s_1), \ldots, (p_n, s_n)\}$ of temporal sentiments, where p_i denotes a temporal position within the document and s_i is a corresponding sentiment for $1 \leq i \leq n$.

Trajectory representation Three phases of trajectory representation were identified for building sentiment trajectories and performing sentiment trajectory clustering. These were sentiment features, pseudo-longitude and latitude transformation, and pixel conversion. A *sentiment feature* is a representation of a set of coordinates formed by the value of an opinion word and its corresponding position in a document on the basis of a specific sentiment corpus. Each Email message is represented by a chronological sequence of sentiment features. The *pseudo-longitude and latitude* represent a set of geospatial coordinates converted by the normalisation of sentiment features. A *pixel* represents a set of map projector coordinates used for displaying patterns on a map that has been pre-scaled in the TRACLUS algorithm. Map projector coordinates are generated through a systematic transformation of geospatial coordinates for visualisation. The main purpose of converting sentiment features into pixels is to allow application of the TRACLUS algorithm to trajectory clustering.

Problems Different from traditional feature-based sentiment analysis, the sequence-based sentiment clustering problem to be solved in this study can be epitomised by two aspects:

1. Discovering sentiment sequences within Email messages; and
2. clustering sentiment polarities based on the sentiment trajectory clusters.

This study focuses on solving the second problem and aims to justify the assumption that sentiment sequences within a document influence the sentiment polarity to be clustered or classified. There are challenges to the two problems: 1) information loss during the transformation process and 2) unbalanced feature distributions in Email datasets. The main aim of this paper was to conduct sequence-based sentiment clustering using the TRACLUS algorithm. To achieve this aim, spatial information is incorporated during the feature extraction process and documents are transformed into trajectories that are computable by the TRACLUS algorithm. The TRACLUS algorithm is an unsupervised learning method that is relatively hard to evaluate in sentiment analysis tasks. Hence, it was modified to be capable of assigning sentiment polarities to each instance that was able to be validated using a pre-labelled dataset. Although a quantitative evaluation is rather hard to undertake due to the lack of a labelled Email dataset, the proposed technique was preliminarily validated via two rounds of experiments. In a pilot experiment, a small set of manually labelled Email data and a larger pre-rated review dataset were utilised. For the main experiment, three benchmark Email datasets (discussed in Chapter 3) were adopted. Note that I do not recommend applying the proposed technique to other types of labelled datasets as it is designed specifically for sentiment sequence features in Email data. Besides, the criteria of sentiment classification vary among studies. Therefore, the empirical results obtained with the review dataset are expected to outperform those of other similar techniques, yet this is not guaranteed. In this study, an accuracy and RMSE value for three-class classification were applied as an evaluation matrix. Instead of binary classification, Email, by nature, contains fewer emotions and more likely to be classified as neural due to its implicitness. In addition to quantitative results, qualitative results were obtained based on the sentiment sequence patterns discovered from the publicly available Enron Email dataset. These patterns were visualised through sentiment clustering results predicted by sentiment sequence within document trajectory representatives and temporal and categorical distributions.

4.4 Proposed sentiment trajectory representation technique

This section presents the sentiment trajectory representation technique proposed for discovering sentiment flow in documents and for clustering sentiment polarities into three classes (positive, neutral and negative) using trajectory space. The approach uses an unsupervised learning-based algorithm with a combination of text mining and trajectory clustering.

Figure 4.2 reflects the general flow of the techniques proposed for discovering sentiment sequence patterns within Email documents. The cleaning phase involves a series of Email cleaning and text normalisation processes, such as tokenisation and stemming (as standard text processing steps) and stop word removal and POS tagging (as sentiment-specific steps). The SWN 3.0 (Baccianella et al., 2010) lexicon is used as an initial sentiment feature generator. The trajectory representation process is implemented to generate sentiment trajectory features with a trajectory format for use with the sentiment sequence clustering algorithm. A modified TRACLUS algorithm is utilised for temporal and sentiment trajectory clustering. Details of the Email cleaning and text normalisation phases are depicted in Section 3.3.2 and Pseudocode 1. To be more specific, the Java package $Pattern$ Berk and Ananian, 2005 is used for Email cleaning purposes in addition to the $EmailParser$ object. It provides pattern matching and replacement functions to remove duplicated content generated by $'reply'$ or $'forward'$ operations, and HTML mark-ups, such as $'\&gt'$, '>', etc. For the text normalisation phase, except for the stop words removal ($STOP\_WORD\_LIST$), which is implemented using Stanford

Core NLP toolkit, Apache Lucene OpenNLP toolkit is used for the rest of the steps, including tokenisation (*tokenise*()), spelling check (*SpellChecker*()) and stemming (*stem*(); Perkins, 2014). Stemming is used instead of lemmatisation in this circumstance for its simpler implementation in the Java programming language.

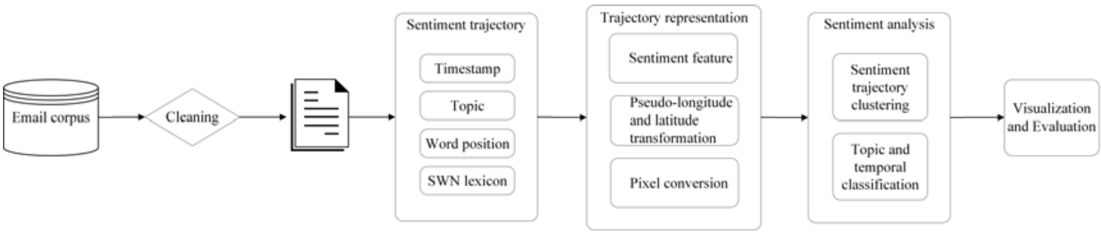

Email corpus
Cleaning
Sentiment trajectory
Timestamp
Topic
Word position
SWN lexicon
Trajectory representation
Sentiment feature
Pseudo-longitude and latitude transformation
Pixel conversion
Sentiment analysis
Sentiment trajectory clustering
Topic and temporal classification
Visualization and Evaluation

FIGURE 4.2: Overall framework for sequence-based document sentiment analysis of Email data.

4.4.1 Trajectory representation

To transform textual documents into sentiment trajectories, a three-phase trajectory representation approach was developed. First and foremost, sentiment features are extracted based on the SWN lexicon derived from documents. Then, sentiment features are transformed into pseudo-longitude and latitude representations using min-max normalisation. Finally, pseudo-longitude and latitude values are converted into scaled map pixels.

Sentiment feature The initial sentiment feature is generated from a refined and structured English sentiment lexicon known as SWN. As illustrated in Section 3.3.1, SWN 3.0 (Baccianella et al., 2010) has a good reputation in performing feature generation in sentiment analysis studies. It contains 147,305 sentiment phrases with six attributes uniquely identifying each item. Each sentiment phrase is identified by the combination of gloss and POS tags, as well as a positive and a negative score generated based on the frequency-weighted average of its relevant cognitive synonyms using a semi-supervised learning method. According to Baccianella et al. (2010), each sentiment phrase is valued by an overall objective score that adds the summation of positive and negative values into one.

Define $\mathcal{SWN}$ as a set of sentiment phrases $\mathcal{A} = \{a_1, a_2, \ldots, a_i\}$ with a set of sentiment values $\mathcal{V} = \{v_1, v_2, \ldots, v_i\}$. Each adjusted sentiment value of a sentiment phrase v_i is calculated by the following equation:

$$v_i = 1 - (PosValue_i + NegValue_i). \tag{4.1}$$

TABLE 4.1: Sample sentiment phrases generated from SWN 3.0.

Index	**Sentiment Phrase∗Tag**	**Sentiment Value**
22971	better∗v	0.5795
35472	flop∗v	-0.0454
54211	depress∗v	-0.0821
120842	judgment∗n	0.0482
13111	nonetheless∗r	-0.375
66857	kitchen∗n	0.0
74416	odious∗a	-0.25
147559	rapid∗a	0.1666

Table 4.1 presents some of the representative sentiment phrases with indexes and values generated from the SWN lexicon. Documents are transformed into vector space through the representation of the temporal position of a feature word and its corresponding value generated from the SWN lexicon.

Pseudo-longitude and latitude transformation As mentioned above, original trajectory mining is applied to data generated from natural movements and represented by geographic coordinates. Herein, attributes in sentiment feature representation are not applicable to the trajectory clustering algorithm. A normalisation method is implemented for converting position and sentiment values into pseudo-longitude values within the longitude degree of $[-180, 180]$ and pseudo-latitude within the latitude degree of $[-90, 90]$. The mathematical equation for the min-max normalisation is as follows:

$$Normalised(v_i) = \frac{v_i - Min(v_i)}{Max(v_i) - Min(v_i)} \times (max - min) + min, \tag{4.2}$$

where, v_i represents the i^{th} row in V and the value of the n^{th} row attributes in $\mathcal{A}$. $Min(v_i)$ represents the minimum value in attribute range $V[i]$, while $Max(v_i)$ represents the maximum value in attribute range $V[i]$.

Pixel conversion To visualise the trajectory clustering results as a scaled map, the $pixelconversion$ method is applied to pseudo-longitude and latitude using spherical Mercator Map Projection (Williams, 1995) formula for transforming geographic coordinates into map pixels. The mathematical formula for map coordinates $[Pixel_x, Pixel_y]$ and for reversed geographic coordinates $[Pseudo_{lon}, Pseudo_{lat}]$ are described as:

$$\begin{aligned} Pixel_x &= R * (\lambda - \lambda_0), \\ Pixel_y &= R * \lg[\tan(\frac{\pi}{4} + \frac{\phi}{2})]. \end{aligned} \tag{4.3}$$

$$\begin{aligned} Pseudo_{lon} &= \frac{Pixel_x}{R} * (\lambda - \lambda_0), \\ Pseudo_{lat} &= \frac{\arctan[10^{\frac{(\frac{\phi}{2} - Pixel_y)*2\pi}{R}} - \frac{\pi}{4}] * 2R}{\pi}. \end{aligned} \tag{4.4}$$

The value of R in Equation 4.3 and Equation 4.4 is related to the width and height of the scaled map. In addition, λ and ϕ represent the original longitude and latitude of the attribute, respectively, and λ_0 represents the natural longitude.

Apart from the sentiment value attribute, the numbers of sentiment phrases, categories and timestamp features are generated for categorical clustering and temporal classification purposes. Date attributes in the database are transformed into the format of standard UTC milliseconds. For instance, a date value ′2001-01-02′ is transformed into ′978393600000′. Meanwhile, category attributes are manually generated using a keyword search from *subject* attributes. A list of feature words is made for each category. For example, the category ′Business Operation′ includes the keywords ′contract′, ′project′, ′company′, etc. A subject containing ′training′ is grouped into ′Employee Training′. More specifically, the process includes two steps. Firstly, an LDA model is used to generate frequent words in subjects to create a categorical keyword list (Dredze et al., 2008). Then, similar keywords are assigned to different categories chosen from a category list developed by Goldstein et al. (2006). Each email is annotated based on the keyword(s) in its subject. If an email subject contains two keywords related to two different categories, its content is manually examined and a final label is determined accordingly. A chronological order of sentiment phrases is generated as attributes of each Email message with message id, $size$, $timestamp$ and $category$. Feature transformation is applied to the position and sentiment attributes for trajectory clustering. A sample Email message in vector representation is presented below, in which *position* represents the i^{th} place (for $i \geq 1$) of a word in each Email message and v represents the corresponding sentiment value of the word SWN lexicon returns. For instance, the word *'like'* is in the second place in a given sentence "I like to eat pizza." and pos-tagged *'like*$*v$*'* values 0.38 in SWN lexicon; hence, feature *'like'* is represented as *'2: 0.38'* in vector space:

$$< id \quad size \quad timestamp \quad category$$

$$[position_1 \quad v_1 \quad position_2 \quad v_2 \quad \ldots \quad position_n \quad v_n] > .$$

4.4.2 Sentiment trajectory clustering

The major component of the sentiment trajectory clustering process is the realisation of TRACLUS. It is a refined and widely adopted trajectory clustering algorithm for discovering subtrajectories in spatial databases (Lee et al., 2007). The fundamental logic of TRACLUS is based on Density-based Spatial Clustering of Applications with Noise(DBSCAN) with a refinement of the similarity measurement. By reducing multi-dimensional line segments into two-dimensional points, the TRACLUS algorithm clusters similar trajectories based on their common subtrajectories. In this study, the TRACLUS algorithm was applied to Email messages transformed into trajectory representations with categorical and temporal features. Since the original TRACLUS algorithm was developed for two-dimensional trajectory datasets with the additional attributes only including trajectory id and size, it was initially modified for storing temporal and categorical information from Email messages. Since the TRACLUS algorithm results in a set of clusters with representative trajectories in pixel coordinates that are to be represented by a TRACLUS embedded map, to better visualise the sentiment sequence in Email messages and sentiment polarity, a Pseudocode ($SentiPC$) for converting trajectories into sentiments is depicted below.

Pseudocode 2 SentiPC

```
Input: A collection of trajectory clusters $\mathcal{SO}$ with representative trajectories from
  TRACLUS results.
Output: Each $c_i$ in $\mathcal{SO}$ with a sentiment value $\mathcal{V}$ and sentiment polarity.
for each $c_i \in \mathcal{SO}$ do /* Polarity based on a 3 likert scale*/
    Get value coordinates $w_j$ from $c_i$;
    Normalise $w_j$ into sentiment value $v_i$;
    Compute the average $\mathcal{V}$ of each $c_i$;
    Convert $\mathcal{V}$ into polarity;
    Write the sentiment value $\mathcal{V}$ to file;
end for
```

As the TRACLUS algorithm performs best with geospatial coordinates, a transformation of the sentiment features into pixels is conducted. The above algorithm is applied for converting pixel coordinates into sentiment values by reversing the map projection equation using Equations 4.3 and 4.4 and normalisation using Equation 4.2. The two processes are denoted jointly in the above Pseudocode 2 as $Normalise$. As illustrated previously, the clustering results produced by the TRACLUS algorithm are depicted as a set of representative trajectories with pixel-converted feature values. In order to visualise the predicted sentiment polarity of each Email message from clustered trajectories, each feature in the pixel representation is to be transformed into its original format. Therefore, for each trajectory cluster *SO*, the value coordinate w_j is stored and converted into a sentiment value v_i by applying the reverse formula (Equation 4.4), used for converting pixel coordinates back into geographic coordinates, and Equation 4.2, used for normalising geographic coordinates into original sentiment values. The predicted sentiment polarity is calculated using the average value of the summation of $\mathcal{V}$ based on a three-point Likert scale. In addition, the categorical and temporal features of each Email message are retrieved and written to files for further grouping and classification purposes.

Following the TRACLUS clustering process, which is performed to better recognise sentiment patterns, a sentiment temporal clustering ($SentiTC$) algorithm is applied to the pruned dataset to group Email messages into temporal categories. The pseudocode for SentiTC algorithm is presented as follows.

Pseudocode 3 SentiTC

```
for each Email message $e_n \in \mathcal{E}$ do
    Create Email object;
    Store timestamps $\mathcal{T}$ from $e_n$ for clustering;
    Create Calendar object;
    Get week of year;
    Get day of week;
    Create group $G_w$ for week of year;
    Create group $G_d$ for day of week;
    for $G_d \in G_w$ do
        if $d \in (1, 5)$ then
            Create group $G_w(weekday)$;
            Put $e_n$ in $G_w(weekday)$;
        else
            Create group $G_w(weekend)$;
            Put $e_n$ in $G_w(weekend)$;
        end if
    end for
end for
```

The above algorithm clusters email messages based on their temporal distribution by storing the temporal information (in milliseconds) of each email message and comparing it with *Calendar* (a predefined object containing dates in milliseconds in Java programming language). Preliminarily, an *Email* is created for storing the attributes of each Email message generated from a MySQL database[11], including message id, subject, timestamp and sentiment features. After creating the predefined *Calendar* object, attribute timestamp *T* of each Email message is converted into the week of the year and the day of the week. An Email message is classified into weekday groups if its matching day is between one and five, otherwise, it is classified into weekend groups. The process is repeated until all Email messages are stored in the corresponding calendar group. The final clustering results are represented in week of the year and day of the week form by transforming milliseconds into dates.

[11] https://www.mysql.com/

4.5 Empirical results and discussion

In this study, two sets of experiments were undertaken to evaluate the feasibility and performance of the proposed technique, denoted as Senti$\mathcal{TRACLUS}$. For qualitative analysis, a collection of Email describing real-life activities was extracted from the Enron Email corpus for conducting empirical experiments. The sentiment trajectory clustering results, in terms of categorical and temporal classifications and sentiment sequence patterns in Email messages, are presented as graphs and tables. For quantitative analysis, two labelled datasets with one manually labelleld Email dataset and a subset of Amazon product review dataset were generated for the pilot experiments and three benchmark datasets discussed in Section 3.2 were utilised for the main experiments. Details were elaborated in the following sections.

4.5.1 Dataset

The main experiments of this research are undertaken on a subcollection of the large and well-developed Enron Email corpus, which contains $7,507$ Email messages exchanged between 1^{st} and 31^{st} of January 2001. Since the main purpose of the research was sentiment clustering, a structured database version[12] of the original dataset was implemented for ease of data cleaning. Experiments were conducted using the Java programming language with Eclipse IDE[13]. Fifteen manually selected categorical phrases, such as 'Company Project', 'Logistic Issue', etc., were used for categorising the Email messages.

As discussed in Section 3.3.1, Email messages were cleaned and normalised using the NLP toolkit (Manning et al., 2014; McCandless et al., 2010). The initial sentiment features were selected using a pruned SWN lexicon (containing $7,077$ words and phrases) to increase the processing speed. The pruning process was done to create a dictionary of the entire corpus and to run the corpus in alignment with the entire SWN lexicon, which originally contained $147,305$ words. A pruned lexicon, including $7,077$ words with corresponding lexical categories and sentiment

[12] http://www.ah-ruhe.de/pub/R/data/$enron - mysqldump\_v5$.sql.gz
[13] https://www.eclipse.org/

values, was stored as a plain text file. To address the details of each step in the feature transformation process, an example is presented below.

Given an Email message with id 22681 in its original format:

```
Mid: 22681
Date: 2001-01-12
Subject: Re: NG Gas Deferred
Content: My social schedule is not the problem...that one
is pretty clear.  But I will look at my work schedule and
have my people call you.  As far as I know any day next
week should be good for me.  Just give me a call.
```

According to the feature extraction process described above, Email message 22681 was converted into vectors using the SWN lexicon and spatial information. For instance, in the phrase 'My social schedule is not the problem...', the words 'social' and 'problem' return '$2 : -0.009$' and '$7 : -0.386$', respectively. The feature-represented Email is displayed as follows:

Mid: 22681

⇒ *22681, 8 (size)*

Date: 2001-01-12

⇒ *979257600000*

Subject: Re: NG Gas Deferred

⇒ *Business Investment*

Content: My social schedule is not the problem...that one is pretty clear. But I will look at my work schedule and have my people call you. As far as I know any day next week should be good for me. Just give me a call.

⇒ *2: -0.009 7: -0.386 30: 0.014 35: 0.038 36: -0.023 37: 0 44: 0.07 48: 0.02.*

Following feature extraction, the outcomes of pseudo-longitude and latitude representation and pixel conversion were calculated using the equations given previously, as below:

```
Pseudo-longitude and latitude representation: <22681, 8,
979257600000, Business Investment, [-179.97: -6.82
-179.79: -43.09 -178.99: -4.69 -178.81: -2.36 -178.78:
-8.18 -178.74: -6 -178.5: 0.69 -178.36: -4.11 ]>

Pixel conversion: <22681, 8, 979257600000, Business
Investment, [0.1: 472.77 0.63: 609.48 3.04: 465.64 3.56:
457.86 3.67: 477.37 3.77: 470.04 4.51: 447.69 4.93:
463.72 ]>
```

4.5.2 Email distribution on Senti$\mathcal{TRACLUS}$ clustering results

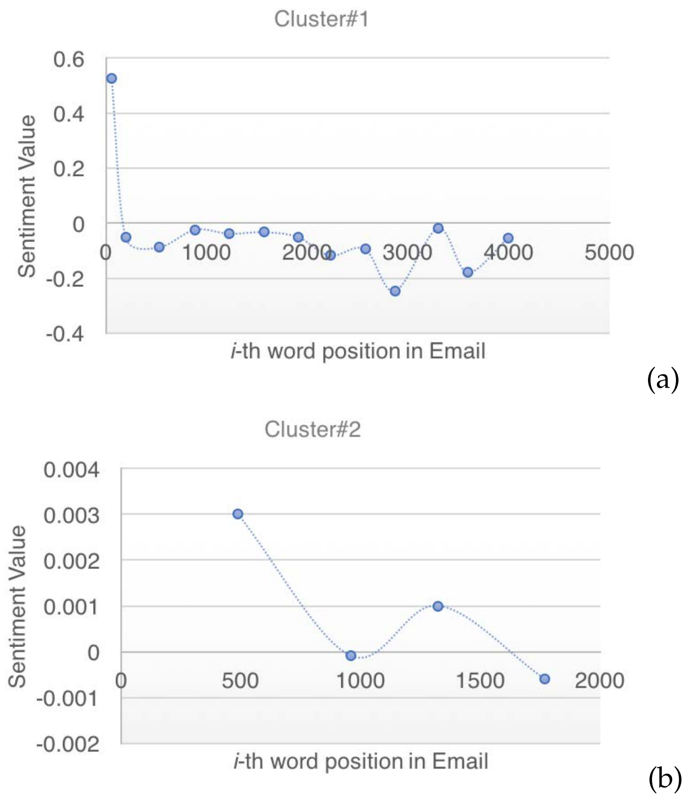
On the basis of the Senti$\mathcal{TRACLUS}$ clustering results, $3,128$ of the $7,077$ Email messages were clustered into two distinctive trajectories, which is about a 44.2% use ratio. Therefore, using Email data for sentiment analysis is a real challenge as it contains massive amounts of noisy data. Empirical results were obtained after applying the Senti$\mathcal{TRACLUS}$ algorithm, which requires two input parameters: $minLns$ and eps. Based on Lee et al. (2007)'s experiments on parameter selection, clustering results can vary enormously. As the algorithm utilised in this study implements functions for automatic detection of suitable parameters, the results were generated with a $minLns$ value of 5 and eps value of 29. Based on the implemented algorithm (Lee et al., 2007), the automatic selection of the eps parameter started with a value 20 and looped until reaching 40. The $minLns$ parameter was determined by rounding the value of the size of all trajectories and dividing it by the number of line segments in a cluster. Looping of eps will cease when the value of entropy is minimised. The mathematical formula for calculating the entropy $\mathcal{E}_{(c)}$ is:

$$\mathcal{E}_{(c)} = \sum_{i=1}^{n} eps[i] * \log_2 \frac{1}{eps[i]}, \quad i \in (1, n). \tag{4.5}$$

In Equation 4.5, $eps[i]$ is a function that calculates the density of the i^{th} cluster in contrast to other clusters, while n represents the total number of line segments.

Clustering results from Senti$\mathcal{TRACLUS}$ are presented by geographic coordinates that do not imply either sentiment polarity or a sentiment sequence. Therefore, features in pixels are converted back into sentiment values based on the same process discussed in Section 4.4.1. A three-class result, comprising Positive ($\mathcal{P}$), Neutral (Neu), and Negative ($\mathcal{N}$), is applied to determine the sentiment polarity of each cluster based on the final sentiment value calculated. Clusters with a final sentiment value above 0 are grouped into the class of $\mathcal{P}$, below 0 are grouped into the class of $\mathcal{N}$, and Emails without clusters are grouped into the class of Neu. The final output of the aforementioned process describes the general trend in the sentiments expressed in the entire dataset, assigns sentiment polarity to clustered Email messages, and identifies outliers and noise in the dataset. Additionally, the general trend in sentiments expressed in the Enron Email dataset is the essential result, which is represented by the sentiment sequence of clusters generated by the Senti$\mathcal{TRACLUS}$ algorithm. Since it is difficult to interpret this kind of trend from the original sentiment sequence, categorical and temporal classification was conducted to further justify the importance of the sentiment sequence and to visualise interesting patterns.

Figure 4.3 illustrates two representative trajectories from Senti$\mathcal{TRACLUS}$ displayed using sentiment values and i^{th} word positions in the Email. Though the final sentiment polarity is -0.47 for the first group and 0.002 for the second, both clusters show a sentiment sequence from positive to negative indicating a frequent pattern of the way most Email messages are addressed. Concretely, the final sentiment polarity is computed using the average value of each cluster. Take the second cluster as an example. The final polarity value of 0.002 was calculated using four sentiment values divided by their sum: 0.003, $-9.53e - 5$, 0.001, and $-6.21e - 4$. This value stands for the overall sentiment polarity of a cluster that is representative of the general trend of all sentiment trajectories in a cluster. It is not the exact sentiment value of the text under investigation; however, a sentiment sequence with a sample Email message (see Table 4.2) is presented to visualise the details of sentiment flow in each cluster. Through the implementation of a trajectory clustering approach, the general sentiment flow of text under



(a)

(b)

FIGURE 4.3: Two frequent sentiment sequences identified from Senti$\mathcal{TRACLUS}$ algorithm.

investigation, as well as the classification of the sentiment polarity (a three-point Likert scale is used in this study) of each Email message can be evaluated.

To justify the importance of sentiment sequences within documents and improve visualisation of the sentiment trajectory clustering results, Email messages are grouped into categories after temporal classification. Temporal classification is conducted on two trajectory clusters individually, which results in four entire weeks being identified, with 20 weekdays and 4 weekends, and 1 incomplete week with 3 weekdays. Figure 4.4 illustrates the overall Email distribution from categorical and temporal perspectives for two clusters generated by the Senti$\mathcal{TRACLUS}$ algorithm. The first group, clustered into positive, contains $3,101$ Emails, whereas the second group, clustered into a slight negative, contains only 25 Emails.

Since the second group contains 25 Email messages, which is not enough for temporal classification, Figure 4.4 only displays Email distributions of the categorical and temporal categories of cluster group one. Nevertheless, the

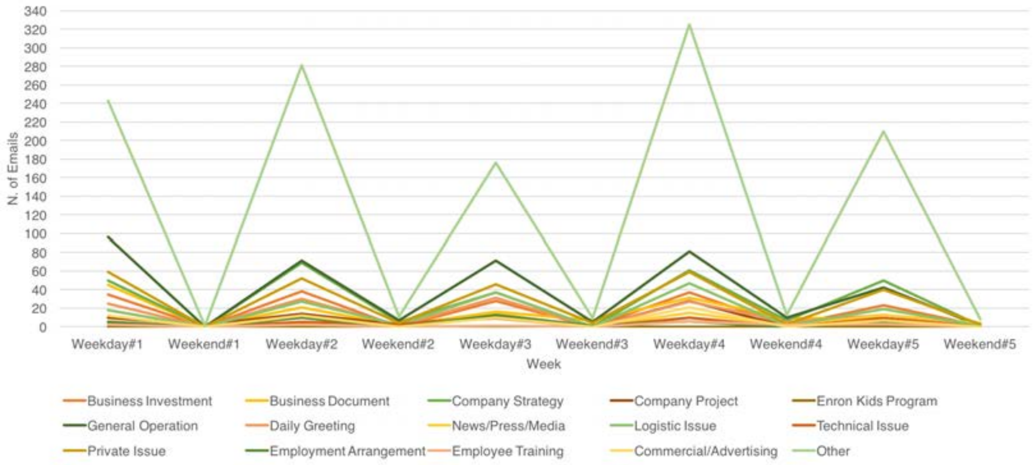


FIGURE 4.4: Email distribution from categorical and temporal perspectives for Senti$\mathcal{TRACLUS}$ cluster group 1.

categorical clustering result for sentiment trajectory group 2 is four for ’Business Investment’, three for ’Business Document’, four for ’Company Strategy’, three for ’Company Project’, two for ’General Operation’, one for ’Daily Greeting’, three for ’Private Issue’, one for ’Employment Arrangement’ and three for ’Other’. In accordance with the categorical and temporal grouping results, most of the Emails are communicated on weekdays during general business hours. As for the distribution of categories, a similar weekly distribution of Emails was obtained. For instance, categories such as ’Private Issue’, ’General Operation’ and ’Company Strategy’ in cluster group 1, have an evenly distributed quantity for each week. This suggests that as a routine communication tool, Emails in these categories not only have the same sentiment polarity but also share a similar way of expressing the sentiments involved.

4.5.3 Sentiment sequence within Email messages

To gain insight into the influence of sentiment sequences within documents, Table 4.2 presents some of the indicative results, including sentiment variation in values within Email messages from trajectory clusters, as well as detailed Email messages with feature words and their corresponding sentiment values.

In Table 4.2, the results indicate that a single sentiment polarity is insufficient for describing the sentiment(s) involved in Emails based on the features generated for

TABLE 4.2: Sample Email sentiment clustering results for Week1.

Sentiment Sequence	Day#	Category	Email Message
0.526 − > -0.052 − > -0.088 − > -0.024 − > -0.038 − > -0.033 − > -0.05 − > -0.118 − > -0.094 − > -0.247	1	Employment Arrangement	⟨78070: [gas: 0, pass: -0.029 current: 0, want: 0.055, send: 0.013, exec: 0, make: 0.021, sure: -0.002, david: 0, call: 0.02, holiday: 0, forward: 0.109, pm: 0, subject: 0, like: 0.38, offer: 0, behalf: 0.042, attach: 0.055, group: 0, talk: -0.066, john: 0, mike: 0, global: -0.208, market: 0, need: -0.045, inform: 0, let: -0.02, pruner: 0] ⟩
− > -0.019 − > -0.178 − > -0.054 Polarity: -0.47	2	Employment Arrangement	⟨306350: [work: 0.016, like: 0.38, resent: -0.542, accept: 0.12, offer: 0, hr: 0, recommend: 0.136, subject: 0, mr: 0, thank: 0, refer: 0.008, let: -0.02, name: 0.015, person: 0, vice: 0.167, houston: 0, print: 0, now: 0.019, http: 0, see: 0.027, attach: 0.055, part: 0, share: 0, holiday: 0, photo: 0]⟩
	4	Business Document	⟨376218: [day: 0.038, roll: -0.003, year: 0, error: -0.355, previous: -0.114, number: -0.118]⟩ ⟨106021: [attach: 0.055, file: 0, contain: 0.068, reflect: 0.11, fact: 0.045, current: 0, custom: 0.21, garden: 0, paper: 0, default: -0.1, limit: -0.068, inform: 0, higher: 0.208, thank: 0]⟩

each Email message. Two categorical groups, 'Employment Arrangement' and 'Business Document', were selected from the first four days in week one. The results in the table justify that the same categorical group in different days shares a similar sentiment variation based on feature values. Additionally, the trajectory representation generated by the Senti$\mathcal{TRACLUS}$ algorithm is able to model the general sentiment sequence within Email messages that are clustered together. Though the Senti$\mathcal{TRACLUS}$ algorithm cannot fully manage the sentiment sequence of each Email message, the results demonstrate its ability to cluster messages with similar sentiment sequences.

To further justify the indispensability of sentiment sequence detection, Table 4.3 compares the clustering results for Emails in the same category from two cluster groups.

TABLE 4.3: Comparative results between two clusters with Emails in the same category from Week#3 Day#4.

Cluster ID	Sentiment Sequence	Category	Email Message
1	Positive − > Negative − > Negative − > Negative − > Negative − > Negative − > Negative − > Negative − > Negative − > Negative − > Negative − > Negative − > Negative	Company Strategy	⟨236969: [base: -0.014, draft: 0, period: 0.024, time: 0.121, mutual: 0, adjust: 0.068, current: 0, fee: 0.125, answer: 0, let: -0.02, talk: -0.066, forward: 0.109, chip: -0.011, van: 0, glenn: 0, subject: 0, worst: -0.5, case: 0.005, seem: 0.06, suspend: -0.196, steam: 0, event: 0, obtain: 0.068, transport: 0.018, site: 0, scope: 0, agreement: 0.017]⟩
2	Positive − > Negative − > Positive − > Negative	Company Strategy	⟨211258: [particular: 0.047, counsel: 0, concern: 0.018, understand: 0.175, seem: 0.06, discuss: 0, make: 0.021, sure: -0.002, attorney: 0, scott: 0, think: 0.064, send: 0.013, short: -0.19, expect: 0.004, similar: 0.103, dip: -0.012, water: 0.009, case: 0.005, need: -0.045, forward: 0.109, pm: 0, subject: 0, mon: 0, jan: 0, bob: 0, rick: 0, paul: -0.042, tom: 0.034,white: 0.028, data: 0, request: 0.045, reliant: 0, inform: 0, rule: 0.084, regard: 0.102, thank: 0]⟩

On the basis of the sentiment sequence identified by the trajectory representatives (see Figure 4.3), cluster group 1 fluctuates more than cluster group two in terms of sentiment variation within Email messages. Table 4.3 displays corresponding Emails from the same category on the same day represented by feature values selected from two clusters, respectively. The coherence among the difference between sentiment values within Email messages in two clusters further justifies the need to consider sentiment sequences when determining the similarity among documents. It also proves the feasibility of applying trajectory clustering techniques to the identification of detailed sentiment sequence patterns within documents.

To support the discussion above, an example is provided. The following Email with *id* 2897 is classified as negative. It was observed from the original Email that the most prominent sentiment of this message is expressed through the phrase 'object to'; however, positive sentiments involved in phrases, such as 'advantage' and 'inclined to', also exist. Traditional feature-based techniques ignore this kind of aspect, whereas sequence-based approaches consider it as a major attribute.

```
Email in feature format: <2897, 23, 979689600000,
Business Investment, [like#v: 0.38 reaction#n: 0.024
notion#n: 0.02 gas#n: 0 revisit#v: 0 code#n: 0 refrain#n:
0 need#v: -0.045 plant#n: -0.06 concern#n: 0.018 exist#v:
0.042 rule#n: 0.084 treatment#n: -0.18 rate#n: 0
project#n: 0 plan#n: 0 object#v: -0.042 market#n: 0
chang#n: 0 let#v: -0.02 thank#v: 0]>
```

```
Email in original format: <2897, [Joe and Christi, I
would like your reaction to this notion.  On the gas
side, FERC is revisiting the marketing affiliate rule and
code of conduct.  One repeated refrain coming from
non-affiliated marketers is that the definition of
marketing affiliate needs to refined to include electric
generators/merchant plants affiliated with the pipeline.
There is a concern that since they are not covered by the
existing rule, they get preferential treatment (timing,
info, rates) that gives an advantage to the affiliate s
projects over those planned by third parties.  Would we
object to changing the definition so that these entities
are considered marketing affiliates?  I would be inclined
to go along with the change, if it doesn’t hurt us.  Let
me know. Thanks.]>
```

Additionally, in terms of efficiency, the computational complexity of the proposed method is considerable. The approach is generally composed of a cleaning phase, a temporal classification phase and a trajectory clustering phase.

The brief time complexity is $O(n^m)$, $O(n)$ and $O(t\log t)$ respectively, where n represents the number of Email messages, m represents the number of words in the pruned SWN lexicon and t represents the number of trajectories. The general space complexity of the trajectory clustering process is $O(s)$, where s represents the total number of line segments.

4.5.4 A case study with labelled datasets

The feasibility of using the proposed approach for discovering sentiment sequences within documents was demonstrated through the qualitative evaluation above. However, its classification performance was not quantitatively validated. Two levels of experiment (pilot and main) were undertaken using different sets of labelled datasets. As discussed previously, the major purpose of the case study was to quantitatively validate the classification accuracy of the proposed method. Therefore, no categorical or temporal classification was involved due to the limitlessness of the features collected in the dataset. To compute the evaluation matrix for the clustering results, the same rule as defined in Section 4.5.2 was applied in the case study. Once the Senti$\mathcal{TRACLUS}$ results were determined by minimising the value of entropy as defined in Equation 4.5, all cluster groups were further categorised into three classes based on the final sentiment value assigned to each cluster. I hereby describe the datasets used for the pilot experiment in detail and present a summary of the class distribution of the datasets used for the main experiment.

- **Pilot datasets:** Two datasets from different sources were utilised for flexibility evaluation. One was a manually labelled Email dataset, denoted as PA (pilot), containing 111 messages generated from a personal Gmail archive. The prelabelled Email distribution contained 30 positive, 73 neutral and 8 negative Emails, respectively. The manual annotation process was similar to that used for the PA dataset discussed in Section 3.2.3. The other set was a subset of Amazon product review data, denoted as Amazon Review, containing $5,000$ reviews (Wang et al., 2010) with ratings generated as for the second test dataset. The Amazon Review dataset was used as a comparative case to validate the validate the hypothesis that the proposed method performs particularly better on Email data compared to other algorithms. To convert ratings ranging from 1 to 5 into a three-point Likert scale, a rule was applied to the original dataset: ratings greater than 3 are negative, those less than 3 are positive, and equal to 3 are neutral. As a result, the prelabelled review distribution contained 3567 positive, 386 neutral and 1047 negative individual reviews.

- **Main datasets:** Three benchmark datasets were utilised as the sources for the experiments. For these, the collection and label conversion processes are described in Section 3.2 and the class distributions are presented in Table 3.1.

4.5.4.1 Experimental settings

Apart from the proposed method, the other four algorithms were utilised in both the pilot and main experiments to comparatively evaluate the performances. Brief descriptions of the four selected algorithms are provided below:

- Baseline: The baseline method is a purely lexicon-based technique using the SWN lexicon. The polarity of each Email message is determined by the sum of its feature values. 'Positive' is assigned if the sum is greater than 0, 'Neutral' if it is equal to 0 and 'Negative' otherwise.
- k-Means (Liu and Lee, 2018): An unsupervised clustering algorithm with high efficiency. Liu and Lee (2015) suggested that a combination of k-Means and SVM performs better than other clustering and classification algorithm combinations. In this study, since the dataset was prelabelled, k-Means was directly implemented to generate sentiment polarities. Euclidean distances were applied as distance measurements. Specifically, when implementing the k-means algorithm, the number of clusters is set to three to be consistent with the three-point Likert scale evaluation standard. To minimise the influence of local optimal, three initial centroids representing positive, neutral and negative, respectively, were chosen from the dataset. The final result of the

evaluation metrics is determined when the Squared Sum Error(SSE) reaches its lowest value.

- SVM (Chang and Lin, 2011): Pre-developed library LibSVM (Chang and Lin, 2011) was implemented in this research. It is a linear classifier that converts the normalised feature representation into vectors with the same dimensionality within the feature space using a linear predictor (hyperplane) approach. In this paper, C-SVM (Chang and Lin, 2011) was implemented for multi-class classification. Since the experimental datasets have close observation and feature values, both linear and Gaussian kernels were trialled with the SVM algorithm, with the linear kernel performing better. In accordance with the standard procedure for linear kernels, parameter $\mathcal{C}$ was under standardised test with multiple values ranging from $[2^{-5}, 2^{10}]$. For both cases, parameter $\mathcal{C}$ of 1 was utilised to generate the highest accuracy rate;
- Multi-Layer Perceptron(MLP) (Gardner and Dorling, 1998): As a representation of neural network models, MLP in Waikato Environment for Knowledge Analysis(WEKA) (Hall et al., 2009) 2009), which incorporates the core concept of backpropagation, was utilised with a batch size of 100 and three hidden layers. Predictive models created by SVM and MLP were evaluated under 10-fold cross-validation considering the sizes of the two datasets.

A standard confusion matrix consisting of *Precision*, *Recall* and *F-measure* is inadequate for evaluating three-class classifications. Instead, the performance of all algorithms was quantitatively measured in terms of *Accuracy* and *Root Mean Squared Error(RMSE)*, where Accuracy measures the percentage of correctly classified instances over the total and RMSE is calculated as the square root of the prediction error. Results with higher accuracy and lower RMSE are preferred. The mathematical formulas for these evaluation criteria are:

$$\begin{aligned} Accuracy &= \frac{\sum_{i=1}^{n} observe_i - positive_i}{n}, \\ RMSE &= \sqrt{\frac{\sum_{i=1}^{n}(predict_i - observe_i)^2}{n}}. \end{aligned} \tag{4.6}$$

Macro F-measure(Macro-F) is the average of each class's *F-measure* value and *Mean Absolute Error(MAE)* measures the average prediction error. These were used in the pilot experiment to provide more evidence of the performance of the proposed method with different data sources.

$$\begin{aligned} Macro-F &= \frac{\sum_{j=1}^{3} \frac{precision_j \times recall_j}{precision_j + recall_j}}{3}, \\ MAE &= \frac{\sum_{i=1}^{n} |predict_i - observe_i|}{n}. \end{aligned} \tag{4.7}$$

4.5.4.2 Classification results

Table 4.4 shows the performance evaluation of the five algorithms on the two pilot datasets. Senti$\mathcal{TRACLUS}$ had the highest Macro-F rate of 69.1% and the lowest MAE rate of 29.7% on the pilot personal Email archive, and an accuracy rate of 71%, which is slightly lower than the highest accuracy rate, on the review dataset. These experimental results demonstrate the improved classification accuracy of the proposed method to some extent.

TABLE 4.4: Pilot experiment results comparing the performance of the proposed algorithm and other classifiers. Bold texts indicate results to be highlighted.

Dataset / Classifier	**PA (pilot)**				**Amazon Review**			
	Accuracy	Macro-F	MAE	RMSE	Accuracy	Macro-F	MAE	RMSE
Baseline	0.351	0.341	0.703	0.900	0.578	0.365	0.763	1.201
k-Means	0.621	0.256	0.378	0.615	0.693	0.206	0.521	0.978
SVM(Chang and Lin, 2011)	0.657	0.264	0.342	0.585	**0.714**	0.280	**0.190**	0.648
MLP(Gardner and Dorling, 1998)	0.549	0.352	0.321	0.507	0.683	0.387	0.223	0.941
SentiTRACLUS	**0.729**	0.691	**0.297**	0.592	0.710	0.278	0.499	0.958

Figure 4.5 further compares the overall classification performance of the proposed method and other methods on both the pilot and main experimental

datasets. In terms of accuracy, Senti$\mathcal{TRACLUS}$ obtained the highest rates of 72.9% and 79.3% on two personal Email archives, respectively, and rates that were slightly less than the highest rates (71% compared to 71.4% and 57.9% compared to 58.2%, respectively) on the Amazon Review and EnronFFP datasets. Though less competitive in terms of error analysis than other supervised learning algorithms, Senti$\mathcal{TRACLUS}$ was more effective with Email datasets, which is the main priority of this research. Moreover, it is computationally economical, as described previously.

4.6 Conclusions

In this chapter, an unsupervised sentiment sequence clustering method, Senti$\mathcal{TRACLUS}$ was proposed to discover sentiment sequence patterns in Email data and classify sentiments in a novel sequential way. The proposed method can be applied to an Email system to analyse the sentiment patterns of archived Emails and detect the sentiment polarities of incoming messages to prioritise workloads, assess business risks or manage customer relationships. By transforming sentiment features into a trajectory representation, a revised TRACLUS algorithm with a combination of sentiment temporal clustering can be implemented to discover sentiment flows in Email messages with categorical and temporal distributions. The results obtained from empirical experiments on a subset of the Enron Email corpus reflect a few patterns that can be summarised in three aspects. First, Email datasets contain much noise, which increases the difficulty of sentiment classification using traditional document-level techniques. Second, the consistency of the trajectory clusters and sentiment features generated prove the feasibility of applying the Senti$\mathcal{TRACLUS}$ algorithm to sentiment sequence clustering with Email data. Finally, the insights gained into the detailed sentiment sequences existing within Email messages, and comparisons of the clustering results, prove that sequences influence sentiment determination and that it is important to consider sentiment sequences in the process of sentiment clustering.

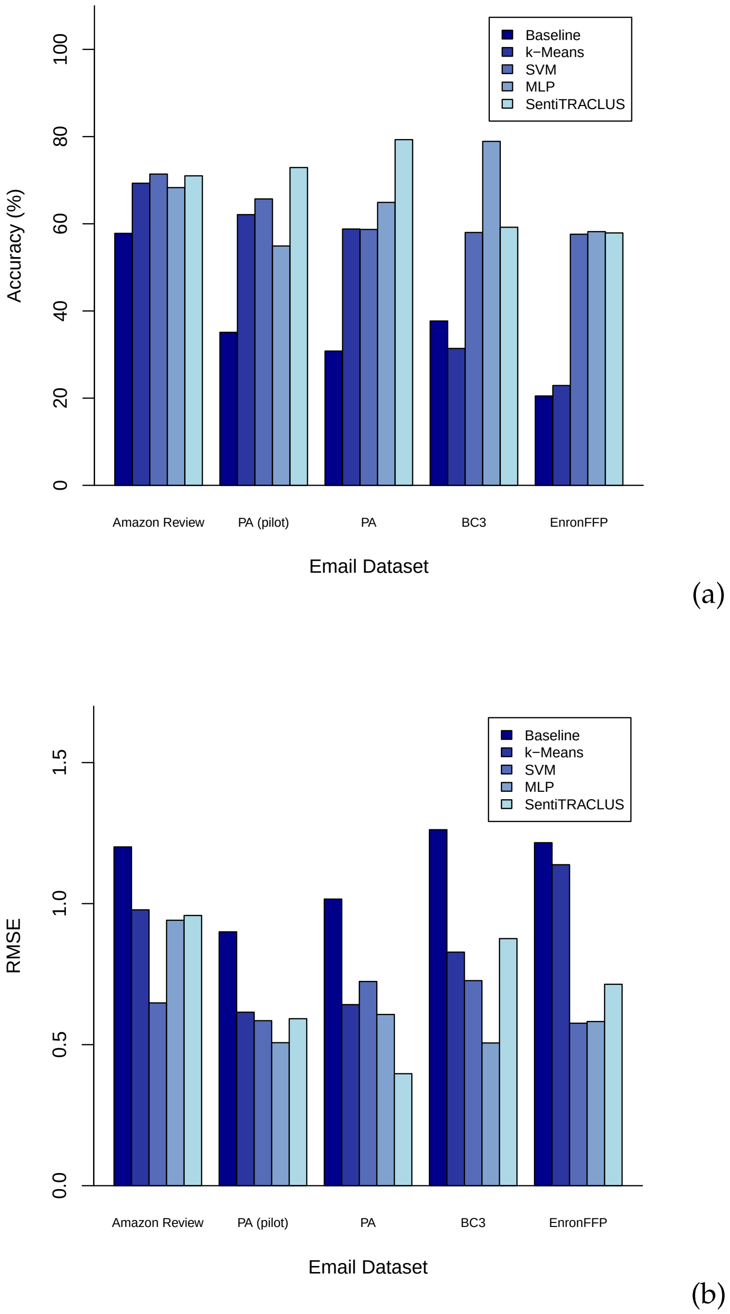


FIGURE 4.5: Overall evaluation of comparative performance with the pilot and main experiments.

Although minimal quantitative analysis was conducted, the results demonstrate the advantages of the proposed sequence-based sentiment clustering method Senti$\mathcal{TRACLUS}$, not only in discovering sentiment sequences within documents but also in accurately classifying sentiments. The technique proposed in this study sets a new direction for sequence-based sentiment clustering. For unstructured and lengthy text data, such as Email, this novel prospective contributes to a deeper understanding of sentiments expressed within the documents and an improvement of sentiment classification accuracy. Unlike state-of-the-art sentence-level sentiment analysis techniques, which aim to improve classification accuracy, the proposed trajectory clustering algorithm was refined and adopted to gain more insights into sentimental variations among single documents and an entire corpus. It is obvious that considering sentiment sequences during the feature extraction process makes a difference in sentiment analysis tasks. Nevertheless, additional comprehensive studies on properly capturing sentiment sequences to improve classification accuracy are required.

5 Sentiment classification of Email data using a sequence-encoded CNN model[14]

In this chapter, I describe the study of a dependency graph-based position encoding technique enhanced with weighted sentiment features and incorporate it into the feature representation process for Email document sentiment classification. Section 5.2 reviews existing studies on document-level sentiment analysis that use deep learning techniques. Section 5.3 describes the proposed sequence-encoded neural classification method. Section 5.4 summarises the main findings of the quantitative evaluation of the proposed method. Section 5.5 concludes the study and highlights its contributions. Figure 5.1 illustrates the topics (in blue and bold) of the Email sentiment analysis framework that will be covered in-depth in this chapter.

5.1 Introduction

Chapter 4 presented a sentiment sequence clustering study that demonstrated that Email data contains sentiment sequence features. Although the trajectory clustering approach was efficient in discovering sentiment sequence patterns, it was less effective in classifying sentiment polarities. Motivated by these observations, this study aims to develop a robust and effective sequence-encoded sentiment classification technique for Email data.

[14] This chapter is written based on the following paper 'Liu, S., & Lee, I. Sequence encoding incorporated CNN model for Email document sentiment classification.' submitted to Applied Soft Computing.

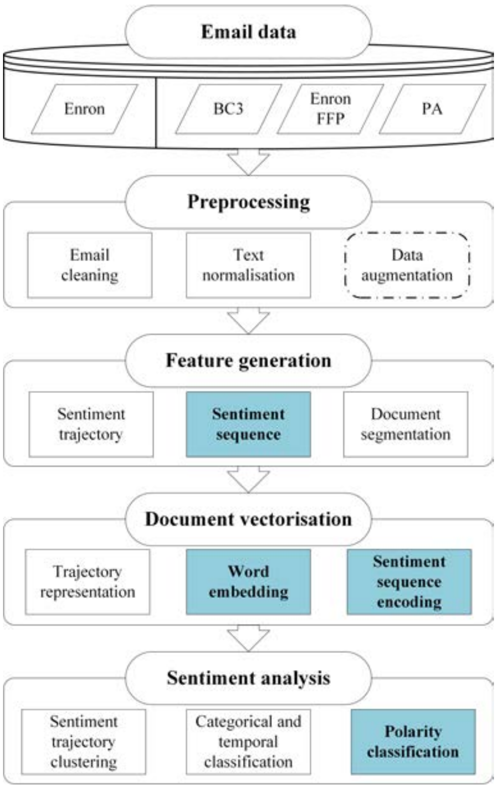


FIGURE 5.1: Overall structure of the Email document sentiment analysis framework, with the specific topics covered in this chapter highlighted in blue and bold.

With recent advances in computing power, more studies are applying deep learning techniques built on neural network models to sentiment analysis tasks (Zhang et al., 2018). The outcomes of these studies prove the robust and effective performance of neural network-based techniques on text classification tasks (Chen et al., 2016; Majumder et al., 2017; Tang et al., 2015a,b). Sequence encoding is a technique developed for modelling the textual structures and discourse relations of words and sentences within a document. Studies indicate that sequence encoding-incorporated methods are effective in various types of text mining tasks, such as question-and-answer problems and cause-and-effect detection (Sukhbaatar et al., 2015; Yang et al., 2016a). Hence, it is expected that a revised deep learning model with position encoding and enhanced sentiment features will be capable of handling lengthy Email issues, and of capturing indirect relations and emotions in Email messages.

This chapter proposes a deep CNN-based model with sentiment sequence encoding that can more accurately classify the sentiments in Email documents. The main contributions of this chapter are:

- introducing two types of sequence encoding methods, using discourse weighting and LSTM network model;
- proposing a dependency graph-based position encoding approach for capturing relational and structural features;
- incorporating sequence encoding with sentiment lexical features into a CNN model for better feature representation and improved classification performance;
- evaluating the sentiment sequence encoding-incorporated CNN model in terms of classification accuracy, and comparison with lexicon-based unsupervised learning and supervised learning approaches;
- examining the effectiveness and influence of the revised data augmentation technique with representative algorithms of three categories: unsupervised learning techniques, supervised learning techniques and neural network models; and
- experimenting with the proposed deep learning model in various conditional settings to investigate the effects of: text cleaning, position features, and sentiment sequence encoding techniques.

5.2 Related Work

As observed in the literature review of Sections 2.2.2 and 2.3.2.2, a distinctive gap was identified between existing techniques of document-level sentiment analysis and Email sentiment analysis that uses deep learning approaches. With the development of various deep learning techniques and the refinement of machine learning techniques, it is necessary to test the efficiency and effectiveness of state-of-the-art techniques on Email sentiment analysis and improve the performance by considering the unique characteristics of Email data.

Feature modelling is an indispensable component of document sentiment classification. With traditional feature modelling techniques, such as BoWs or TF-IDF, document-level sentiment classification suffers from inferior classification accuracy and the inability to model intrinsic relations (Bhatia et al., 2015; Chen et al., 2016; Majumder et al., 2017; Tang et al., 2015a).

Unlike traditional machine learning classifiers that require manual or semi-supervised selection of input features, deep learning models take advantage of automatic feature extraction. For instance, Bhatia et al. (2015) presented a rhetorical structure theory-based neural network to improve lexicon-based sentiment analysis. Chen et al. (2016) proposed a hierarchical LSTM model with user and product attention to incorporate user preferences and product characteristics in document-level sentiment analysis. Majumder et al. (2017) developed a technique that utilises a CNN model to extract features from documents for personality detection and document modelling. In summary, with feature modelling techniques such as word embedding, deep learning has improved the performance of document sentiment classification.

However, the approaches developed in previous studies have been demonstrated as effective and efficient only with short review documents with low variance in their length distribution (e.g., similar numbers of words within sentences and similar numbers of sentences within a document). Note that Email has inherent special characteristics such as high variance in length, lengthy replies, high duplication, anomalies, and indirect relationships. Hence, it is inappropriate to apply these pre-developed deep learning models directly to Email, since they are designed to handle short reviews with similar lengths but without high duplication and anomalies.

5.3 Proposed sequence-encoded neural classification method

In this section, the general workflow of the proposed method is discussed. It consists of four major phases: Email cleaning, feature generation, document vectorisation and neural sentiment classification. Briefly, an enhanced position feature generation method with sentiment lexical features is introduced and incorporated into a sentiment sequence-encoded CNN model for document vectorisation. A revised and refined neural network architecture is developed based on deep CNN, which uses combined word embedding and sentiment sequence encoding as an input layer for Email document sentiment classification. Figure 5.2 illustrates the general workflow of the proposed method, including Email cleaning, feature generation, document vectorisation and neural sentiment classification. It also provides an overview of the neural model, which uses word embedding and sentiment sequence encoding as input for a CNN-based network.

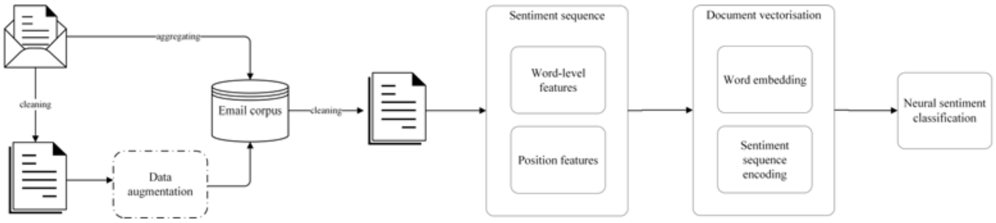


FIGURE 5.2: Overall framework of the proposed sentiment sequence encoding-incorporated CNN model for Email document sentiment classification.

As mentioned in Chapter 2, neural network models are sensitive to the scale of the training data and the distribution of class labels. Hence, a data augmentation phase using a random word replacement is implemented to minimise the influence of imbalanced class distribution, to tune model parameters and to control model fitting. The detailed data augmentation process and sample outputs are presented in Section 3.3.1. Details of each of the other steps are described in the following subsections.

In this approach, a thorough Email cleaning phase is conducted before converting raw Email messages into numeric feature vectors. Unlike datasets developed for general sentiment analysis, the Email documents used for empirical experimentation in this study contained unnecessary information, such as mark-ups and signature blocks, which may negatively influence the performance of Email sentiment analysis. In brief, the entire phase is divided into an Email-specific cleaning process and a standard text normalisation process, with the details discussed in Section 3.3.2.

In this study, the steps used for Email cleaning and text normalisation mainly follow those illustrated in Pseudocode 1. To highlight, a Python module re (Goyvaerts and Levithan, 2012), which provides regular expression operations to generate and remove duplicated content beginning with the keyword $'original'$ in $'reply'$ or $'forward'$ Emails, as well as unstructured expressions and mark-ups, such as $'\&gt'$, $'- - -'$ and etc. As for the standard text normalisation process, no stop word removal is executed, to maintain the integrity of sentences and syntactic relations among phases in Emails. The Python $nltk$ toolkit is utilised to implement tokenisation ($tokenize$() and $sent\_tokenize$()), lowercase conversion ($lowercase$()), spelling check ($SpellChecker$()), POS tagging, and lemmatisation ($lemmatize$(); Perkins, 2014).

5.3.1 Feature generation

This process generates two sets of features: word-level features and position features. Although the literature indicates that neural networks are less sensitive than conventional supervised learning algorithms to pre-generated features when text mining with standard datasets, Email, by its nature, contains more implicit features that cannot be learned by neural models. In order to overcome this problem, a separate feature generation process is implemented to produce additional features that are required for sentiment sequence encoding.

5.3.1.1 Word-level features

As discussed in the previous section, lemmatisation for word standardisation is performed as a normalisation step for building vocabularies and generating word-level features. Studies indicate that using lexical resources with deep learning algorithms improves the performance of text classification to some extent (Mikolov et al., 2013; Rao et al., 2018; Yang et al., 2016a). In this study, negation and SWN lexicon features are incorporated into document vectorisation. Background knowledge and theoretical support for SWN 3.0 can be found in Section 3.3.1. The calculations of the sentiment values of sentiment phrases were mathematically defined in Equation 4.1, with sample outputs presented in Table 4.1. A widely used bi-gram model is applied to extract sentiment terms and phases from SWN lexicons, and a list of pruned sentiment features based on the input dataset is computed. Sentiment values are further adjusted by a negation handling process using a $NEGATION\_WORD\_LIST$ derived from (Wilson et al., 2005).

5.3.1.2 Position features

The position of a word is a feature that was initially introduced in natural language processing (Collobert et al., 2011) for semantic role labelling. Recent research undertaken by Yang et al. (2016a) introduced position encoding as a feature representation method for deep CNN for relation classification. Studies also emphasise the importance of structural and relational information in word representations for sentiment analysis (Bhatia et al., 2015; Maas et al., 2011; Tang et al., 2015a). For instance, Bhatia et al. (2015) proved the effectiveness and improvement of revised discourse depth weighting with recursive neural networks used for document sentiment classification. It has been well observed that neural network models with added discrete distance features (modelling relational and structural information captured in word vectorisation) perform better in text mining tasks.

As discussed in the previous section, Email data has an issue of lengthiness that can lead to data sparsity problems. Hence, I implemented the min-max normalisation function using the mean value, a function that is frequently utilised as a standard data cleaning procedure in feature extraction processes (Khan et al., 2016, 2017). Through the implementation of the min-max normalisation function, I scaled the initial positions of words in relation to the length of the corresponding sentence. Denote $\mathcal{S} = \{s_1, s_2, \ldots, s_p\}$ as a collection of sentences in each Email message and $ed_i \in \mathcal{ED}$, $\mathcal{ST} = \{st_{k1}, st_{k2}, \ldots, st_{kq}\}$ as a list of tokens in each sentence, where $s_k \in \mathcal{S}$ consists of q words, and $\mathcal{P} = \{p_{k1}, p_{k2}, \ldots, p_{kq}\}$ as a list of corresponding positions, in which p_{kq} represents the position of the q^{th} word in a sentence s_k. The $normalise$ function is mathematically formulated as:

$$normalise(kj) = 1 + (p_{kj} - \frac{L(s_k)}{2}) * \frac{1}{L(s_k) - 1}, \tag{5.1}$$

where p_{kj} represents the initial position of word st_{kj} in sentence s_k for $1 \leq k \leq p \wedge 1 \leq j \leq q$. The $L()$ function computes the length of sentence s_k where word st_{kj} belongs, and returns the length as input to obtain a relative distance of st_{kj}.

Plain Text(PT)-based position A popular linear scaling method (Aqil Burney et al., 2012) is applied to initial positions of words that are represented by the chronological order of words in a sentence (see Figure 5.3). Let the PT-based position of a sentence s_k be a row matrix $\mathcal{PT}$ composed of elements $[pt-pf(1), pt-pf(2), \ldots, pt-pf(kj)]$, in which each element $pt-pf(kj)$ is calculated by a function $pt-pf()$ defined as:

$$\begin{aligned} pt - pf(kj) &= 1 + normalise(kj) * \frac{s_k}{L(ed_i)}, \\ PT &= pt - pf(1) \oplus pt - pf(2) \oplus \cdots \oplus pt - pf(kj), \end{aligned} \tag{5.2}$$

where s_k represents the k^{th} sentence in the input document ed_i, $L(ed_i)$ represents the number of sentences in the document, and $\oplus$ is a concatenation operator. In Equation 5.2, $pt - pf(kj)$ of word st_{kj} in sentence s_k is computed by the scaled fraction of $L(ed_i)$ and the normalised position $normalise(kj)$ of word st_{kj}.

Dependency Graph(DG)-based position Compared to a plain text-based position, a DG-based position is capable of capturing more syntactic information and semantic relations among words and phrases (Bhatia et al., 2015; Yang et al., 2016a). For instance, given a sentence, "I have never had a holiday in Venice.", represented with the dependency graph structure (see Figure 5.3), seven sets of dependency relations are discovered. Among them, relations between terms 'I' and 'had', 'have' and 'had', 'never' and 'had', and 'had' and 'holiday' are parallel, and are of the same level of importance. Hence, equal positions are assigned to each dependent term 'I', 'have', 'never' and 'holiday' in the dependency structure. The basic concept of the DG-based position is derived from previous research conducted by Nakagawa et al. (2010) and Yang et al. (2016a). The tree-based position feature (Yang et al., 2016a) was revised by building a linear weighting function in order to incorporate position encoding.

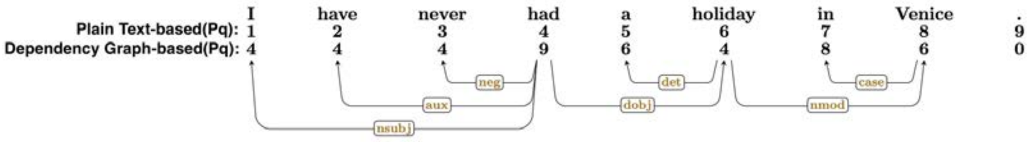

I have never had a holiday in Venice .
Plain Text-based(Pq): 1 2 3 4 5 6 7 8 9
Dependency Graph-based(Pq): 4 4 4 9 6 4 8 6 0
neg
aux
nsubj
det
dobj
case
nmod

FIGURE 5.3: A dependency graph representing a sample sentence with PG-based and DT-based position p_{kj} annotations for $p_{kj} \in \mathcal{P}$.

The initial input for the DG-based position is a row vector of tree-based positions of words in a sentence. Figure 5.3 displays a sample dependency graph representing a sentence annotated with the PT-based and DG-based position p_r. Pre-developed functions, involving $sent\_tokenize()$, $raw\_parse()$ and $convert\_tree()$, in the Stanford NLP toolkit Manning et al., 2014 are implemented to perform sentence tokenisation, tree parsing and dependency graph conversion on Email documents. The details are presented in Pseudocode 4. The same $normalise()$ function formulated in Equation 5.1 is utilised to compute the normalised DG-based position of words. To compute the discourse depth of a sentence in a document, the $weight()$ operator derived from Bhatia et al. (2015) is used, as defined below:

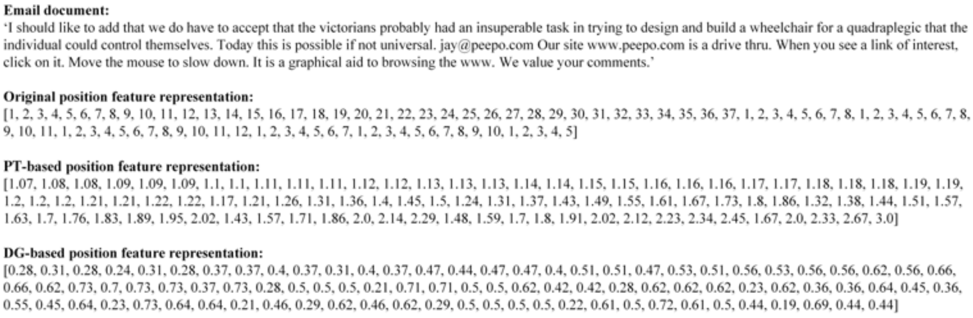
$$weight(k) = \max(0.5, 1 - \frac{s_k}{L(ed_i)})\ (1 \leq k \leq p), \tag{5.3}$$

where $L(ed_i)$ represents the number of sentences in the input document $ed_i \in \mathcal{ED}$, and $s_k \in \mathcal{S}$. In the above equation, the first sentence s_1 returns the highest weight of $1 - \frac{1}{L(ed_i)}$, and when $p > \frac{2}{L(ed_i)}$, the corresponding weight remains to be 0.5 for $s_k \in \mathcal{S} \wedge ed_i \in \mathcal{ED}$.

Let the DG-based position of a sentence s_k be a row matrix $\mathcal{DG}$ composed of elements $[dg - pf(1), dg - pf(2), \ldots, dg - pf(kj)]$, where each element $dg - pf(kj)$ is computed as below:

$$\begin{aligned} &dg - pf(kj) = normalise(kj) * weight(k), \\ &DG = dg - pf(1) \oplus dg - pf(2) \oplus \cdots \oplus dg - pf(kj), \end{aligned} \tag{5.4}$$

where $normalise(kj)$ returns the normalised value of position p_{kj} for $1 \leq k \leq p \wedge 1 \leq j \leq q$, and the DG-based position is calculated by the multiplication of the weighted value of a sentence $weight(k)$ and the normalised position $normalise(kj)$.



Email document:
'I should like to add that we do have to accept that the victorians probably had an insuperable task in trying to design and build a wheelchair for a quadraplegic that the individual could control themselves. Today this is possible if not universal. jay@peepo.com Our site www.peepo.com is a drive thru. When you see a link of interest, click on it. Move the mouse to slow down. It is a graphical aid to browsing the www. We value your comments.'

Original position feature representation:
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 1, 2, 3, 4, 5, 6, 7, 8, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 1, 2, 3, 4, 5, 6, 7, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 1, 2, 3, 4, 5]

PT-based position feature representation:
[1.07, 1.08, 1.08, 1.09, 1.09, 1.09, 1.1, 1.1, 1.11, 1.11, 1.11, 1.12, 1.12, 1.13, 1.13, 1.13, 1.14, 1.14, 1.15, 1.15, 1.16, 1.16, 1.16, 1.17, 1.17, 1.18, 1.18, 1.18, 1.19, 1.19, 1.2, 1.2, 1.2, 1.21, 1.21, 1.22, 1.22, 1.17, 1.21, 1.26, 1.31, 1.36, 1.4, 1.45, 1.5, 1.24, 1.31, 1.37, 1.43, 1.49, 1.55, 1.61, 1.67, 1.73, 1.8, 1.86, 1.32, 1.38, 1.44, 1.51, 1.57, 1.63, 1.7, 1.76, 1.83, 1.89, 1.95, 2.02, 1.43, 1.57, 1.71, 1.86, 2.0, 2.14, 2.29, 1.48, 1.59, 1.7, 1.8, 1.91, 2.02, 2.12, 2.23, 2.34, 2.45, 1.67, 2.0, 2.33, 2.67, 3.0]

DG-based position feature representation:
[0.28, 0.31, 0.28, 0.24, 0.31, 0.28, 0.37, 0.37, 0.4, 0.37, 0.31, 0.4, 0.37, 0.47, 0.44, 0.47, 0.47, 0.4, 0.51, 0.51, 0.47, 0.53, 0.51, 0.56, 0.53, 0.56, 0.56, 0.62, 0.56, 0.66, 0.66, 0.62, 0.73, 0.7, 0.73, 0.73, 0.37, 0.73, 0.28, 0.5, 0.5, 0.5, 0.21, 0.71, 0.71, 0.5, 0.5, 0.62, 0.42, 0.42, 0.28, 0.62, 0.62, 0.62, 0.23, 0.62, 0.36, 0.36, 0.64, 0.45, 0.36, 0.55, 0.45, 0.64, 0.23, 0.73, 0.64, 0.64, 0.21, 0.46, 0.29, 0.62, 0.46, 0.62, 0.29, 0.5, 0.5, 0.5, 0.5, 0.22, 0.61, 0.5, 0.72, 0.61, 0.5, 0.44, 0.19, 0.69, 0.44, 0.44]

FIGURE 5.4: A sample Email document represented as position features.

Figure 5.4 illustrates the differences between PT-based position representation, the DG-based position representation and the original position features using an example Email document.

5.3.2 Document vectorisation

As neural networks take vectorised text documents as inputs, converting texts from the document space into the vector space using document vectorisation methods is implemented as the third phase of the proposed approach. Techniques for transforming documents into numerical vectors are widely studied, as the process of document vectorisation has a significant influence on the performance of classification algorithms. In this study, the document vectorisation phase is divided into two parts: word embedding for word-level syntactic and semantic information (in order to model local variations), and sentiment sequence encoding for sentence-level relational and structural information (in order to model global variations).

5.3.2.1 Word embedding

Compared to the one-hot encoding technique (Harris and Harris, 2015), which transforms text documents into high-dimensional word representations that are computationally expensive, word embedding techniques efficiently convert documents into distributed feature representations with a fixed length of continuous vectors, which greatly assists in classification performance (Collobert et al., 2011; Yang et al., 2016a).

The advent of various word embedding learning techniques developed for semantic parsing helps neural network models to capture precise contextual similarities when mapping terms to vectors for natural language processing. Studies demonstrate that pre-trained word embedding models, such as GloVec or Google Word2Vec, show solid performance that outperforms randomly post-trained models (Rao et al., 2018) in various sentiment analysis tasks (Majumder et al., 2017; Tang et al., 2015a,b). In this research, as trial experiments indicated that there were no major performance gaps among the different word embedding models, an extended version of GloVec (Pennington et al., 2014) that was trained on a corpus composed of a vocabulary of 400,000 words with a dimensionality of 100 was utilised for its moderate space and time complexity.

5.3.2.2 Sentiment sequence encoding

In this study, a sentiment sequence encoding technique was developed using the LSTM model to encode aggregated DG-based position features and sentiment lexical features on the basis of the tree-based position-encoding technique proposed by Yang et al. (2016a). The encoding process is further divided into 1) sentiment sequence feature aggregation, which aims to map sentiment features with positions, and 2) LSTM encoding, which adds an LSTM layer to the concatenated sentiment sequence features to build a sequence embedding layer in the proposed neural sentiment classification model.

Sentiment sequence feature aggregation. To concatenate sentiment features with positions, an aggregation method is exploited using a matrix product function. Denote $\mathcal{SWN} = \{swn_{k1}, swn_{k2}, \ldots, swn_{kq}\}$ to be a set of phrases with a sentiment score of the words assigned to each sentence of $s_k \in \mathcal{S}$. If a word $st_{kj} \in \mathcal{ST}$ exists in the SWN lexicon, a corresponding sentiment value $swn_{kj} \in \mathcal{SWN}$ is calculated by the weighted average synset score of all synset terms belonging to the word. swn_{kj} will be set to zero otherwise.

Let θ be a matrix composed of the refined sentiment values swn_{kj} concatenated with the normalised positions $normalise(kj)$, and δ be a matrix composed of the scaled positions $pt - sf(kj)$ and $weight(k)$, in which $weight(k)$ is extended to a column matrix with a row number of j for each sentence s_k in the matrix. The aggregated PT- and DG-based sentiment sequence matrix $pt - ssf$ and $dg - ssf$ for a sentence s_k is constructed by the $matrix\ product$ of transposed θ and δ, and by the $matrix\ product$ of transposed θ and the weighted value of a sentence $weight(k)$, respectively. The mathematical formulas are:

$$\begin{aligned} &pt - ssf = \theta^T \cdot \delta, \\ &pt - ssf_{kj} = \sum_{ab} \theta_{kj_a} \delta_{kj_b},\ a, b \in (1, 2), \end{aligned} \tag{5.5}$$

where δ represents a matrix of two columns $\in \mathbb{R}^{2 \times kj}$ composed of transposed vector PT^T and vector $weight(k)$. The $\cdot$ operator is a matrix product that returns a column vector consisting of the sum of the multiplication of θ_{kj_a} and δ_{kj_b}, for $a, b \in (1, 2)$.

$$\begin{aligned} &dg - ssf = (\theta^T \cdot weight(k))^T, \\ &dg - ssf_{kj} = \sum_{a} \theta_{kj_i} weight(k),\ a \in (1, 2), \end{aligned} \tag{5.6}$$

where θ represents a matrix of two rows concatenated by a vector $normalise(kj)$ and a vector swn_{kj}, and θ^T represents the transpose of the concatenated matrix. The $\cdot$ operator is a matrix product that returns a column vector composed of the sum of the multiplication of θ_{kt_i} and $weight(k)$, where θ_{kj_a} is composed of two values $normalise(kj)$ and swn_{kj} for $1 \leq k \leq p \ \wedge \ 1 \leq j \leq q$. A Pseudocode 4 that exposits the process of building a $dg - ssf$ matrix is presented below.

Pseudocode 4 DG-Sentiment Sequence Feature Aggregation

```
Input: An Email document $ed_i$ consists of a collection of sentences $\mathcal{S}$;
Output: A row matrix $SSF$ of sentiment sequence features for a sentence $s_k$ in the Email document $ed_i$;
for each Email document $ed_i \in \mathcal{ED}$ do
    Tokenise $ed_i$ into a collect of sentences $\mathcal{S}$ using $sent\_tokenize()$ function;
    for each sentence $s_k \in \mathcal{S}$ do
        Compute the length of each sentence $L(s_k)$;
        Parse each sentence $s_k$ using $raw\_parse()$ function into dependency trees;
        Convert dependency trees into tuples using $convert\_tree()$ function;
        for each token $st_{kj} \in \mathcal{ST}$ do
            Let $\Phi$ and $swn$ be two row matrices of size $q$;
            Compute $normalise(kj)$ for initial position $p_{kj}$;
            Set $\Phi \leftarrow normalise(kj)$;
            Extract $swn_{kj}$ from $SWN$;
            if $st_{kj-1} \in NEGATION\_WORD\_LIST$ then
                for $1 \leq k \leq p \ \wedge \ 1 \leq j \leq q$ do
                    if $p_{kj}$ is equal to $p_{kj-1}$ then
                        Set $swn_{kj} \leftarrow swn_{kj} * (-1)$;
                    end if
                end for
            end if
            Set $swn \leftarrow swn_{kj}$;
        end for
        Concatenate $\Phi$ and $swn$ into $[\Phi, swn]$;
        Compute $[\Phi, swn]^T * weight(k)$;
        for $1 \leq k \leq p \ \wedge \ 1 \leq j \leq q$ do
            Let $sum_{kj} = 0$;
            Set $sum_{kj} \leftarrow sum_{kj} + [normalise(kj) \times weight(k) + swn_{kj} \times weight(k)]$;
            Set $dg - ssf_{kj} \leftarrow sum_{kj}$;
        end for
        Set $dg - ddf \leftarrow dg - ssf_{kj}$;
    end for
end for
```

LSTM encoding. To generate sentiment sequence encoding and concatenate the encoded features into the proposed neural network model, an LSTM layer is applied to the above sentiment sequence matrix. Compared with conventional RNNs, LSTM implements a variant architecture that is specifically designed to capture long-term dependencies in sequence structured data. To minimise the exploding gradient problem, LSTM introduces an individual memory cell that stores the hidden state of the previous memory state and three sigmoid gates to control the gradient flow (Hochreiter and Schmidhuber, 1997). By computing element-wise multiplication with hyperbolic tangent activation function between cells, only necessary state information is retained and updated so that the gradient value is kept within a certain range that will neither vanish nor explode (Hochreiter and Schmidhuber, 1997). A standard LSTM layer, consisting of input gates, forget gates and output gates, is adopted for encoding the sentiment sequence with timesteps and hidden states from sentiment sequence features. In detail, at each time step t, the forget gate computes the output of the current time step based on the previous hidden state and the input of the current time step. Element-wise multiplication is then applied to the previous cell state and the forget gate with a sigmoid activation function to update the current cell state. The final output is a hidden state that is updated by multiplication of the output cell and the previous cell state with a hyperbolic tangent function activated.

The mathematical formula for the working mechanism of each LSTM network unit at each time step t is described as follows:

$$\begin{aligned}
&\begin{pmatrix} i_t \\ f_t \\ o_t \end{pmatrix} = \sigma \begin{pmatrix} W_i \cdot [h_{t-1}, s_t] + b_i \\ W_f \cdot [h_{t-1}, s_t] + b_f \\ W_o \cdot [h_{t-1}, s_t] + b_o \end{pmatrix} \\
&\bar{c}_t = tanh(W_o \cdot [h_{t-1}, s_t] + b_o) \\
&c_t = f_t \odot c_{t-1} + i_t \odot \bar{c}_t \\
&h_t = o_t \odot tanh(c_t),
\end{aligned} \tag{5.7}$$

where s_t represents the input of an LSTM unit at time step t; and $i_t, f_t, o_t, \bar{c}_t, c_t, h_t$ are the input gate, forget gate, output gate, temporal cell state, current cell state and output state of the LSTM unit, respectively. Additionally, W_i, W_f, W_o represent weight vectors added to the input, forget and output gate, and b_i, b_f and b_o represent bias vectors added to each layer. Each layer is further updated using the sigmoid σ activation function. The output of the LSTM unit h_t is generated by looping through a memory cell state c_t that is computed by an element-wise multiplication $\odot$ function on the forget gate f_t and the previous cell state c_{t-1}, the input gate i_t and the temporal cell state $\bar{c}_t$, in which the temporal cell state $\bar{c}_t$ is calculated by the hyperbolic function $tanh$ activated output gate o_t.

To build inputs for an LSTM layer, sentences and documents are first padded with zeros to ensure a fixed length of timesteps. Denote the sentiment sequence features representing document ed_i as a matrix $\mathbb{E} \in \mathbb{R}^{h \times v \times s}$, where h represents the number of hidden units, and v and s refer to the maximum numbers of words in a sentence and sentences in a document based on the input dataset. With the return sequence parameter of the LSTM layer set to true, the output results in a sentiment sequence encoding SSE are denoted as the matrix $\mathbb{E} \in \mathbb{R}^{v \times s}$.

5.3.3 Neural sentiment classification

For the neural sentiment classification phase, I revised a convolutional neural architecture based on one of the classic variant CNN models developed by Kim (2014) with parameter tuning regulated as per Zhang and Wallace (2015). To be more specific, the proposed model consists of five main steps as illustrated in Figure 5.5.

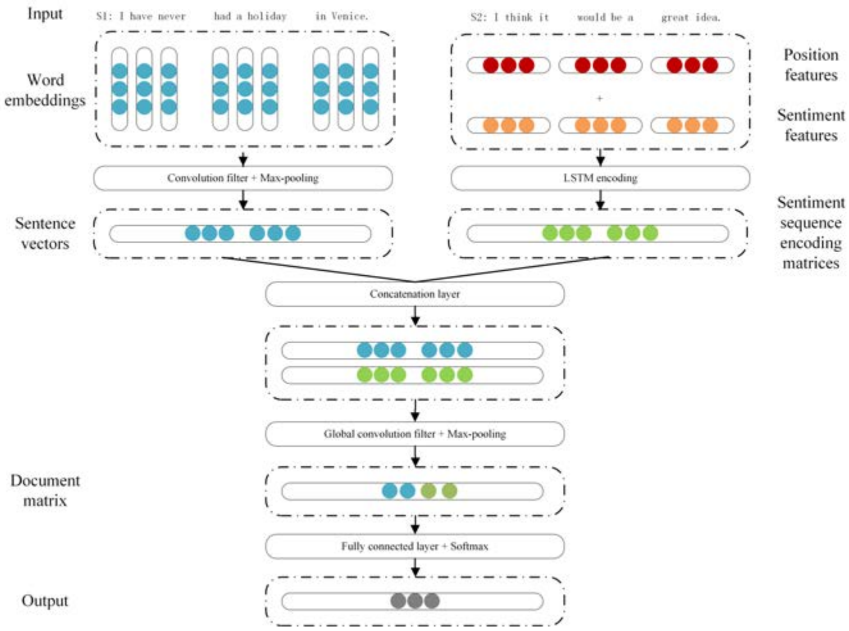


FIGURE 5.5: Overall structure of the proposed sentiment sequence encoding incorporated CNN model. The neural model is presented with two sample input sentences, where the word embedding layer is the representation of the first input sentence, and the sentiment sequence feature layer, concatenated by position and sentiment features, is the representation of the second input sentence. For the sentence vectors and document matrix, neurons are presented with a concatenation of two sentences.

First of all, the input is built on n documents. Each document is represented by a vectorised word embedding matrix denoted by $ed_i \in \mathbb{R}^{d \times v \times s}$, in which d refers to the dimension of word embeddings, v refers to the sentence vocabulary size, and s refers to the maximum number of sentences in the corpus. In this study, d is set to a fixed-length of 100 dimensions for the pre-trained word embedding model, and v and s are set to the maximum numbers of words in a sentence and sentences in a document, respectively, based on the input dataset. For documents with a sentence vocabulary v and number of sentences s less than the maximum, the sentences and documents are padded with dummy values to obtain fixed-length inputs.

The second step is to aggregate documents represented by word embeddings into sentence vectors. A convolutional filter $W_v \in \mathbb{R}^{v \times n \times d}$ is applied to each sentence matrix $s_p \in \mathbb{R}^{v \times d}$, where n represents the value in the range of the filtering window size parameter. To normalise the convolutional filter output, a bias $b_v \in \mathbb{R}^v$ and a Rectified Linear Unit(ReLU) non-linearity activation function are added to sentence vector s_k. Then, a max-pooling function is implemented to reduce the dimensionality of matrix s_k to $\mathbb{R}^{v \times 1}$.

In the third step, sentiment sequence encoding matrices are used as inputs and are concatenated with sentence matrices. As discussed in Section 5.3.2, the output of the LSTM encoded sentiment sequences for each document is presented as a matrix $\mathbb{E} \in \mathbb{R}^{v \times s}$. For the forth step, an aggregated sentence matrix for a document $\mathbb{S} \in \mathbb{R}^{v \times s}$ and a sentiment sequence encoding matrix $\mathbb{E}$ are concatenated with document vectors ed_n^{se} using the element-wise maximum of all sentences in a document. This results in a document matrix denoted by $ed_n^{se} \in \mathbb{R}^{v \times 2}$ that is calculated by:

$$
\begin{aligned}
ed_n^s &:= \max_{\{1 \leq i \leq s\}} \mathbb{S}_{in}, \\
ed_n^e &:= \max_{\{1 \leq i \leq s\}} \mathbb{E}_{in}, \\
ed_n^{se} &= ed_n^s \oplus ed_n^e, n \in \mathbb{R}^{v \times 2},
\end{aligned}
\tag{5.8}
$$

where := is an element-wise maximum operator that returns the maximum value of matrix $\mathbb{S}_{in}$ and matrix $\mathbb{E}_{in}$, and assigns it to the revised document matrix $ed^s e_n$.

Finally, a fully connected softmax layer is implemented with a global convolutional filter applied to the current document matrix $ed_n^{se}\mathbb{R}^{s\times 2}$ and a softmax function returns the class probability of an input document. The global convolutional filter consists of a combination of a weight $W_s \in \mathbb{R}^{s\times n\times 2}$ and a bias $b_s \in \mathbb{R}^2$, and a weight $W_e \in \mathbb{R}^{s\times n\times 2}$ and a bias $b_e \in \mathbb{R}^2$. The process is formulated as:

$$ed_{in}^{se} = f(W_s \cdot s + b_s + W_e \cdot e + b_e), i \in (1, s) \wedge n \in (1, 2), \tag{5.9}$$

where $f(i)$ represents a $ReLU(i)$ activation function that normalises the input into a positive value by returning $max(0, i)$.

5.4 Empirical experiments

In this section, I discuss the datasets and experimental settings used for performance evaluation. I obtained empirical results through experiments that compared the proposed model against other widely-used sentiment classification algorithms. I also report on the findings of further experiments investigating the effects of various feature representation techniques on different neural network models.

5.4.1 Datasets

For the empirical experiments, I used the three benchmark datasets and their corresponding augmented datasets described in Section 3.2 and 3.3. The class distribution of each dataset can be found in Table 3.1 and 3.4.

5.4.2 Comparative methods

To justify the effectiveness of the proposed method, I performed a set of comparative evaluations with recent Email sentiment classification methods (Chhaya et al., 2018; Liu and Lee, 2018) as well as baseline methods and state-of-the-art approaches to sentiment classification. Review on literature indicated that the study on Email sentiment analysis was limited, especially in terms of the quantitative evaluations. Hence, representative algorithms from different categories were selected for generating benchmarking results for Email sentiment classification. The comparative methods can be divided into three major categories—unsupervised learning, supervised learning, and neural network algorithms—and are described next.

Unsupervised learning techniques. Two clustering-based approaches and one lexicon-based baseline method comprise the unsupervised learning category.

- **Baseline**: A purely lexical approach based on features using an SWN lexicon and BoWs. The final sentiment polarity is determined by the accumulated value of features, of which above zero is classified as positive, below zero as negative and equal to zero as neutral.
- ***k*-Means** (Liu and Lee, 2018): An unsupervised clustering approach with a revision of the model evaluation to extend it to sentiment classification based on the study of Liu and Lee (2018). To reduce the computational complexity, three initial centroids representing the positive, negative and neutral class respectively were chosen from the dataset and the number of clusters k was set to be three.
- **Senti**$\mathcal{TRACLUS}$ (Liu and Lee, 2018): A sequence-based approach developed by Liu and Lee (2018) that modifies the original TRACLUS algorithm proposed for spatiotemporal datasets. The main feature of this method is that it transforms documents into trajectories that incorporate their sentiment sequences for better sentiment classification performance. The polarity of each Email is determined by the cluster group it is assigned to. A cluster group with a final sentiment value above zero is labelled as positive, below zero as negative and equal to zero as neutral.

Supervised learning techniques. Four state-of-the-art supervised learning techniques—NB, Radial Basis Function Neural(RBFN), RF and SVM—are included in the supervised learning category. These four algorithms are chosen as the representatives of the probabilistic, neural-based, ensemble learning and discriminative classifiers respectively. Decision of the options is made based on their popularity in the community of sentiment analysis and previous applications to Email mining (Bogawar and Bhoyar, 2012; Chhaya et al., 2018; Liu and Lee, 2015).

- **NB** (Lewis, 1998): A probabilistic classifier introduced for comparison with SVM and trained using the SWN lexicon and BoWs as features.
- **RBFN** (Scholkopf et al., 1997): A neural network model introduced as a replacement for MLP due to its high efficiency and trained using the SWN lexicon and BoWs as features. It is a standard three-layer neural network model with a non-linear RBF hidden layer with Gaussian radial basis weighting and Euclidean distance concatenation.
- **RF** (Breiman, 2001): A tree-based classifier with randomly distributed vectors and a voting mechanism. This model is also trained using the same SWN lexicon and BoWs as features.
- **SVM** (Chang and Lin, 2011): A linear classification model derived from LibSVM (Chang and Lin, 2011). As SVM yielded the best results of the four supervised algorithms used in the experiments, I implemented four further variants of the feature sets used for training the SVM algorithm in order to test the effect of position embeddings. 1) an SVM$_{swn+bow}$ model trained with a combination of an SWN lexicon and BoWs; 2) an SVM$_{n-gram}$ model trained with the n-gram language model; 3) an SVM$_{aggwe}$ model trained with aggregated word embeddings by averaging the word embeddings of a certain term in a document; and 4) an SVM$_{aggwe+ssf}$ model trained with aggregated word embeddings and sentiment sequence features, where each term is represented by a *dot product* of its word embeddings and DG-based positions as described in Section 5.3.2.

Neural network models. Three standard neural network models, MLP, LSTM and CNN, are included for comparison. As literature indicated that no existing deep neural network model has ever been utilised for Email sentiment analysis, these three models are chosen as the representative models considering their wide applications in recent studies on sentiment analysis.

- **MLP** (Gardner and Dorling, 1998): A simple three-layer multilayer perceptron neural network with nonlinear activation function was adopted and trained using the pre-trained GloVec word embeddings with a dimension of 100.
- **LSTM** (Hochreiter and Schmidhuber, 1997): A special variant of RNNs that handles long-term dependencies using three gate layers and tangent activation for memory cells. The model was also trained using the pre-trained GloVec word embeddings with a dimension of 100.
- **CNN** (Kim, 2014): A classic variant of deep neural networks that captures features using convolutional filters with weights and bias. This was a pure convolutional-based model trained using the same pre-trained GloVec word embeddings with a dimension of 100.

5.4.3 Experimental settings

In this section, a detailed discussion on the experimental settings for algorithms in each category is presented.

- **Unsupervised learning techniques.** For clustering-based methods, including k-Means and Senti$\mathcal{TRACLUS}$, Within Cluster Sum of Squared(WCSS) error evaluation was utilised as a model stopping criterion and for label assignment.
- **Supervised learning techniques.** All algorithms are implemented using packages developed by Hall et al. (2009) in Python programming language. The popular 10-fold cross-validation technique was used in all experiments.
- **Neural network models.** As discussed in the previous section, parameter tuning based on the rules defined in Zhang and Wallace (2015) and Stochastic Gradient Descent(SGD) optimisation (Bottou, 2010) are implemented for MLP and CNN-based models. As for LSTM, semi-supervised parameter adjustments (Hochreiter and Schmidhuber, 1997) are adopted. The hyper-parameter settings for the proposed neural network model are listed in

TABLE 5.1: Hyperparameter settings for using the proposed neural models with three datasets.

Parameter	BC3	EnronFFP	PA
maxSL	165	111	99
maxSD	46	34	20
filter window	[2,3,4]	[3,4,5]	[2,3,4]
filter size	32	64	32
hidden state	32	64	32
batch size	16	32	20
epoch number	20	50	30

Table 5.1. To clarify, $maxSL$ refers to the maximum length of a sentence, $maxSD$ refers to the maximum number of sentences in a document for the input dataset, $batch\ size$ and $epoch\ number$ are shared among all neural-based models, $hidden\ state$ is shared between LSTM cell in the proposed model and the plain LSTM model, and $filter\ window$ and $filter\ size$ are shared among all CNN-based models. The same 100-dimension pre-trained GloVec is used as a word embedding matrix for the three compared models. Experiments with the neural network models were conducted using the popular 10-fold cross-validation technique for consistency with the supervised learning methods used. Note that the parameter values used for the neural network models in this study were set as recommended by the original approaches (Bottou, 2010; Gardner and Dorling, 1998; Hochreiter and Schmidhuber, 1997; Kim, 2014; Zhang and Wallace, 2015).

Experiments using predictive models, including all supervised learning techniques and neural network models, were evaluated by 10-fold cross-validation considering the size of the datasets. In terms of the evaluation matrix, I used accuracy and RMSE as quantitative measurements, which are described in detail in Section 4.5.4.1 and mathematically in Equation 4.6. Compared to other measurements, accuracy and RMSE are more widely adopted for multi-class classification tasks and relevant to this study as the accuracy and error rate of an algorithm serves as an explicit and fundamental reference on its classification performance.

5.4.4 Classification results

In this section, I describe and discuss the classification performance of the proposed model with three base variations: $\mathcal{PF}-\mathcal{CNN}$ for position feature-incorporated CNN models, $\mathcal{SSF}-\mathcal{CNN}$ for sentiment sequence feature-incorporated CNN models, and $\mathcal{SSE}-\mathcal{CNN}$ for sentiment sequence encoding-incorporated CNN models. Their performance is compared with that of the other machine learning algorithms described in Section 5.4.3. Specifically, I utilise an embedding layer with a random uniform distribution within a range $[-0.25, 0.25]$ and a dimension of 50 for the position and sentiment sequence features for the $\mathcal{PF}-\mathcal{CNN}$- and $\mathcal{SSF}-\mathcal{CNN}$-based models. Table 5.2 summarises the overall performance of the different machine learning and deep learning algorithms, in which the results of the proposed neural network model and its variations were selected as the best of two position features.

TABLE 5.2: Overall performance comparison of the various methods under study. The symbol $*$ indicates the best result of two types of position features. Bold text highlights important results.

Dataset Classifier	**BC3**		**EnronFFP**		**PA**	
	Accuracy	RMSE	Accuracy	RMSE	Accuracy	RMSE
Baseline	0.373	1.262	0.205	1.216	0.308	1.016
k-Means	0.314	0.828	0.229	1.138	0.588	0.642
SentiTRACLUS(Liu and Lee, 2018)	0.592	0.876	0.579	0.714	**0.793**	**0.397**
NB(Lewis, 1998)	0.533	0.925	0.550	0.842	0.581	0.810
RBFN(Scholkopf et al., 1997)	0.596	0.892	0.528	0.779	0.607	0.786
RF(Breiman, 2001)	0.557	0.831	0.578	0.72	0.600	0.785
SVM $_{swn+bow}$	0.580	0.727	0.576	0.792	0.587	0.724
SVM $_{n-gram}$	0.592	0.721	0.594	0.721	0.615	0.737
SVM $_{aggwe}$	0.584	0.882	0.595	0.675	0.623	0.706
SVM $_{aggwe+ssf}$	**0.612**	**0.818**	**0.603**	**0.657**	0.637	0.656
MLP(Gardner and Dorling, 1998)	0.789	0.506	0.582	0.651	0.649	0.607
LSTM(Hochreiter and Schmidhuber, 1997)	0.852	0.461	0.586	0.652	0.588	0.642
CNN(Kim, 2014)	0.852	0.461	0.598	0.634	0.653	0.606
PF-CNN $_{*}$	0.856	0.418	0.697	0.636	0.669	0.591
SSF-CNN $_{*}$	0.872	0.413	0.704	0.586	0.738	0.512
SSE-CNN $_{*}$	**0.886**	**0.323**	**0.743**	**0.522**	**0.821**	**0.422**

5.4.4.1 Overall performance

The main findings of Table 5.2 can be summarised in the following four points. Also, noted that a one-tail paired t-test was used as a test of significance, since it is commonly used in data mining and provides sufficient power to detect an effect (McCarroll, 2016).

1. First, the experimental results prove that capturing the sequence and relational information of words and phrases in a document produces better sentiment classification performance. Note that SVM$_{aggwe+ssf}$ performed the best of all machine learning algorithms on the BC3 and EnronFFP datasets, while Senti$\mathcal{TRACLUS}$, a sequence-based approach, performed the best on the PA dataset, with an RMSE of 0.397.

2. Second, the comparison of SVM with other feature representation methods indicates that word embeddings generally provide better performance than lexical SWN features with accuracies of 58.4%, 59.5% and 62.3% obtained for the BC3, Enron FFP and PA datasets, respectively. With deeper analysis, it was found that features represented by a combination of word embeddings and sentiment sequence features further improved the performance of the classifier, with SVM$_{aggwe+ssf}$ achieving the highest accuracies on all three datasets (61.2%, 60.3% and 63.7%, respectively) compared with the other machine learning algorithms. A significance test of accuracy showed that SVM$_{aggwe}$ with aggregated word embeddings performed better than the basic lexicon-based SVM$_{swn+bow}$ (p = 0.084; 90% confidence). Even SVM$_{aggwe+ssf}$ with combined word embeddings and sequence features was significantly better than SVM$_{aggwe}$ (p = 0.053; 90% confidence). A significance test of RMSE showed that SVM$_{aggwe+ssf}$ was significantly better than SVM$_{aggwe}$ (p = 0.042; 90% confidence).

3. Third, the proposed model, $\mathcal{SSE}-\mathcal{CNN}$, yielded the best classification results with the BC3 and EnronFFP datasets, obtaining the highest accuracy rates of 88.6% and 74.3% and the lowest RMSEs of 0.323 and 0.522, respectively. For the PA dataset, the model was most accurate (82.1%) but had

a slightly worse RMSE (0.422) than Senti$\mathcal{TRACLUS}$ (0.397). Therefore, these empirical results demonstrate the superior effectiveness of the proposed model on Email document sentiment classification.

4. Last, it was observed that Senti$\mathcal{TRACLUS}$ performed the best among the unsupervised machine learning approaches, SVM performed the best among the supervised approaches and CNN performed the best among the deep learning methods. Obviously, the proposed method $\mathcal{SSE}-\mathcal{CNN}$ outperformed these three approaches in terms of accuracy and RMSE in general. A significance test of accuracy showed that $\mathcal{SSE}-\mathcal{CNN}$ was significantly better than CNN (p = 0.054) and Senti$\mathcal{TRACLUS}$ (p = 0.085), and SVM$_{aggwe+ssf}$ (p = 0.018). A significance test of RMSE showed that $\mathcal{SSE}-\mathcal{CNN}$ was significantly better than CNN (p = 0.01), and SVM$_{aggwe+ssf}$ (p = 0.058).

I conducted further evaluations based on the experiments with neural network models to explore the influences of document cleaning, position features and sentiment sequence encoding. The details are presented in the following sections.

5.4.4.2 Effect of Email document cleaning and data augmentation

A discussion of the effects of cleaning and data augmentation is made in this section. As stated in previous sections, Email data, especially that extracted from real-life situations, contains unstructured contents, mark-ups and other information that is unnecessary for sentiment classification. Accordingly, it is assumed that conducting Email-specific cleaning will enhance classification performance. To better understand the effects of cleaning, experiments were undertaken with three raw and cleaned datasets and different adaptations of neural network models.

Table 5.3 compares the classification results of the neural network models using raw and cleaned data from the three datasets as input. As shown in the table, the overall classification performance is better with cleaned data, regardless of which classifier is used. All cleaned results are better than raw ones, except for the LSTM model with the PA dataset. This issue may be due to the LSTM model's mechanism

TABLE 5.3: Performance comparison of neural network models with raw and cleaned datasets.

Model \ Dataset	BC3				Enron FFP				PA			
	Raw		**Cleaned**		**Raw**		**Cleaned**		**Raw**		**Cleaned**	
	Accuracy	RMSE	Accuracy	RMSE	Accuracy	RMSE	Accuracy	RMSE	Accuracy	RMSE	Accuracy	RMSE
MLP	0.789	0.528	0.789	0.506	0.564	0.682	0.582	0.651	0.641	0.637	0.649	0.607
LSTM	0.792	0.574	0.852	0.461	0.589	0.641	0.594	0.637	0.594	0.637	0.588	0.642
CNN	0.826	0.491	0.852	0.461	0.586	0.651	0.598	0.634	0.637	0.656	0.653	0.606
PF-CNN	0.836	0.485	0.856	0.418	0.692	0.640	0.697	0.636	0.651	0.626	0.669	0.591
SSF-CNN	0.859	0.472	0.872	0.413	0.697	0.604	0.704	0.586	0.730	0.518	0.738	0.512
SSE-CNN	0.874	0.346	0.886	0.323	0.711	0.530	0.743	0.522	0.798	0.430	0.821	0.422

of capturing long-term dependencies instead of local features, but further analysis is needed to make a solid conclusion. The accuracy statistics show that cleaning significantly improves the accuracy with raw datasets (p = 0.089 for BC3, p = 0.014 for EnronFFP, and p = 0.022 for PA). Similarly, the RMSE statistics show that cleaning improves the accuracy with raw datasets (p = 0.034 for BC3, p = 0.012 for EnronFFP, and p = 0.03 for PA).

Furthermore, as explained in the previous section, the implementation of data augmentation is done to handle cases with insufficient training data and imbalanced class distributions. Therefore, I explored the performance of the algorithms at different levels on augmented datasets that were analysed in two additional sets of experiments. Table 5.4 and Figure 5.6 summarise the results of the two experiments.

TABLE 5.4: Performance comparison of algorithms of different categories on original and augmented datasets. Results for augmented datasets are achieved using a ratio of $100 : 1$ to its original. Bold text highlights important results.

Model \ Dataset	BC3				EnronFFP				PA			
	Original		**Augmented**		**Original**		**Augmented**		**Original**		**Augmented**	
	Accuracy	RMSE	Accuracy	RMSE	Accuracy	RMSE	Accuracy	RMSE	Accuracy	RMSE	Accuracy	RMSE
Baseline	0.373	1.262	0.500	0.914	0.205	**1.216**	0.297	1.406	0.308	1.016	0.459	0.937
SentiTRACLUS	0.592	0.876	0.616	0.771	0.579	**0.714**	0.659	0.758	0.793	**0.397**	0.814	0.434
SVM $_{aggwe+ssf}$	0.612	0.818	0.832	0.450	0.603	0.657	0.742	0.517	0.637	0.656	0.727	0.614
CNN	0.852	0.461	0.897	0.313	0.598	0.634	0.788	0.522	0.653	0.606	0.801	0.461
SSE-CNN	0.886	0.323	0.927	0.271	0.743	0.522	0.824	0.493	0.821	0.422	0.847	0.394

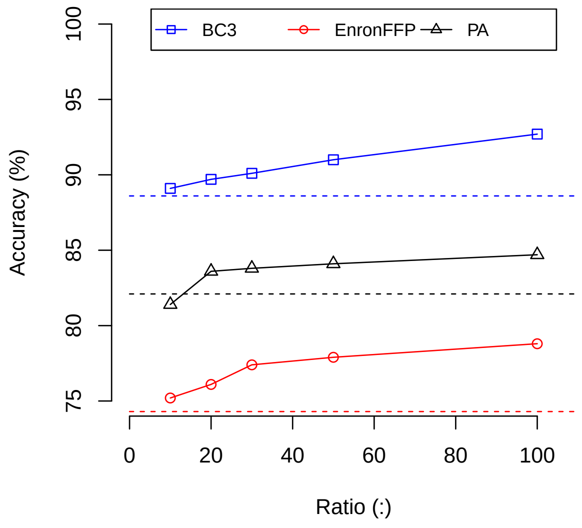
The results in Table 5.4 were obtained by experimenting with five representative algorithms from each category of the comparison methods run on original and augmented datasets with a ratio of 100 to their original. A class distribution of datasets with a ratio of 100 is presented in Table 3.4. In general, the algorithms performed better on the augmented datasets. For statistical evaluation, t-tests were used to determine whether the increases in accuracy with augmented data were significant (p = 0.034 for BC3, p = 0.008 for EnronFFP, and p = 0.018 for PA). Though the differences in RMSE rates between baseline and Senti$\mathcal{TRACLUS}$ algorithms are smaller, t-test results indicate there was a significant reduction in RMSE values with augmented data analysed by supervised learning techniques and neural network models (p = 0.09 for BC3, p = 0.05 for EnronFFP, and p = 0.096 for PA).

Figure 5.6 further illustrates the classification accuracy of the proposed $\mathcal{SSE} - \mathcal{CNN}$ model with augmented data, both balanced and imbalanced, at different ratios. Dotted lines indicate the benchmark values of the original datasets. With the steady growth in accuracy rates with both imbalanced and balanced ratios, augmented data is demonstrated to improve the performance of neural network models, with balanced augmented data outperforming imbalanced data.

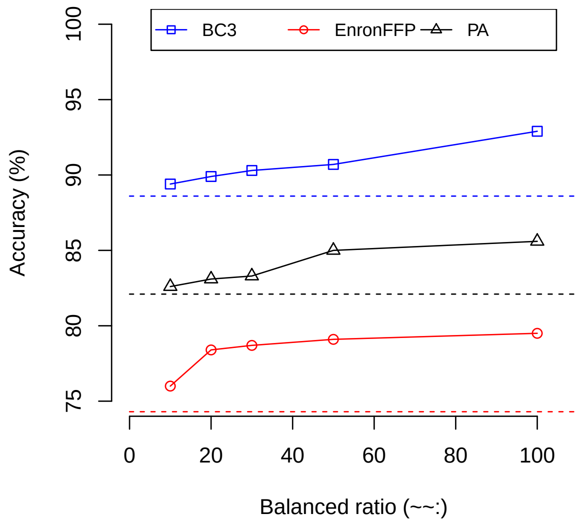
5.4.4.3 Effect of position features and sentiment sequence encoding

As a significant component of the proposed method, the effect of position features and sentiment sequence encoding was analysed by conducting in-depth comparative experiments. Table 5.5 summarises the performance of the proposed model with different position features. In detail, the effects of position features were evaluated with the three types of models (plain CNN-based models $\mathcal{PT} - \mathcal{CNN}$ and $\mathcal{DG} - \mathcal{CNN}$, CNN models with sentiment sequence features $\mathcal{PT} - \mathcal{SSF} - \mathcal{CNN}$ and $\mathcal{DG} - \mathcal{SSF} - \mathcal{CNN}$, and CNN models with a component of sentiment sequence encoding $\mathcal{PT} - \mathcal{SSE} - \mathcal{CNN}$ and $\mathcal{DG} - \mathcal{SSE} - \mathcal{CNN}$).

As shown in Table 5.5 above, though PT-based position features improved the classification performance of the plain CNN model, DG-based position features produced even better results with all three datasets and model settings. Therefore, it is believed that the DG-based approach is the preferred technique for capturing relational and structural information among sentences, and neural network models with DG-based position features assist in better sentence modelling and sentiment classification for Email documents. The tests of significance show that the



(A) Accuracy over ratios.



(B) Accuracy over balanced ratios.

FIGURE 5.6: Classification accuracy with regard to different levels of augmentation, where dot lines indicate the benchmark values of original datasets: a) the number of ratios to be augmented; b) the number of balanced ratios to be augmented.

DG-based approach generally outperforms its corresponding PT-based approach and, in particular, it significantly improves the corresponding PT-based approach with the PA dataset ($p = 0.081$ for accuracy, $p = 0.03$ for RMSE).

For further analysis of the effects of sentiment sequence encoding, experiments

TABLE 5.5: Performance comparison of the proposed model with PT- and DG-based position features.

Dataset / Position Features	**BC3**		**EnronFFP**		**PA**	
	Accuracy	RMSE	Accuracy	RMSE	Accuracy	RMSE
PT-CNN	0.852	0.431	0.696	0.637	0.669	0.591
DG-CNN	**0.856**	**0.418**	**0.697**	**0.636**	**0.726**	**0.524**
PT-SSF-CNN	0.872	0.415	0.702	0.601	0.733	0.547
DG-SSF-CNN	**0.872**	**0.413**	**0.704**	**0.586**	**0.738**	**0.512**
PT-SSE-CNN	0.877	0.38	0.743	0.522	0.784	0.453
DG-SSE-CNN	**0.886**	**0.323**	0.743	0.522	**0.821**	**0.422**

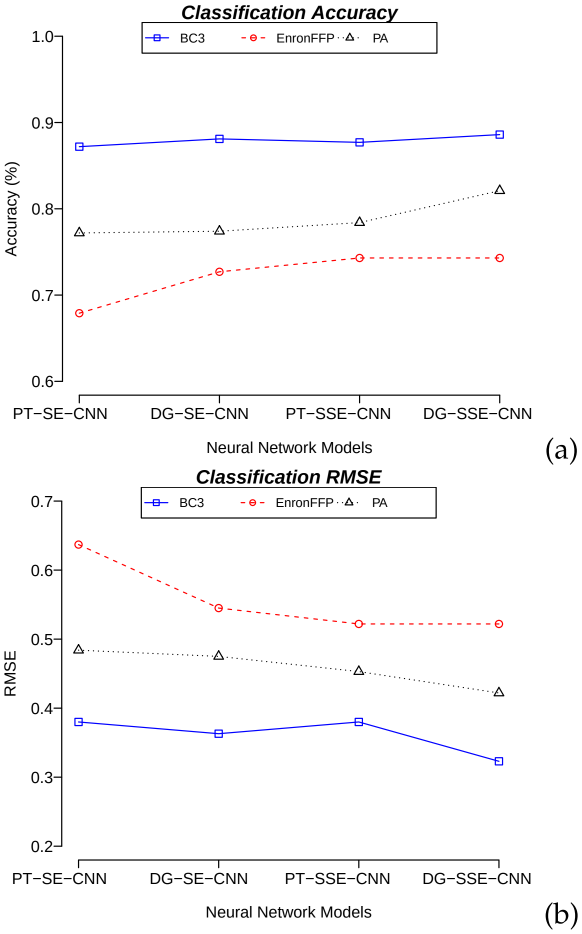


FIGURE 5.7: Performance comparison of sequence encoding with and without sentiment features among three datasets: a) classification accuracy; b) classification RMSE.

with and without sentiment features were undertaken. Figure 5.7 compares the performance of sequence encoding-incorporated CNN models with and without sentiment features. As shown in the figure, sentiment feature-enhanced sequence encoding models performed better than non-sentiment sequence encoding-incorporated neural models. In addition, the results in Table 5.5 and

Figure 5.7 indicate that LSTM-encoded sentiment sequence features performed better than a random uniformed embedding layer.

5.5 Conclusions

This chapter described the study of a sequence-encoded neural classification method for Email document sentiment classification called $\mathcal{SSE}-\mathcal{CNN}$. In the proposed model, sentiment sequence encoding is built on LSTM-encoded sentiment sequence features on the basis of tree-based position features. Discourse sentence weighting and sentiment features are extracted using a sentiment lexicon. Each Email document is represented by a concatenation of a sentence-level dependency-graph matrix and a negation-scaled SWN lexicon feature matrix used as an addition to word embedding for document vectorisation. A deep CNN model is revised accordingly by aggregating word embedding into sentence embedding with sentiment sequence encoding as sentence vectors. The final class assignment is achieved through a global convolutional filter with a softmax function.

The proposed model was quantitatively evaluated against other well-developed algorithms using three real-life Email datasets. The empirical results prove the effectiveness of the proposed neural network model in sentiment classification of Email documents, as well as the positive effects of word positions and relational information on classification performance. Additionally, considering the potential influences of inadequate training data and imbalanced class distributions, it is suggested that data augmentation is a potentially reasonable approach to solving these issues. Performance evaluation and statistical testing demonstrated the significance of using data augmentation with neural network models to improve classification accuracy.

6 Document-level multi-topic sentiment classification with topic-weighted BiLSTM for Email data[15]

In this chapter, I describe a framework for document-level multi-topic sentiment classification for Email data using a topic-weighted BiLSTM model. In Section 6.2, research gaps are identified through a literature review on topic modelling in sentiment analysis. Section 6.3 elaborates on the major phases of the framework, involving document segmentation and multi-topic neural sentiment classification. Section 6.4 summarises the findings of the analysis and evaluation of the proposed approach. Section 6.5 highlights the main contributions of this chapter and draws conclusions. Figure 6.1 illustrates the topics that are covered in this chapter.

6.1 Introduction

Though a significant improvement has been observed in the performance of document sentiment classification in recent years due to the prevalence of neural network models, several challenges still exist due to the complex semantic relations and dependency structures existing among words and sentences. Recent studies are inclined to explore and capture intrinsic sentiment relations and their weighted

[15] This chapter is written based on the following paper 'Liu, S., Lee, K., & Lee, I. Document-level multi-topic sentiment classification of Email data with BiLSTM and data augmentation.' accepted with a minor revision by Knowledge-based Systems.

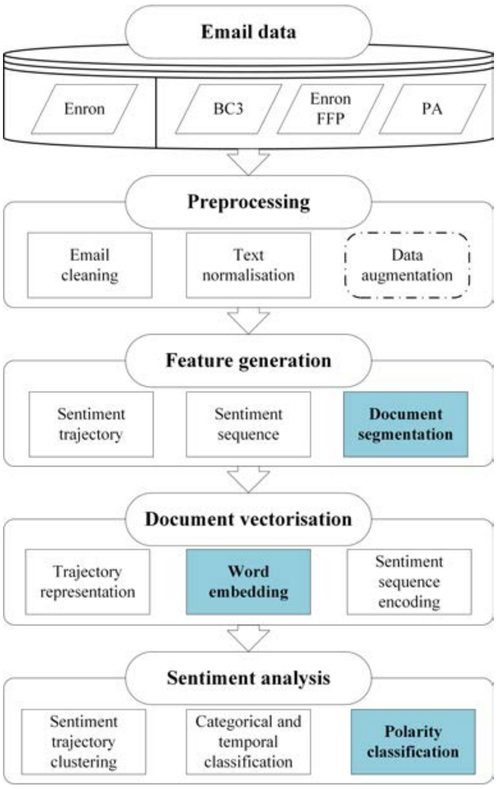


FIGURE 6.1: Overall structure of the Email document sentiment analysis framework, with the specific topics covered in this chapter highlighted in blue and bold.

contributions to the whole document by modelling sentences or aspects within documents to increase the classification accuracy (Bhatia et al., 2015; Chen et al., 2016; Ruder et al., 2016; Yang et al., 2016b; Yin et al., 2017). In particular, methods dependent on a hierarchical structure of documents either consider relevant positions and relational features based purely at the sentence-level (and presume the beginnings and endings sentences have more meaning than the other parts; Bhatia et al., 2015; Chen et al., 2016; Yang et al., 2016b), or include an attention mechanism to model aspects and their corresponding sentiments simultaneously (which necessitates a predefined vocabulary of aspects; Ruder et al., 2016; Yin et al., 2017).

Due to the unique multi-topic feature of Email data (as addressed in Chapter 1), sentiment classification of Emails with multiple topics at the document-level must be considered. I define the problem as being similar to the document-level multi-aspect sentiment classification studied in previous literature (Yin et al., 2017). Nevertheless, as suggested in Figure 1.1, sentiment analysis that involves the concept of aspects (either document-level multi-aspects or aspect-level sentiment classification), takes the following features as input: a list of aspect seed terms, a fixed number of aspect ratings, or aspect labels for sentences (Poria et al., 2016; Ruder et al., 2016; Yin et al., 2017). Note that, a multi-topic Email document is observed with none of the above mentioned features. Namely, these pre-defined features are not available for Email data, instead they need to be populated from the data. Therefore, it is inappropriate to treat multi-topic Email documents as identical as aspects in reviews and other documents. This exploratory approach is more flexible, versatile and suitable than the confirmatory and pre-defined feature based approaches. Figure 1.1 (b) in Chapter 1 shows review and Email examples, which contain a clear set of seed terms for the former (room, value and service) but none for the latter.

In consideration of the unique features of Email documents, including implicit topic-related words and no distinct differences among topics, it is hypothesised that incorporating topic features through unsupervised topic modelling will improve the performance of Email document classification. In this chapter, I discuss the development of a Multi-Topic Bidirectional Long Short-term Memory(MT-BiLSTM) model for document-level sentiment classification of Email data. The main contributions of the study are as follows:

- proposing a framework for document-level multi-topic sentiment classification of Email data;
- improving semantic text segmentation techniques with LDA topic modelling for converting Email into topic segments;
- developing a neural network model for multi-topic sentiment classification using BiLSTM with topic embeddings and topic weighting vectors;
- providing diverse experiments on the performance of the proposed model for comparison with various widely adopted techniques;
- evaluating the classification performance using different parameter settings in the LDA topic model; specifically, the input number of topics and different term weighting methods; and

- examining the effectiveness of the revised data augmentation technique with the proposed model.

6.2 Related work

It becomes ubiquitous to model topics through an unsupervised aspect extraction as the research focus gradually inclined to aspect-level sentiment analysis. Literature advises that topic modelling methods for aspect-level or aspect involved sentiment analysis are mainly categorized into unsupervised learning-based and deep learning-based approaches (Onan et al., 2016; Poria et al., 2016; Ruder et al., 2016; Yin et al., 2017).

As deep learning-based topic modelling approaches require topic labels for training, they are not suitable for the present study on Email sentiment classification study, as pre-labelled training data are unavailable. Thus, unsupervised learning-based approaches are reviewed with in-depth analysis. Among the various existing unsupervised topic models, LDA (Blei et al., 2003) is the most widely-adopted and well-developed model for sentiment analysis tasks. LDA is a generative probabilistic method for modelling collections of discrete data, such as a text corpus (Onan et al., 2016; Poria et al., 2016). For instance, Onan et al. (2016) proposed a weakly-supervised approach that utilizes only minimal prior knowledge—in the form of seed words—to enforce a direct correspondence between topics and aspects. Poria et al. (2016) utilized the concept of semantic similarity to improve the effectiveness of existing LDA models in terms of aspect extraction.

As LDA operates with a full generative model and is capable of handling long documents, it is an ideal candidate for modelling topics in Email documents without a pre-trained corpus or fixed list of topic seeds.

6.3 Proposed framework for document-level multi-topic Email sentiment analysis

In this section, I describe the proposed framework for document-level multi-topic Email sentiment analysis, as presented in Figure 6.2. The general workflow of the framework includes 1) cleaning of Email contents, 2) converting documents into topic segments using LDA topic modelling and semantic text segmentation, and 3) classifying documents into sentiment classes using the MT-BiLSTM neural network model. Note that a data augmentation phase using random word replacement is part of the framework for handling data scarcity and imbalanced class distribution. Further details can be reviewed in Section 3.3.2.

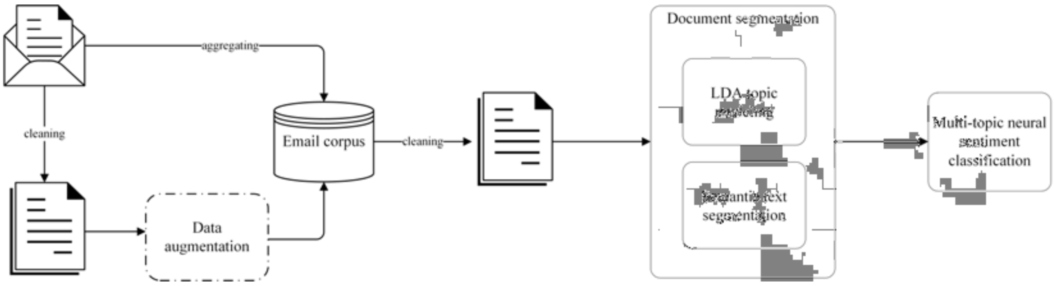


FIGURE 6.2: Overall framework for the proposed document-level multi-topic Email sentiment analysis method.

To acquire high-quality and effective analytical results, data quality should be ensured by implementing data cleaning and normalisation methods. A comprehensive elaboration of the cleaning phase, involving Email cleaning and text normalisation, is provided in Section 3.3.2 and Pseudocode 1. For this study, in particular, different text normalisation tasks are conducted at two sub-steps in the document segmentation phase. First of all, I use the same Python module re (Goyvaerts and Levithan, 2012; as discussed in Section 5.3.1) to filter out duplicated content portions from Emails contents. In terms of input data for LDA topic modelling, to identify more meaningful and reasonable topic distributions, I perform full text normalisation, including tokenisation ($tokenize()$), lowercase conversion ($lowercase()$), spelling check ($SpellChecker()$), short word removal ($len()$), stop word removal ($STOP\_WORD\_LIST$) and lemmatisation ($lemmatize()$; Perkins, 2014). In terms of input data for semantic text segmentation, to maintain the syntactic relations among phases and the semantic integrity of sentences in Emails, a minimal level of text normalisation is utilised at this stage, comprising tokenisation ($tokenize()$), lowercase conversion ($lowercase()$) and lemmatisation ($lemmatize()$; Perkins, 2014) via the Python $nltk$ toolkit.

6.3.1 Document transformation into topic segments

The main component of the proposed framework is document transformation, which aims to model documents based on topic representations and split them into topic segments. In brief, this phase is further divided into an LDA topic modelling process and a semantic text segmentation process. For each Email document, the former step returns a list of topics with sets of keywords as representations, then the number of topics is treated as an input parameter for the latter step of text segmentation to split the document into n segments.

6.3.1.1 LDA topic modelling

As discussed in previous sections, LDA, a generative topic model based on Bayesian probabilistic theory, is a widely adopted technique for modelling text corpora with topic probabilities (Blei et al., 2003). Moreover, previous studies indicated a relatively better performance of the LDA model over other topic modelling methods, such as LSA and Non-negative Matrix Factorization(NMF) (Dredze et al., 2008; Sharaff and Nagwani, 2016) Gensim[16], a well-developed Python library for various statistical modelling (Rehurek and Sojka, 2010), is utilised to implement the various functions involved in the LDA topic modelling process. To generate topic representations for a collection of N documents in a corpus, an $LDAModel$ object is initialised, with documents vectorised using $TFIDFVectorizer$[17] and a value α that specifies the number of topics used as input parameters. Once the LDA model is constructed, the $get\_document\_topics$ function is utilised to return the topic representation with a

[16] https://radimrehurek.com/gensim/

[17] Experiments were conducted with three feature generation methods: n-gram, Word2vec and TF-IDF, among which TF-IDF yielded the best results.

list of topics and their probabilistic distributions for each document. A minimum probability threshold value θ is set on the basis of an adjusted mean calculated as the summation of the mean and skewness of an asymmetric unimodal distribution. The relevant mathematical formulas are presented below:

$$\begin{aligned}\bar{\mathcal{P}} &= \frac{1}{n*m}\sum_{i}^{n}\sum_{j}^{m}\mathcal{P}_{ij}\\ \sigma &= \sqrt{\frac{1}{n*m-1}\sum_{i}^{n}\sum_{j}^{m}(\mathcal{P}_{ij}-\bar{\mathcal{P}})^2}\\ \theta &= \bar{\mathcal{P}}+\sum_{i}^{n}\sum_{j}^{m}\left(\frac{\mathcal{P}_{ij}-\bar{\mathcal{P}}}{\sigma}\right)^3,\end{aligned} \tag{6.1}$$

where $\mathcal{P}_{ij}$ represents the probability of the j^{th} topic that belongs to the i^{th} document for $1 \leq j \leq \alpha$ and $1 \leq i \leq N$.

A revised list of topic representations for each document is generated by removing topics with probabilities less than θ, and storing them with the number of topics assigned to each document for computation in the next step.

6.3.1.2 Semantic text segmentation

In this step, the pre-developed package $TextSegment$ [18] is first utilized to perform text segmentation with the number of topics used as an input parameter. In general, the segmentation process is performed by the $get\_segment\_texts$ function. The basic working mechanism operates on a greedy heuristic algorithm, which chooses the best split point iteratively by computing the weighted distances of words to a segment centroid. The weighted distance of a word is computed by multiplying an entropy with a cosine distance between the average centroids and word embeddings using the pre-trained Glovec model (Pennington et al., 2014); to be consistent with the neural model used hereafter). Equation 6.2 indicates the

[18] https://github.com/ReemHal/Semantic-Text-Segmentation-with-Embeddings

arithmetic calculation for individual entropy ($entropy(i)$) and weighted distance($word\_distance(i)$).

$$\begin{aligned} entropy(i) &= -\frac{f_i}{f} * \log_2 \frac{f_i}{f} \\ word\_distance(i) &= entropy(i) * cos\left(\frac{e_i * entropy(i)}{i+1}, e_i\right), \end{aligned} \tag{6.2}$$

where f_i and f represent the frequency of the i^{th} word and the sum of the frequencies of all words in a document, respectively. e_i represents the word embeddings of the i^{th} word in a document, and the $cos()$ function computes the cosine distance between $\frac{e_i * entropy(i)}{i+1}$ (centroid) and e_i.

Subsequently, the cosine similarity measurement is applied to TF-IDF vectorized words in topic segments and topic representations to assign each topic segment to the corresponding topic in each component. To explain the process in detail, Pseudocode 5, for transferring Email documents into topic segments($EmailTTS$), is presented. Denote $\mathcal{ED}$ as a collection of Email documents composed of messages $\{ed_1, ed_2, \ldots, ed_n\}$, and for each Email document $ed_i \in \mathcal{ED}$, denote $\mathcal{TD}$ as a list of topics $\{td_1, td_2, \ldots, td_m\}$ assigned, and $\mathcal{KW}$ as a list of keywords $\{kw_1, kw_2, \ldots, kw_p\}$ that represents each topic $td_j \in \mathcal{TD}$; $\mathcal{TS}$ as a list of topic segments $\{ts_1, ts_2, \ldots, ts_v\}$ generated, and $\mathcal{TW}$ as a set of token words $\{tw_1, tw_2, \ldots, tw_q\}$ that belongs to each topic segment $ts_j \in \mathcal{TS}$.

6.3.2 Multi-topic neural sentiment classification

The proposed multi-topic neural sentiment classification model is built upon a topical structure with two BiLSTM layers, as introduced by Graves and Schmidhuber (2005). Figure 6.3 illustrates the overall structure of the proposed MT-BiLSTM. The outputs of the first topic-level BiLSTM layer are concatenated with a topic-embedding layer and fed into a document-level BiLSTM that is multiplied by a topic-weighting vector (a weighted representation of topic segments for a given topic).

Pseudocode 5 EmailTTS

```
Input: A set of post-processed Email documents $\mathcal{ED}$;
Output: Each Email document $ed_i \in \mathcal{ED}$ represented by a list of topics $\mathcal{TD}$, in
which a topic is associated with a keyword list $\mathcal{KW}$ and a token list $\mathcal{TW}$;
for each Email document $ed_i \in \mathcal{ED}$ do
    Tokenise $ed_i$ into a collection of words;
    Apply $TFIDFVectorizer$ to $ed_i$;
    Store vectorised document as Email corpus $\mathcal{C}$;
end for
Initialise an $LDAModel$ object;
Train on the Email corpus $\mathcal{C}$;
Initialise two empty dictionaries $\mathcal{TD}$ and $\mathcal{N}$;
for each Email document $ed_i \in \mathcal{ED}$ do
    Apply $get\_document\_topics()$ to $ed_i$;
    Return a temporary topic list $\mathcal{TD}$;
    for each topic $td_j \in \mathcal{TD}$ do;
        Apply $TFIDFVectorizer$ to the corresponding keyword list $\mathcal{KW}$;
        Compute average TF-IDF value $\alpha$ for $\mathcal{KW}$;
        Get a probability $p_j$ for the topic;
        if $p_j < \theta$ then /*$\theta$ is defined in Equation 6.1*/
            Remove topic $td_j$ from $\mathcal{TD}$;
        end if
    end for
    Append the number of topics $n_i$ to $\mathcal{N}$;
    Initialise a $TextSegment$ object;
    Apply $get\_segment\_texts()$ to $ed_i$ with $n_i$ as input;
    Return a topic segment list $\mathcal{TS}$;
    for each topic segment $ts_j \in \mathcal{TS}$ do;
        Apply $TFIDFVectorizer$ to the corresponding token list $\mathcal{TW}$;
        Compute average TF-IDF value $\beta$ for $\mathcal{TW}$;
        Compute cosine similarity between $\alpha$ and $\beta$;
        Assign $\mathcal{TW}$ to $td_j$;
    end for
end for
```

6.3.2.1 Document and topic representation

Word embedding is a technique that maps terms into numeric vectors to precisely capture semantic information and contextual similarity for text mining tasks (Mikolov et al., 2013). To obtain document and topic representations for input, all topic segments are padded to length l using padding tokens and dummy topic segments and topics are inserted to ensure that documents are represented by a fixed number of topic segments and topics.

Given a set of input $\mathcal{ED}$ containing n documents, each document is represented

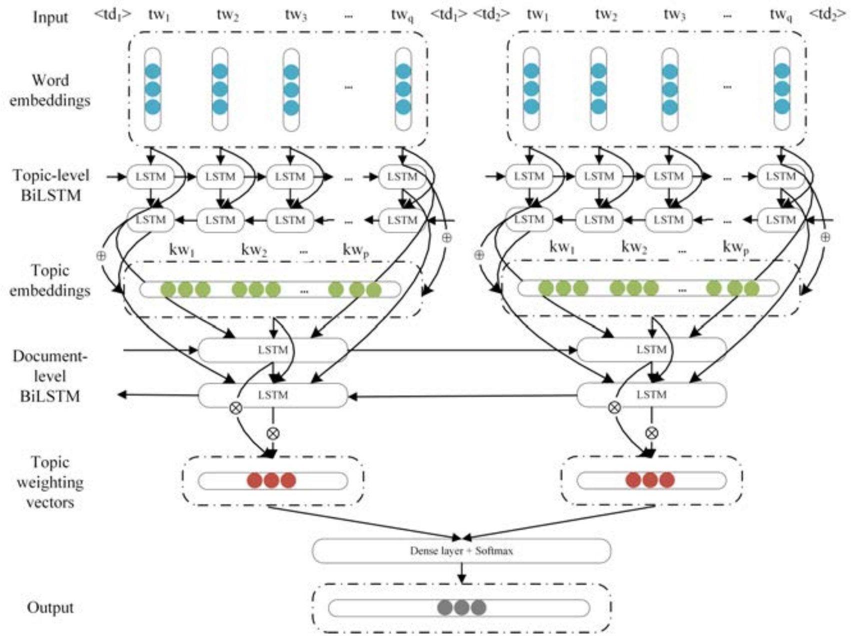


FIGURE 6.3: Overall model structure of MT-BiLSTM for document-level sentiment analysis. Given a sample document ed_i that has two topics $< td_1 >$ and $< td_2 >$. A topic-level BiLSTM is applied to each topic segment that is represented by word vectors $tw_1, tw_2, tw_3, ..., tw_q$ with length q. A time-distributed representation of topic segments is concatenated with a topic embedding layer that is represented by keyword vectors $kw_1, kw_2, ..., kw_q$ using $\oplus$ operator, and fed into a document-level BiLSTM. A probability distributed topic segment is further multiplied by a topic weighting vector using $\otimes$ operator, and fed into a final dense layer for output.

by a vectorized three-dimensional matrix denoted by $ed_i \in \mathbb{R}^{d \times l \times t}$ where d refers to the embedding dimensions of words, l refers to the maximum length of a topic segment, and t refers to the maximum number of topics in the corpus.

Each topic segment is associated with a topic represented by a fixed length of keywords p and a topic weighting vector w with length 1. Topic embeddings for each topic is calculated by averaging a dimension of d_t of word embeddings for all keywords $[\frac{\sum_{w=1}^{p} e_{w1}}{p} : \frac{\sum_{w=1}^{p} e_{wd_t}}{p}]$. Hence, given two sets of input $\mathcal{TL}$ and $\mathcal{W}$ containing n topic and weighting lists, respectively, each topic is represented by a vectorized two-dimensional matrix denoted by $tl_i \in \mathbb{R}^{d_t \times t}$ and each weighting is represented by a vectorized matrix denoted by $w_i \in \mathbb{R}^t$.

6.3.2.2 Bidirectional LSTM

The LSTM network (Hochreiter and Schmidhuber, 1997) is an extended variant of a traditional feed-forward neural network. The most apparent advantage of LSTM over other recurrent neural models is its ability to handle vanishing and exploding gradient problems. LSTM manages to capture long-term dependencies from sequentially-structured data by iteratively updating the memory state from a series of building blocks. Each building block centres a memory cell state that is updated by recurrent input information filtered by three functional gates using a sigmoid activation function. A *forget* gate manipulates the update of the current memory state by either forgetting or memorising the recurrent inputs, and an *input* gate and *output* gate control the flow of recurrent inputs by either erasing or keeping the current cell state (Hochreiter and Schmidhuber, 1997).

BiLSTM (Graves and Schmidhuber, 2005) was developed based on two LSTM layers that not only compute the hidden states of a forward sequence but also the hidden states of a backward sequence. By using two LSTM layers that proceed data in both directions, BiLSTM is capable of modelling the sequential dependencies of a piece of text from both the previous and successive contexts. Denote $\overrightarrow{\mathcal{H}}$ as a series of hidden states $[h_1, h2, \ldots, h_t]$ generated by a forward sequence, and $\overleftarrow{\mathcal{H}}$ as a series of hidden states $[h_t, h_{t-1}, \ldots, h_1]$ generated by a backward sequence. A BiLSTM computes the output sequence V_t at a given time step t by concatenating a $\overrightarrow{h_T} \in \overrightarrow{\mathcal{H}}$ and a $\overleftarrow{h_T} \in \overleftarrow{\mathcal{H}}$, which is mathematically denoted as:

$$\begin{aligned} V_t &= \overrightarrow{h_T} \oplus \overleftarrow{h_T} \\ &= W_{\overrightarrow{h_v}} \cdot \overrightarrow{h_T} + W_{\overleftarrow{h_v}} \cdot \overleftarrow{h_T} + b_V, \end{aligned} \tag{6.3}$$

where W refers to a weight matrix, and b refers to a bias vector for the corresponding input hidden vector.

6.3.2.3 Document-level multi-topic Bi-LSTM

In the proposed model, a topic-level BiLSTM is first applied to each topic segment represented by word embeddings. The result is two sequences of hidden vectors denoted by two matrices $\overrightarrow{\mathcal{H}_{ts}} \in \mathbb{R}^{h \times l}$ and $\overleftarrow{\mathcal{H}_{ts}} \in \mathbb{R}^{h \times l}$ where h is the size of hidden layers and l is the length of the given topic segment. Then, a topic embedding matrix $\mathcal{E}_t \in \mathbb{R}^{d_t}$ is concatenated to the final hidden state of both forward sequence $\overrightarrow{h_{ts}}$ and backward sequence $\overleftarrow{h_{ts}}$. The output vector $\mathcal{H}_t$ is given by:

$$\mathcal{H}_t = [\overrightarrow{h_{ts}} \oplus \mathcal{E}_t, \overleftarrow{h_{ts}} \oplus \mathcal{E}_t], \tag{6.4}$$

where $\mathcal{H}_t \in \mathbb{R}^{(h+d_t)\times 2}$ is a vector representation of each topic segment concatenated with topic embeddings in a document.

Subsequently, I apply a document-level BiLSTM to each document represented by topic segment vectors, resulting in two sequences of hidden vectors denoted by matrices $\overrightarrow{\mathcal{H}_T} \in \mathbb{R}^{(h+d_t)\times 2\times t}$ and $\overleftarrow{\mathcal{H}_T} \in \mathbb{R}^{(h+d_t)\times 2\times t}$. Finally, a $softmax$ layer is implemented to output the probability distribution of each weighted topic segment scaled by a topic weighting vector w_i to the overall sentiment:

$$\begin{aligned} \mathcal{H}_d &= [\overrightarrow{\mathcal{H}_T} \otimes w_i, \overleftarrow{\mathcal{H}_T} \otimes w_i], i \in (1, t), \\ y &= softmax(W_d * \mathcal{H}_d + b_d), \end{aligned} \tag{6.5}$$

where $\mathcal{H}_d \in \mathbb{R}^{h+d_t}$, and $\otimes$ reflect a point-wise multiplication operator that multiplies the topic weighting value w_i with each element in the matrix $\overrightarrow{\mathcal{H}_T}$ and $\overleftarrow{\mathcal{H}_T}$, and W and b refer to a weight matrix and a bias vector for the $softmax$ function, respectively.

6.4 Empirical experiments

In this section, I describe the preparation and adjustment of the datasets and parameter settings used for the different techniques under study. As Email sentiment classification is rarely studied, experimental results are reported to compare the proposed neural classification model with various widely adopted techniques at different levels, involving lexicon-based, machine learning, and deep learning approaches. Additionally, I justify the options used with the term weighting techniques and the parameters involved in LDA topic modelling by conducting a comparative analysis of the classification performance of the proposed model.

6.4.1 Datasets

For benchmarking purposes, I undertook quantitative evaluation and analysis with three publicly available labelled datasets and their corresponding augmented datasets (described in Section 3.2 and 3.3). The class distribution of each dataset can be found in Table 3.1 and 3.4.

6.4.2 Comparative methods

Effectiveness evaluations of the proposed MT-BiLSTM model and its variants were conducted via experiments with a set of comparative methods involving recent approaches to Email sentiment classification (Chhaya et al., 2018; Ezpeleta et al., 2016; Liu and Lee, 2018), lexical and machine learning-based baseline approaches, and state-of-the-art neural network-based approaches to document sentiment classification. Three machine learning-based algorithms, including baseline, Senti$\mathcal{TRACLUS}$ and SVM were selected based on their outperformance over other comparative methods in the same category as discussed in Chapter 5. Three basic neural network models, including MLP, LSTM and CNN were selected to provide benchmarking results and three advanced models, including BiLSTM, H-BiLSTM and HAN, were selected for their outstanding performance reported by recent studies. A brief description of each method is presented below.

- **Baseline**: A lexical-based approach that predicts sentiment polarities by computing BoW-weighted SWN lexicon features. An Email document is classified as positive if the final weighted value is above zero, negative if below zero and neutral if equal to zero.

- **Senti$\mathcal{TRACLUS}$** (Liu and Lee, 2018): A sequence-based approach that performs clustering on documents transformed into sentiment trajectories using a revised TRACLUS algorithm developed by Liu and Lee (2018). A final sentiment value is assigned to each cluster generated by the Senti$\mathcal{TRACLUS}$ algorithm and then grouped into a class of three.

– **SVM** (Chang and Lin, 2011): A benchmarked supervised-learning approach that yields the best performance in comparison to others. Aggregated word embeddings and sentiment sequences are generated as training features. The approach yields the best results of all supervised learning techniques, as reported in Chapter 5 Table 5.2;

– **MLP** (Gardner and Dorling, 1998): A classic feed-forward neural network model with three layers of perceptrons controlled by a nonlinear activation function.

– **LSTM** (Hochreiter and Schmidhuber, 1997): An extended variant of a traditional feed-forward neural network model that operates on a series of building blocks that contain a memory cell state and three multiplicative gates. A hidden state of 100 is set as the input parameter.

– **CNN** (Kim, 2014): A classic variant of conventional deep neural networks that implements convolutional filters with learned weights and bias. A window size of $[3, 4, 5]$ with a filter size of 32 for each convolutional layer is defined as an input parameter.

– **SSE-CNN**: A sequence-encoded CNN model that was proposed in Chapter 5. Detailed structure of the model can be referred in Section 5.3.

– **BiLSTM** (Graves and Schmidhuber, 2005): A bidirectional LSTM developed by Graves and Schmidhuber (2005) that has a concatenated layer of one forward LSTM and one backward LSTM and a hidden state of 100.

– **H-BiLSTM** (Ruder et al., 2016): A hierarchical-based bidirectional LSTM that is composed of a sentence-level BiLSTM layer and a document-level BiLSTM layer. Both layers are set to a hidden state of 100.

– **HAN** (Yang et al., 2016b): A hierarchical attention network based on a hierarchical structure with a bidirectional Gated Recurrent Unit(GRU) and attention mechanism at both word- and sentence-level. A hidden state of 100 is set for both bidirectional GRU layers and attention layers.

TABLE 6.1: Hyperparameter settings for using the proposed MT-BiLSTM with three datasets.

Parameter	BC3	EnronFFP	PA
maxTS	214	105	98
maxNT	4	3	3
hidden state	100	150	100
dropout probability	0.3	0.5	0.3
learning rate	0.01	0.01	0.01
batch size	32	64	32
num epochs	15	50	30

6.4.3 Experimental settings

Two sets of parameters are involved in the proposed model. Experimental results with different parameters of LDA topic modelling are reported in the following section. Apart from that, Table 6.1 summarises the hyperparameter settings of the neural network models for each Email dataset. Note that the maximum length of a topic segment is $maxTS$, and the maximum number of topics $maxNT$ varies with the different input parameters used in the LDA model. The empirical results reported here were generated based on 10 topics with the $TF - IDF$ term weighting method for the topic model, where a truncated $maxTS$ performed better than the original maximum length of topic segments. The same $batchsize$ and $numepochs$ of each dataset are used in all deep learning-based models. Additionally, the pre-trained GloVec model Pennington et al., 2014 is employed with a dimension of 100 for both word embeddings and topic embeddings, considering its adequate coverage and moderate processing time.

The empirical results were evaluated with 10-fold cross-validation in view of the moderate size of the datasets. Considering the same reason as explained in Section 5.4.3 that accuracy and error rate are more straightforward performance evaluation measures, the evaluation criteria are accuracy and RMSE (formulated as per Equation 4.6, which were averaged from 10 sets of experiments as standard matrices for multi-class classification tasks. The effectiveness of the proposed model is justified, as it produced higher accuracy and lower RMSEs than the other approaches.

6.4.4 Classification results

The foremost group of classification performance is presented and profiled on the basis of our proposed model with three base variations: including $Topic - BiLSTM$ for MT-BiLSTM model without topic embeddings and topic weighting vectors, $Topic - TE - BiLSTM$ for topic embeddings incorporated MT-BiLSTM models, and $Topic - TW - BiLSTM$ for topic weighting incorporated MT-BiLSTM models, compared with other algorithms described in Section 6.4.3. Table 6.2 concludes the performance of different algorithms for three datasets respectively where the outcomes of our proposed model and its three variants are opted from the best among different parameter settings of the LDA topic model.

TABLE 6.2: Overall performance of the various methods under study. The symbol $*$ indicates the best result from various experimental settings. Bold text highlights important results.

Dataset Classifier	**BC3**		**EnronFFP**		**PA**	
	Accuracy	RMSE	Accuracy	RMSE	Accuracy	RMSE
Baseline	0.373	1.262	0.205	1.216	0.308	1.016
SentiTRACLUS(Liu and Lee, 2018)	0.592	0.876	0.579	0.714	0.793	0.397
SVM(Chang and Lin, 2011)	0.612	0.818	0.603	0.657	0.637	0.656
MLP(Gardner and Dorling, 1998)	0.789	0.506	0.582	0.651	0.649	0.607
LSTM(Hochreiter and Schmidhuber, 1997)	0.852	0.461	0.586	0.652	0.588	0.642
CNN(Kim, 2014)	0.852	0.461	0.598	0.634	0.653	0.606
BiLSTM(Graves and Schmidhuber, 2005)	0.873	0.512	0.742	0.552	0.788	0.442
H-BiLSTM(Ruder et al., 2016)	0.874	0.512	0.739	0.574	0.817	**0.396**
HAN(Yang et al., 2016b)	0.861	0.556	0.721	0.621	0.742	0.508
SSE-CNN	**0.886**	**0.323**	**0.743**	**0.522**	**0.821**	0.422
Topic-BiLSTM $*$	0.903	0.317	0.770	0.472	0.841	0.377
Topic-TE-BiLSTM $*$	0.913	**0.282**	0.781	0.459	0.852	0.359
Topic-TW-BiLSTM $*$	0.897	0.319	0.779	0.470	0.850	0.372
MT-BiLSTM $*$	**0.918**	0.295	**0.788**	**0.439**	**0.859**	**0.355**

The same one-tail paired t-tests as used in Section 5.4.3.1 were utilised to test for significant differences. The major findings presented in Table 6.2 can be summarised as the following three points:

1. First, the proposed MT-BiLSTM model obtained the highest accuracy rate of 91.8%, 78.8% and 85.9% for the BC3, Enron FFP and PA dataset, respectively. Though the lowest RMSE value of 0.282 for the BC3 dataset was acquired by

$Topic - TE - BiLSTM$ model, $MT - BiLSTM$ model manages to achieve the lowest RMSE value of 0.439 and 0.355 for the rest two datasets. These empirical results demonstrate the effectiveness of our proposed MT-BiLSTM model in terms of Email document sentiment classification.

2. Second, the hypothesis that Email sentiments can be classified through a document-level multi-topic approach is upheld; the proposed model achieved significantly better performance over $\mathcal{SSE} - \mathcal{CNN}$, obtaining the best results among all existing state-of-the-art methods. Its accuracy was 3.2%, 4.5% and 3.8% higher for the BC3, Enron FFP and PA, respectively. The significance tests for differences in accuracy (p = 0.005) and RMSE (p = 0.034) show that $MT - BiLSTM$ has remarkably better classification performance than $\mathcal{SSE} - \mathcal{CNN}$.

3. Last, the observation that all topic-based BiLSTM models ($Topic - BiLSTM$, $Topic - TE - BiLSTM$, $Topic - TW - BiLSTM$ and $MT - BiLSTM$) perform better than other methods is further validation that topic-based neural network models incorporating topic-related features accurately predict sentiments at document-level and provide better classification performance than other document-based algorithms. For instance, the base variation of the proposed model $Topic - BiLSTM$ acquired accuracy rates of 90.3%, 77.0% and 84.1%, and RMSEs of 0.317, 0.472 and 0.377 for the BC3, Enron FFP and PA datasets, respectively. This outperforms all baseline methods, such as SVM (p = 0.013 for accuracy and p = 0.038 for RMSE), $H - BiLSTM$ (p = 0.002 for accuracy and p = 0.087 for RMSE), and $SSE - CNN$ (p = 0.009 for accuracy and p = 0.068 for RMSE).

The proposed method is composed of a preprocessing phase, a document segmentation phase and a neural classification phase where the document segmentation phase further contains an LDA topic modelling phase and a semantic text segmentation phase. Their Big-O time complexities are: $O(n)$, $O(nmt)$, $O(sr + skr)$, $O((s + k + 1)^r)$, respectively, where n represents the number of Email documents, m represents the number of words in Email documents, t represents the number of initial topics, s represents the number of topic segments, r represents the number of filtered topics and k represents the number of topic keywords. The space complexity for neural sentiment classification phase is $O(s^h dr + k^d r)$, where h represents the number of hidden states and d represents the dimension of word embeddings.

Two additional groups of experiments were conducted to investigate the effects of using the revised data augmentation technique and of using LDA topic modelling with different parameter settings. These provided further evaluation of the overall framework proposed in this study.

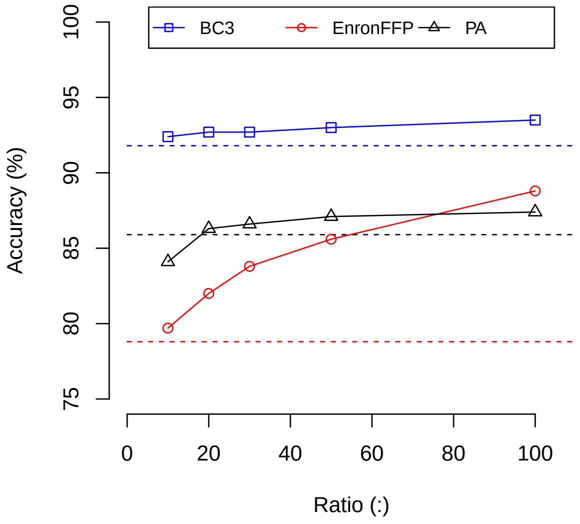
6.4.4.1 Effect of Email data augmentation

To evaluate the classification performance of the algorithms when run on the original and augmented datasets, I undertook two sets of experiments; one to compare the proposed MT-BiLSTM model with its variants, and the other to compare the use of different augmentation ratios with the proposed method. Representative classification results with data augmentation are presented in Table 6.3 and Figure 6.4 with a detailed discussion of the findings made in this section.

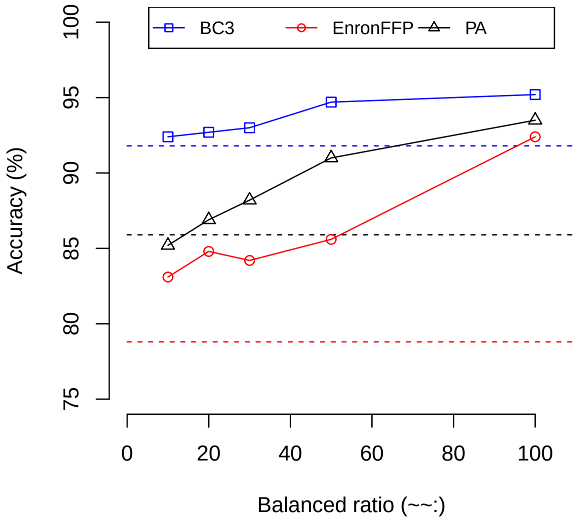
TABLE 6.3: Performance comparison of topic-based neural network models with original and augmented datasets. Results for augmented datasets were achieved using a ratio of $100:1$ to its original.

Dataset / Model	BC3				Enron FFP				PA			
	Original		**Augmented**		**Original**		**Augmented**		**Original**		**Augmented**	
	Accuracy	RMSE	Accuracy	RMSE	Accuracy	RMSE	Accuracy	RMSE	Accuracy	RMSE	Accuracy	RMSE
Topic-BiLSTM	0.903	0.317	0.934	0.347	0.770	0.472	0.797	0.468	0.841	0.377	0.859	0.375
Topic-TE-BiLSTM	0.913	0.282	0.935	0.270	0.781	0.459	0.859	0.405	0.852	0.369	0.858	0.377
Topic-TW-BiLSTM	0.897	0.319	0.931	0.291	0.779	0.470	0.824	0.522	0.850	0.372	0.870	0.361
MT-BiLSTM	0.918	0.295	0.935	0.259	0.788	0.439	0.888	0.434	0.859	0.355	0.874	0.354

As shown in Table 6.3, topic-based neural network models achieved better performance with augmented datasets than original datasets. For example, $MT-BiLSTM$ produced accuracy rates of 93.5%, 88.8% and 87.4%, which was equivalent to increases of 1.7%, 10.0% and 1.5% for each rate on all three datasets. In terms of statistical evaluations, t-test results indicated that there were significant increases in accuracy when using data augmentation (p = 0.004 for BC3 and p =



(A) Accuracy over ratios.



(B) Accuracy over balanced ratios.

FIGURE 6.4: Classification accuracy with regard to different levels of augmentation, where dot lines indicate the benchmark values of original datasets: a) the number of ratios to be augmented; b) the number of balanced ratios to be augmented.

0.008 for PA). Furthermore, Figure 6.4 illustrates the classification accuracy of the MT-BiLSTM model with augmented data, both balanced and imbalanced, at different ratios. According to the results, data augmentation has a remarkable positive influence on the performance of neural network models, with balanced augmented data outperforming imbalanced data.

6.4.4.2 Effect of LDA topic modelling with different parameter settings

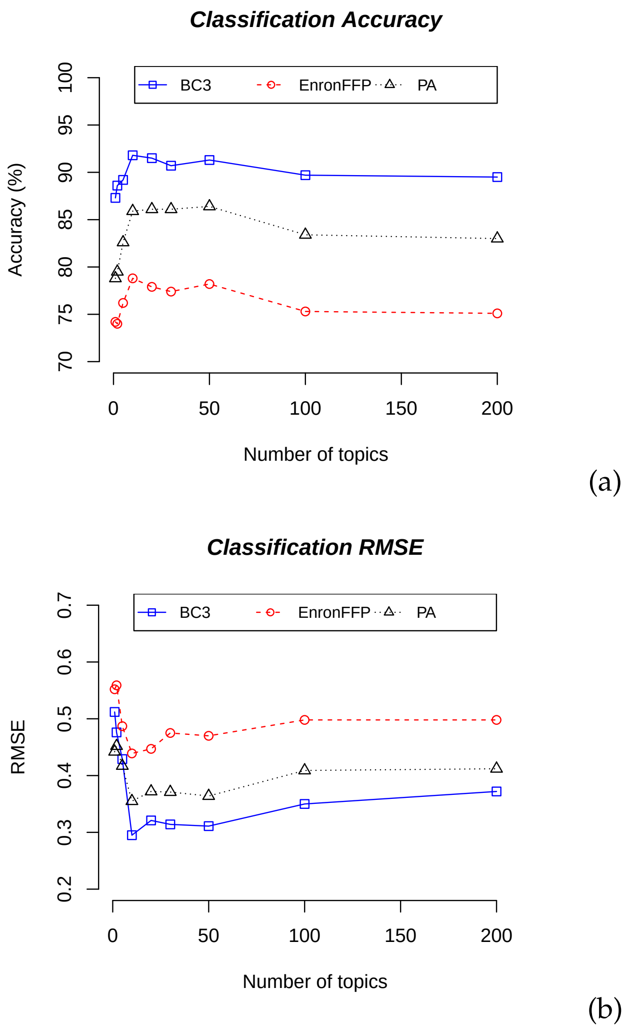
As the proposed MT-BiLSTM model is notably dependent on the topic-level inputs generated by LDA topic modelling, an in-depth analysis of the experiments comparing the two parameters that influence the outputs of the LDA model was undertaken. I first illustrate the comparative results of different term weighting methods, which mainly influence the input topic weighting vectors for the MT-BiLSTM model. Since different term weighting methods generate varied features as inputs for the LDA model, different topic weighting vectors and topic distributions were obtained accordingly. Table 6.4 summarises the classification performance of the MT-BiLSTM model with different term weighting methods, including $TF - IDF$, $n - gram$, $w2v$, in which $TF - IDF$ was the final option for the proposed model as it yielded the best results.

TABLE 6.4: Classification performance with different term weighting methods. Bold text highlights the important results.

Method	**BC3**		**Enron FFP**		**PA**	
	Accuracy	RMSE	Accuracy	RMSE	Accuracy	RMSE
TF-IDF	**0.918**	**0.295**	**0.788**	**0.439**	**0.859**	**0.355**
n-gram	0.837	0.476	0.729	0.574	0.790	0.497
w2v	0.891	0.325	0.777	0.450	0.852	0.420

I then evaluated the influence of LDA topic modelling (with different numbers of topics as an input parameter) on the classification performance of the MT-BiLSTM model with all three datasets and a fixed number of keywords 10 for the topic embeddings, considering the size of the vocabulary. Figure 6.5 compares the performance of the LDA model with input numbers of topics within the interval $[1, 2, 5, 10, 20, 30, 50, 100, 200]$ where the results of an input topic number of 1 are equivalent to those of the BiLSTM model.

As shown in Figure 6.5, LDA with an input topic number of 10 achieved the highest accuracy rates, of 91.8% and 78.8%, and the lowest RMSEs, of 0.295 and 0.439, on the BC3 and EnronFFP datasets, respectively. Although with the PA dataset the highest accuracy rate of 86.1% was acquired with a topic number of 20, the corresponding RMSE was not the lowest. Judging by the efficiency and



(a)

(b)

FIGURE 6.5: Classification performance in relation to the number of topics in terms of (a) accuracy and (b) RMSE.

effectiveness, a topic number of 10 was ultimately chosen for reporting the overall classification results of all three datasets.

6.5 Conclusion

To undertake document-level multi-topic sentiment classification of Email data, an MT-BiLSTM model was introduced to model structural dependencies at the topic-level within documents, using document segmentation based on multi-topic features. LDA topic modelling was utilised with semantic text segmentation to transfer documents into topic segments, where each topic segment is associated with a topic representation and probability distribution. Along with documents represented by topic segments, topic embeddings and topic weighting vectors obtained during the LDA modelling process were utilised as additional inputs for the proposed model. A topic-level BiLSTM concatenated with topic embeddings was applied to generate a vector representation of topic segments, and a document-level BiLSTM scaled by a topic weighting vector was applied to generate a weighted probability distribution of each topic segment for output.

Empirical experiments demonstrated that the proposed model produced higher classification accuracies and lower error rates than all other comparative algorithms. This proves the effectiveness of the proposed model in Email document classification. The results also indicate that topic-based models have an advantage over conventional document-based models. In addition, I conducted further evaluation of the effects of parameter settings on LDA topic modelling to quantitatively justify the options used in the model. The results indicate that the proposed algorithm provided the best performance on the three tested Email datasets with different initial numbers of topics. This implies that the proposed method is relatively dependent on the input parameters used in the LDA topic model. Hence, one possible improvement to the proposed method could be to incorporate an automatic searching algorithm to determine the appropriate input parameters for the LDA topic model. Moreover, as LDA topic modelling operates on a Gibbs sampler, which estimates the posterior probability of the topic distribution by iteratively sampling the topic assignments of training documents, adjusting the LDA topic modelling phase of the proposed method to handle Email documents with unseen topics is another potential improvement.

7 Conclusion

This chapter makes concluding remarks on the entire study. The thesis chapters are summarised and the research hypotheses are reviewed. The main contributions, findings and limitations of the research are described, and potential future research directions are highlighted.

7.1 Summary of thesis chapters

A summary of thesis chapters is made by reviewing the hypotheses and evaluating the evidence that supports them from each chapter.

The hypotheses relevant to Chapter 3 were:

> *1.1 Email cleaning with text normalisation will reduce the impact of noise and unstructured content and positively influence classification performance.*
>
> *1.2 Data augmentation will solve the scarcity and imbalanced class distribution issues that are common to labelled Email data.*
>
> *1.3 Supervised learning techniques and neural network models will provide better classification performance with augmented datasets than non-augmented ones.*

Chapter 3 presented an overall framework for document-level sentiment analysis of Email data and outlined methods involved in the preprocessing stage, involving data augmentation, Email cleaning and text normalisation. In detail, Hypothesis 1.1 was tested through the utilisation of pre-developed Email cleaning packages and natural language processing functions to standardise and normalise raw data before feature generation. The tests of significance shown in Table 5.3 provide evidence to support this hypothesis, as algorithms performed better with preprocessed data than with raw data. Hypotheses 1.2 and 1.3 were tested by implementing data augmentation with a random word replacement technique that uses a k-NN classifier trained on word embeddings and a WN lexicon. The experimental results summarised in Tables 5.4 and 6.3 indicate that, compared with unsupervised learning techniques, supervised learning techniques and neural network models provide remarkably better classification performance with augmented data. Moreover, Figures 5.6 and 6.4 present data that further support the hypotheses, as balanced class distributions provided higher classification accuracy than imbalanced ones.

The hypotheses relevant to, and examined in, Chapter 4 were:

> *2.1 Sentiment sequence features can be embedded in sentiment trajectories built from Email documents and captured through sentiment trajectory representation.*
>
> *2.2 Sentiment sequence features will contribute positively to classification performance and can be discovered through a trajectory clustering approach.*

In Chapter 4 I proposed an unsupervised sequence-based approach for Email sentiment clustering and sentiment sequence discovery. In detail, Hypothesis 2.1 was tested by representing Email documents with a set of features involving sentiment lexicons, categories and timestamps, and by converting feature-represented Email data into sentiment trajectories using pseudo-longitude and latitude transformation and pixel conversion. Senti$\mathcal{TRACLUS}$, a revised TRACLUS algorithm that outputs sentiment sequences and polarities, was developed to perform clustering analysis on sentiment trajectories. Figure 4.3 presented two frequent sentiment sequences obtained from the Senti$\mathcal{TRACLUS}$ algorithm that support the aforementioned two hypotheses. Moreover, the time and space complexity discussed in Section 4.5.3, along with the results in Figure 4.3, further indicate the capability and efficiency of the Senti$\mathcal{TRACLUS}$ algorithm in discovering sentiment sequence patterns in Email data as stated in Hypothesis 2.2. Lastly, the sample sentiment clustering results from Email messages in categorical and temporal groups, presented in Tables 4.2 and 4.3, support the practical usefulness of categorical and temporal classification in visualising sentiment patterns, as described in Hypothesis 2.2.

The hypotheses relevant to, and examined in, Chapter 5 were:

> *3.1 Sentiment sequences can be encoded through position and sentiment lexical features.*
>
> *3.2 Sentiment sequence-encoded CNN models will provide better classification performance than baseline, unsupervised learning and supervised learning approaches.*
>
> *3.3 Algorithms with sentiment sequence features will provide better classification performance than algorithms without them.*

In Chapter 5, I introduced a revised CNN model for Email sentiment classification, with sentiment sequences encoded by an LSTM model based on position and SWN features. The most important part of the model, position features, were extracted for an exploration of DG- and PT-based position-encoding methods. The experimental results discussed in Section 5.4.3.3 support Hypothesis 3.1, as the dependency-graph-based position-encoding approach yielded better classification performance than the plain-text-based approach, and the approach using sentiment sequences encoded by an LSTM model outperformed the approaches using sequence encoding and position features. In support of Hypothesis 3.2, the empirical results presented in Section 5.4.3.1 indicate that, generally, neural network models outperformed the baseline, unsupervised and supervised learning approaches. Specifically, word embedding and the proposed $\mathcal{SSE} - \mathcal{CNN}$ model obtained better results than the other algorithms on all three datasets. Additionally, Hypothesis 3.3 was supported by the same results where, for each category, sequence-incorporated approaches (Senti$\mathcal{TRACLUS}$, SVM $_{aggwe+ssf}$ and $\mathcal{SSE} - \mathcal{CNN}$) yielded better results than non-sequence-incorporated approaches in the same category and, specifically, SVM with sentiment sequence features outperformed SVM with traditional BoW features.

The hypotheses relevant to, and examined in, Chapter 6 were:

> *4.1 LDA topic modelling and semantic text segmentation techniques can effectively model the multi-topic features of Email documents.*
>
> *4.2 Topic weighting and topic features generated by the LDA topic modelling will improve the performance of polarity classification.*
>
> *4.3 Multi-topic features will positively contribute to classification performance and MT-BiLSTM will outperform other sentence- or document-level neural network models.*

In Chapter 6 I designed a topic-weighted BiLSTM model for document-level multi-topic Email sentiment classification that uses LDA topic mdoelling and semantic text segmentation. The proposed MT-BiLSTM model was built on three inputs: multi-topic represented documents, topic embeddings and topic weightings. Hypothesis 4.1 was tested by representing documents and topics as word embeddings for input to the MT-BiLSTM model. The empirical results presented in Table 6.2 support Hypotheses 4.2 and 4.3, as all multi-topic feature-incorporated neural network models exhibited better classification performance than other comparative approaches. Moreover, BiLSTM with topic embeddings and weightings performed better than its other variants, which further supports these hypotheses.

7.2 Summary of research contributions

The contributions of this research can be categorised into technical and empirical contributions. The primary technical contribution of this research is the development of a framework for document-level Email sentiment analysis that efficiently analyses sentiment sequences and effectively classifies sentiments in Email data. Answers to the four research questions based on the framework were explored by conducting studies on 1) sentiment sequence clustering, 2) sequence-encoded neural sentiment classification and 3) multi-topic neural sentiment classification. A summary of Chapters 4 to 6 that addresses Research Questions 2 to 4 is provided next.

- **Chapter 4**. *"Research Question 2: How to effectively capture sentiment sequence features and discover sentiment sequence patterns within Email data?"* This question was answered by modelling Email documents as sentiment trajectories that can be compiled using trajectory clustering methods. A three-phase trajectory representation method was designed to convert textual Emails into sentiment trajectory representations. The Senti$\mathcal{TRACLUS}$ algorithm, which is an adapted TRACLUS algorithm, was developed for sentiment sequence discovery. In addition, a categorical and temporal classification phase was devised to obtain readable sentiment polarity results and assist in visualising sentiment sequence patterns.

- **Chapter 5**. *"Research Question 3: How to encode sentiment sequence features in a neural network model for robust and accurate sentiment polarity classification?"* This question was answered by modelling Email documents as word embeddings and sentiment sequence-encoded representations. A position encoding method was developed based on dependency graph-based position features weighted by discourse depth. Position features were aggregated by sentiment lexical features generated from an SWN lexicon, then encoded into sentiment sequences using an LSTM layer. These served as an additional input for the $\mathcal{SSE} - \mathcal{CNN}$ model, which is a revision of the classic CNN model, for polarity classification.

- **Chapter 6**. *"Research Question 4: How to capture multi-topic features and model documents with multi-topic segments for effective sentiment polarity classification?"* This question was answered by modelling Email documents as topic segments to serve as inputs for the $MT - BiLSTM$ model based on the traditional BiLSTM model. A document segmentation method based on LDA topic modelling and semantic text segmentation was introduced to generate documents that were represented by topic segments, keywords and weights. Textual topic segments and topic keywords were vectorised by word embeddings and fed into the $MT - BiLSTM$ model for polarity classification.

The empirical contributions of this research are mainly reflected in the following three aspects. The first contribution was to answer "*Research Question 1: What preprocessing methods are essential in addressing unstructured and noisy contents in Email data and can solve the issues of data scarcity and imbalanced class distributions in labelled Emails?*" This was achieved by conducting empirical experiments that compared the performance of analyses conducted with cleaned, augmented and raw datasets. The second contribution is the three labelled Email datasets that were used for classification. As discussed in the previous sections, one of the challenges of Email sentiment analysis is the lack of ground-truth data. Though two of the three benchmark Email datasets used in this research originated from public sources, they were not labelled in a way that suited document-level sentiment classification. Hence, a personal Email dataset was obtained and manually labelled with three sentiment polarities at the document-level. This dataset is publicly available[19] to those interested in using it for further analysis. The third contribution is a set of evaluation results on Email sentiment polarity classification using algorithms at different levels, including baseline, unsupervised learning, supervised learning and deep learning algorithms. These results can be served as a basic reference for comparison with future analysis techniques.

7.3 Summary of research findings

As illustrated earlier in this thesis, in Section 1.3, the main research problem identified was the design and development of the four functions, involving noise handling, sentiment sequence, sentiment classification and quantitative evaluation, contained in the document-level Email sentiment analysis framework. The four functions were accomplished through an exploration of the three studies conducted in this thesis and outlined in the summary of the main contributions in Section 7.2. A summary of the key research findings in terms of the four functions is presented as follows.

[19] http://doi.org/ 10.13140/RG.2.2.14545.68968/1

- **Noise handling.** The essentiality of noise handling for Email sentiment analysis was justified through a qualitative analysis on the Email distribution over Senti$\mathcal{TRACLUS}$ clustering results conducted in Chapter 4 and a classification evaluation on the effect of Email document cleaning in Chapter 5. In Section 4.5.2, the sentiment sequence clustering results of the proposed Senti$\mathcal{TRACLUS}$ algorithm on the real-life Enron dataset only had a 44.2% use ratio, which reflected the complex structure and noisy contents contained in Email data as the proposed method operated on a density-based function and was sensitive to outliers. Moreover, a comparative analysis on the classification performance of neural network models on raw and cleaned data discussed in Section 5.4.4.2 further proved the effectiveness of appropriate preprocessing for Email sentiment classification as neural network models obtained higher accuracy rates on cleaned data than raw data for all three benchmark datasets.

- **Sentiment sequence.** The existence of sentiment sequence within Email data and the feasibility of using a sequence-based clustering approach for Email sentiment classification was proved through a qualitative evaluation on the sentiment sequence within Email documents and a case study with labelled datasets in Chapter 4. The consistency of the trajectory clusters and sentiment features generated as shown in the tables in Section 4.5.3 indicated the presence of sentiment sequence features in Email documents. Though the classification results reported in Section 4.5.4.2 reflected that the proposed sequence-based clustering algorithm outperforms other comparative methods for some datasets, it failed to obtain satisfied performance with significance.

- **Sentiment classification.** The effectiveness of incorporating sentiment sequence and multi-topic features into the process of sentiment classification of Email data was substantiated through a quantitative evaluation on the methods proposed in Chapter 5 and 6. On the one hand, empirical results presented in Section 5.4.4 proved the effectiveness of the proposed $SSE - CNN$ that incorporates the dependency-graph based position features and relational information on classification performance. On the other hand, empirical results presented in Section 6.4.4 demonstrated an advantage of the proposed topic-based model $MT - BiLSTM$ over traditional document-based models. Furthermore, an evaluation on the effect of LDA topic modelling with results discussed in Section 6.4.4.2 implied an dependence of the $MT - BiLSTM$ model on the number of topics as an input for the LDA model.

- **Quantitative evaluation.** The overall quantitative evaluations proves the feasibility and effectiveness of the proposed framework for discovering sentiment sequence within Email data using sequence-based clustering approach and classifying Email sentiments with improved performance using sentiment sequence and multi-topic features. Additionally, to tackle the issues of data scarcity and imbalanced class distribution derived from the publicly available labelled datasets, a data augmentation method with random word replacement was implemented and proved its positive effect on the sentiment classification accuracy of Email data with results analysed in Section 5.4.4.2 and 6.4.4.1.

7.4 Limitations and future directions

For the final remarks of the thesis, I highlight some limitations of this research. Based on these, potential topics for future research are identified and discussed. Four potential topics are:

1. **Empirical experiments with larger genuine Email datasets.** In this study, three medium-sized Email datasets were used in the empirical experiments. Only the PA dataset was initially annotated with appropriate labels for the purpose of document-level sentiment polarity classification. Though proper label conversion methods were implemented for the other two datasets with conceptual and technical support from the literature, the effects of these label conversion methods are untested, as this was beyond the scope of this thesis. Moreover, though I experimented with mitigating the influences of data scarcity and imbalanced class distributions, the ability of a data augmentation

method to fully synthetic the complexity linguistic and syntactic structure in textual information is questionable. Hence, future work could explore various label conversion approaches to performance improvement and apply them to larger genuine Email messages originating from a wider range of users.

2. **Exploration of the wider utilisation of meta-information for analysis.** In Chapter 5, the experimental results indicated that with the visual support of meta-information, e.g., categories and timestamps, sentiment sequence patterns are more interpretable and meaningful. Moreover, as thread detection, which mainly extracts useful information from meta-information, is one of the most popular tasks in Email mining, it is reasonable to expect that gaining insights into Email communication patterns by detecting threads before conducting sentiment analysis will further improve classification performance. Hence, future work could investigate the possibility of incorporating meta-information as features for sentiment classification and visualisation.

3. **Exploration of potential applications of the proposed framework to real-time Email systems.** The literature review indicated that Email summarisation and visualisation are two of the most commonly adopted tasks in Email mining. Some interactive visual analytical systems have been developed with restricted sentiment analysis functionality. As a systematic and comprehensive framework for document-level sentiment analysis of Email data was developed and validated via empirical experiments, applying the framework to real Email systems is the logical next step. Additionally, future work could also focus on refining the proposed framework with user labelling functionality so that more labelled Email data could be continuously supplied to the system to better train it.

4. **Exploration of fine-grained Email sentiment classification.** As briefly discussed in Chapter 2, some studies have utilised a fine-grained set of emotional state labels to more closely model real-world scenarios. Email communication might well fit into one of these scenarios. As the proposed framework was initially built for multi-class classification, it can be easily

adjusted to suit the needs of fine-grained sentiment classification tasks. Therefore, future work could aim to increase the granularity of sentiment classification by exploring the Email-specific sentiment labelling systems and features that are most appropriate for capturing sentiment traits based on labels.

Bibliography

Abbasi, Ahmed, Hsinchun Chen, and Arab Salem (2008). "Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums". In: *ACM Transactions on Information Systems (TOIS)* 26.3, p. 12.

Alm, Cecilia Ovesdotter, Dan Roth, and Richard Sproat (2005). "Emotions from text: machine learning for text-based emotion prediction". In: *Proceedings of the conference on human language technology and empirical methods in natural language processing*. Association for Computational Linguistics, pp. 579–586.

Aqil Burney, Badar Sami et al. (2012). "Urdu text summarizer using sentence weight algorithm for word processors". In: *International Journal of Computer Applications* 975, p. 8887.

Baccianella, Stefano, Andrea Esuli, and Fabrizio Sebastiani (2010). "Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining." In: *Lrec*. Vol. 10. 2010, pp. 2200–2204.

Bao, Jun-Peng et al. (2004). "Semantic sequence kin: A method of document copy detection". In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, pp. 529–538.

Barberis, Nicholas, Andrei Shleifer, and Robert Vishny (1998). "A model of investor sentiment". In: *Journal of financial economics* 49.3, pp. 307–343.

Berk, Elliot and C Scott Ananian (2005). "JLex: A lexical analyzer generator for Java (TM)". In: *Department of Computer Science, Princeton University. Version* 1.

Bermingham, Luke and Ickjai Lee (2015). "A general methodology for n-dimensional trajectory clustering". In: *Expert Systems with Applications* 42.21, pp. 7573–7581.

Bespalov, Dmitriy et al. (2012). "Sentiment classification with supervised sequence embedding". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, pp. 159–174.

Bhatia, Parminder, Yangfeng Ji, and Jacob Eisenstein (2015). “Better document-level sentiment analysis from rst discourse parsing”. In: *arXiv preprint arXiv:1509.01599*.

Blanzieri, Enrico and Anton Bryl (2008). “A survey of learning-based techniques of email spam filtering”. In: *Artificial Intelligence Review* 29.1, pp. 63–92.

Blei, David M, Andrew Y Ng, and Michael I Jordan (2003). “Latent dirichlet allocation”. In: *Journal of machine Learning research* 3.Jan, pp. 993–1022.

Bogawar, Pranjal S and Kishor K Bhoyar (2012). “Email mining: a review”. In: *IJCSI International Journal of Computer Science Issues* 9.1, pp. 429–434.

Bottou, Léon (2010). “Large-scale machine learning with stochastic gradient descent”. In: *Proceedings of the 19th International Conference on Computational Statistics*. Springer, pp. 177–186.

Breiman, Leo (2001). “Random forests”. In: *Machine learning* 45.1, pp. 5–32.

Caragea, Cornelia et al. (2014). “Mapping moods: geo-mapped sentiment analysis during hurricane Sandy”. In: *Proc. of ISCRAM*.

Chang, Chih-Chung and Chih-Jen Lin (2011). “LIBSVM: a library for support vector machines”. In: *ACM transactions on intelligent systems and technology (TIST)* 2.3, p. 27.

Chen, Huimin et al. (2016). “Neural sentiment classification with user and product attention”. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1650–1659.

Chhaya, Niyati et al. (2018). “Frustrated, Polite, or Formal: Quantifying Feelings and Tone in Email”. In: *Proceedings of the Second Workshop on Computational Modeling of People’s Opinions, Personality, and Emotions in Social Media*, pp. 76–86.

Coletto, Mauro et al. (2016). “Sentiment-enhanced Multidimensional Analysis of Online Social Networks: Perception of the Mediterranean Refugees Crisis”. In: *arXiv preprint arXiv:1605.01895*.

Collobert, Ronan et al. (2011). “Natural language processing (almost) from scratch”. In: *Journal of Machine Learning Research* 12.Aug, pp. 2493–2537.

Das, Sanjiv and Mike Chen (2001). “Yahoo! for Amazon: Extracting market sentiment from stock message boards”. In: *Proceedings of the Asia Pacific finance association annual conference (APFA)*. Vol. 35. Bangkok, Thailand, p. 43.

Das, Sanjiv R, Seoyoung Kim, and Bhushan Kothari (2019). “Zero-Revelation RegTech: Detecting Risk through Linguistic Analysis of Corporate Emails and News”. In: *The Journal of Financial Data Science* 1.2, pp. 8–34.

Dave, Kushal, Steve Lawrence, and David M Pennock (2003). “Mining the peanut gallery: Opinion extraction and semantic classification of product reviews”. In: *Proceedings of the 12th international conference on World Wide Web*, pp. 519–528.

David, James P and Jerry Suls (1999). “Coping efforts in daily life: Role of Big Five traits and problem appraisals.” In: *Journal of personality*.

Dehiya, Vasundhara and Klaus Mueller (2016). “Analyzing Hillary Clinton’s Emails”. In: *Poster Abstracts of IEEE VIS*.

Diakopoulos, Nicholas, Mor Naaman, and Funda Kivran-Swaine (2010). “Diamonds in the rough: Social media visual analytics for journalistic inquiry”. In: *2010 IEEE Symposium on Visual Analytics Science and Technology*. IEEE, pp. 115–122.

Dill, Jody C and Craig A Anderson (1995). “Effects of frustration justification on hostile aggression”. In: *Aggressive Behavior* 21.5, pp. 359–369.

Dredze, Mark et al. (2008). “Generating summary keywords for emails using topics”. In: *Proceedings of the 13th international conference on Intelligent user interfaces*. ACM, pp. 199–206.

Eckman, Paul (1972). “Universal and cultural differences in facial expression of emotion”. In: *Nebraska symposium on motivation*. Vol. 19, pp. 207–284.

Ezpeleta, Enaitz, Urko Zurutuza, and José María Gómez Hidalgo (2016). “Does sentiment analysis help in bayesian spam filtering?” In: *International Conference on Hybrid Artificial Intelligence Systems*. Springer, pp. 79–90.

Fukuhara, Tomohiro, Hiroshi Nakagawa, and Toyoaki Nishida (2007). “Understanding Sentiment of People from News Articles: Temporal Sentiment Analysis of Social Events.” In: *ICWSM*.

García-Hernández, René Arnulfo, José Francisco Martínez-Trinidad, and Jesús Ariel Carrasco-Ochoa (2006). “A new algorithm for fast discovery of maximal sequential patterns in a document collection”. In: *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, pp. 514–523.

Gardner, Matt W and SR Dorling (1998). “Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences”. In: *Atmospheric Environment* 32.14-15, pp. 2627–2636.

Goldstein, Jade et al. (2006). “Annotating Subsets of the Enron Email Corpus.” In: *CEAS*.

Goyvaerts, Jan and Steven Levithan (2012). *Regular expressions cookbook*. O’reilly.

Graves, Alex and Jürgen Schmidhuber (2005). “Framewise phoneme classification with bidirectional LSTM and other neural network architectures”. In: *Neural networks* 18.5-6, pp. 602–610.

Hall, Mark et al. (2009). “The WEKA data mining software: an update”. In: *ACM SIGKDD explorations newsletter* 11.1, pp. 10–18.

Hangal, Sudheendra, Monica S Lam, and Jeffrey Heer (2011). “Muse: Reviving memories using email archives”. In: *Proceedings of the 24th annual ACM symposium on User interface software and technology*. ACM, pp. 75–84.

Harris, Sarah and David Harris (2015). *Digital design and computer architecture: arm edition*. Morgan Kaufmann.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). “Long short-term memory”. In: *Neural computation* 9.8, pp. 1735–1780.

Jindal, Nitin and Bing Liu (2006). “Identifying comparative sentences in text documents”. In: *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, pp. 244–251.

Khan, Farhan Hassan, Usman Qamar, and Saba Bashir (2016). “SWIMS: Semi-supervised subjective feature weighting and intelligent model selection for sentiment analysis”. In: *Knowledge-Based Systems* 100, pp. 97–111.

Khan, Farhan Hassan, Usman Qamar, and Saba Bashir (2017). “A semi-supervised approach to sentiment analysis using revised sentiment strength based on SentiWordNet”. In: *Knowledge and information Systems* 51.3, pp. 851–872.

Kim, Soo-Min and Eduard Hovy (2004). “Determining the sentiment of opinions”. In: *Proceedings of the 20th international conference on Computational Linguistics*. Association for Computational Linguistics, p. 1367.

Kim, Yoon (2014). “Convolutional neural networks for sentence classification”. In: *arXiv preprint arXiv:1408.5882*.

Koven, Jay et al. (2016). “InVEST: Intelligent visual email search and triage”. In: *Digital Investigation* 18, S138–S148.

Kumar, Vipin and Sonajharia Minz (2013). “Mood classifiaction of lyrics using SentiWordNet”. In: *Computer Communication and Informatics (ICCCI), 2013 International Conference on*. IEEE, pp. 1–5.

Kundi, Fazal Masud et al. (2014). “Detection and scoring of internet slangs for sentiment analysis using SentiWordNet”. In: *Life Science Journal* 11.9, pp. 66–72.

Lee, Jae-Gil, Jiawei Han, and Kyu-Young Whang (2007). “Trajectory clustering: a partition-and-group framework”. In: *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*. ACM, pp. 593–604.

Lewis, David D (1998). “Naive (Bayes) at forty: The independence assumption in information retrieval”. In: *European conference on machine learning*. Springer, pp. 4–15.

Li, Bofang et al. (2015). “Learning document embeddings by predicting n-grams for sentiment classification of long movie reviews”. In: *arXiv preprint arXiv:1512.08183*.

Li, Wei-Jen, Shlomo Hershkop, and Salvatore J Stolfo (2004a). “Email archive analysis through graphical visualization”. In: *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, pp. 128–132.

Li, Yifan, Jiawei Han, and Jiong Yang (2004b). “Clustering moving objects”. In: *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 617–622.

Li, Zhenhui et al. (2010). “Incremental clustering for trajectories”. In: *International Conference on Database Systems for Advanced Applications*. Springer, pp. 32–46.

Liu, Bing (2012). “Sentiment analysis and opinion mining”. In: *Synthesis lectures on human language technologies* 5.1, pp. 1–167.

Liu, Bing et al. (2010). “Sentiment analysis and subjectivity.” In: *Handbook of natural language processing* 2.2010, pp. 627–666.

Liu, Sisi and Ickjai Lee (2015). “A hybrid sentiment analysis framework for large email data”. In: *Intelligent Systems and Knowledge Engineering (ISKE), 2015 10th International Conference on*. IEEE, pp. 324–330.

Liu, Sisi and Ickjai Lee (2018). “Discovering sentiment sequence within email data through trajectory representation”. In: *Expert Systems with Applications* 99, pp. 1–11.

Liu, Sisi, Ickjai Lee, and Guochen Cai (2016). “Sentiment Clustering with Topic and Temporal Information from Large Email Dataset”. In: *Proceedings of the 30th Pacific Asia Conference on Language, Information and Computation: Posters*, pp. 363–371.

Maas, Andrew L et al. (2011). “Learning word vectors for sentiment analysis”. In: *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*. Association for Computational Linguistics, pp. 142–150.

Majumder, Navonil et al. (2017). “Deep learning-based document modeling for personality detection from text”. In: *IEEE Intelligent Systems* 32.2, pp. 74–79.

Manning, Christopher D et al. (2014). “The stanford corenlp natural language processing toolkit.” In: *ACL (System Demonstrations)*, pp. 55–60.

Mao, Yi and Guy Lebanon (2007). “Isotonic conditional random fields and local sentiment flow”. In: *Advances in neural information processing systems*, pp. 961–968.

Matsumoto, Shotaro, Hiroya Takamura, and Manabu Okumura (2005). “Sentiment classification using word sub-sequences and dependency sub-trees”. In: *Pacific-Asia conference on knowledge discovery and data mining*. Springer, pp. 301–311.

McCandless, Michael, Erik Hatcher, and Otis Gospodnetic (2010). *Lucene in action: covers Apache Lucene 3.0*. Manning Publications Co.

McCarroll, Danny (2016). *Simple statistical tests for geography*. Chapman and Hall/CRC.

Mikolov, Tomas et al. (2013). “Distributed representations of words and phrases and their compositionality”. In: *Advances in neural information processing systems*, pp. 3111–3119.

Miller, George A (1995). “WordNet: a lexical database for English”. In: *Communications of the ACM* 38.11, pp. 39–41.

Mitchell, Tom M et al. (1997). *Machine learning*.

Mohammad, Saif M and Tony Wenda Yang (2011). “Tracking sentiment in mail: How genders differ on emotional axes”. In: *Proceedings of the 2nd workshop on computational approaches to subjectivity and sentiment analysis*. Association for Computational Linguistics, pp. 70–79.

Moraes, Rodrigo, JoãO Francisco Valiati, and Wilson P GaviãO Neto (2013). “Document-level sentiment classification: An empirical comparison between SVM and ANN”. In: *Expert Systems with Applications* 40.2, pp. 621–633.

Mukherjee, Arjun and Bing Liu (2012). “Aspect extraction through semi-supervised modeling”. In: *Proceedings of the 50th annual meeting of the association for computational linguistics: Long papers-volume 1*. Association for Computational Linguistics, pp. 339–348.

Nakagawa, Tetsuji, Kentaro Inui, and Sadao Kurohashi (2010). “Dependency tree-based sentiment classification using CRFs with hidden variables”. In: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 786–794.

Nasukawa, Tetsuya and Jeonghee Yi (2003). “Sentiment analysis: Capturing favorability using natural language processing”. In: *Proceedings of the 2nd international conference on Knowledge capture*. ACM, pp. 70–77.

Nightingale, Deborah J et al. (2008). *Origin of Email & Misuses of the Term “Email”*.

Oelke, Daniela et al. (2009). “Visual opinion analysis of customer feedback data”. In: *Visual Analytics Science and Technology, 2009. VAST 2009. IEEE Symposium on*. IEEE, pp. 187–194.

Onan, Aytug, Serdar Korukoglu, and Hasan Bulut (2016). “LDA-based Topic Modelling in Text Sentiment Classification: An Empirical Analysis”. In: *Int. J. Comput. Linguistics Appl.* 7.1, pp. 101–119.

Pang, Bo, Lillian Lee, et al. (2008). “Opinion mining and sentiment analysis”. In: *Foundations and Trends® in Information Retrieval* 2.1–2, pp. 1–135.

Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan (2002). “Thumbs up?: sentiment classification using machine learning techniques”. In: *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics, pp. 79–86.

Pennington, Jeffrey, Richard Socher, and Christopher Manning (2014). "Glove: Global vectors for word representation". In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543.

Perkins, Jacob (2014). *Python 3 text processing with NLTK 3 cookbook*. Packt Publishing Ltd.

Pestian, John P et al. (2012). "Sentiment analysis of suicide notes: A shared task". In: *Biomedical informatics insights* 5.Suppl. 1, p. 3.

Poria, Soujanya et al. (2016). "Sentic LDA: Improving on LDA with semantic similarity for aspect-based sentiment analysis". In: *2016 international joint conference on neural networks (IJCNN)*. IEEE, pp. 4465–4473.

Proskurnia, Julia et al. (2017). "Template Induction over Unstructured Email Corpora". In: *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, pp. 1521–1530.

Rao, Guozheng et al. (2018). "LSTM with sentence representations for document-level sentiment classification". In: *Neurocomputing* 308, pp. 49–57.

Ravi, Kumar and Vadlamani Ravi (2015). "A survey on opinion mining and sentiment analysis: tasks, approaches and applications". In: *Knowledge-Based Systems* 89, pp. 14–46.

Rehurek, Radim and Petr Sojka (2010). "Software framework for topic modelling with large corpora". In: *In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Citeseer.

Ruder, Sebastian, Parsa Ghaffari, and John G Breslin (2016). "A hierarchical model of reviews for aspect-based sentiment analysis". In: *arXiv preprint arXiv:1609.02745*.

Saura, Jose Ramon and Dag R Bennett (2019). "A Three-Stage method for Data Text Mining: Using UGC in Business Intelligence Analysis". In: *Symmetry* 11.4, p. 519.

Scholkopf, Bernhard et al. (1997). "Comparing support vector machines with Gaussian kernels to radial basis function classifiers". In: *IEEE transactions on Signal Processing* 45.11, pp. 2758–2765.

Scott, Sam and Stan Matwin (1999). "Feature engineering for text classification". In: *ICML*. Vol. 99. Citeseer, pp. 379–388.

Sharaff, Aakanksha and Naresh Kumar Nagwani (2016). "Email thread identification using latent Dirichlet allocation and non-negative matrix factorization based clustering techniques". In: *Journal of Information Science* 42.2, pp. 200–212.

Shen, Jianqiang, Oliver Brdiczka, and Juan Liu (2013). "Understanding email writers: Personality prediction from email messages". In: *International Conference on User Modeling, Adaptation, and Personalization*. Springer, pp. 318–330.

Spicer, Dag (2016). "Raymond Tomlinson: Email Pioneer, Part 2". In: *IEEE Annals of the History of Computing* 38.3, pp. 78–83.

Sukhbaatar, Sainbayar, Jason Weston, Rob Fergus, et al. (2015). "End-to-end memory networks". In: *Advances in neural information processing systems*, pp. 2440–2448.

Tai, Chih-Hua et al. (2015). "Mental disorder detection and measurement using latent Dirichlet allocation and SentiWordNet". In: *Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on*. IEEE, pp. 1215–1220.

Tang, Duyu (2015). "Sentiment-specific representation learning for document-level sentiment analysis". In: *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*. ACM, pp. 447–452.

Tang, Duyu, Bing Qin, and Ting Liu (2015a). "Document modeling with gated recurrent neural network for sentiment classification". In: *Proceedings of the 2015 conference on empirical methods in natural language processing*, pp. 1422–1432.

Tang, Duyu, Bing Qin, and Ting Liu (2015b). "Learning semantic representations of users and products for document level sentiment classification". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Vol. 1, pp. 1014–1023.

Tang, Guanting, Jian Pei, and Wo-Shun Luk (2014). "Email mining: tasks, common techniques, and tools". In: *Knowledge and Information Systems* 41.1, pp. 1–31.

Tang, Huifeng, Songbo Tan, and Xueqi Cheng (2009). "A survey on sentiment detection of reviews". In: *Expert Systems with Applications* 36.7, pp. 10760–10773.

Tang, Jie et al. (2005). "Email data cleaning". In: *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. ACM, pp. 489–498.

Tomlinson, Ray (2009). "The first network email". In: *Site de Ray Tomlinson*.

Turney, Peter D (2002). "Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews". In: *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, pp. 417–424.

Ulrich, Jan, Gabriel Murray, and Giuseppe Carenini (2008). "A publicly available annotated corpus for supervised email summarization". In: *Proc. of aaai email-2008 workshop, chicago, usa*.

Wade, James B et al. (1990). "An emotional component analysis of chronic pain". In: *Pain* 40.3, pp. 303–310.

Wang, Hongning, Yue Lu, and Chengxiang Zhai (2010). "Latent aspect rating analysis on review text data: a rating regression approach". In: *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 783–792.

Wang, Leyi and Rui Xia (2017). "Sentiment lexicon construction with representation learning based on hierarchical sentiment supervision". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 502–510.

Wang, Shoujin et al. (2016). "Training deep neural networks on imbalanced data sets". In: *2016 international joint conference on neural networks (IJCNN)*. IEEE, pp. 4368–4374.

Wang, William Yang and Diyi Yang (2015). "That's so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using# petpeeve tweets". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 2557–2563.

Wanner, Franz et al. (2009). "Visual sentiment analysis of rss news feeds featuring the us presidential election in 2008". In: *VISSW*.

Wei, Chih-Ping and Yu-Hsiu Chang (2007). "Discovering event evolution patterns from document sequences". In: *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* 37.2, pp. 273–283.

Wei, Jason W and Kai Zou (2019). "Eda: Easy data augmentation techniques for boosting performance on text classification tasks". In: *arXiv preprint arXiv:1901.11196*.

Wiebe, Janyce, Rebecca Bruce, and Thomas P O'Hara (1999). "Development and use of a gold-standard data set for subjectivity classifications". In: *Proceedings of the 37th annual meeting of the Association for Computational Linguistics*, pp. 246–253.

Williams, RT (1995). "Lambert and Mercator map projections in geology and geophysics". In: *Computers & Geosciences* 21.3, pp. 353–364.

Wilson, Theresa et al. (2005). "OpinionFinder: A system for subjectivity analysis". In: *Proceedings of HLT/EMNLP 2005 Interactive Demonstrations*.

Wu, Desheng Dash, Lijuan Zheng, and David L Olson (2014). "A decision support approach for online stock forum sentiment analysis". In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 44.8, pp. 1077–1087.

Wu, Yanzhao et al. (2019). "A comparative measurement study of deep learning as a service framework". In: *IEEE Transactions on Services Computing*.

Wu, Yingcai et al. (2010). "OpinionSeer: interactive visualization of hotel customer feedback". In: *IEEE transactions on visualization and computer graphics* 16.6, pp. 1109–1118.

Yang, Yunlun et al. (2016a). "A position encoding convolutional neural network based on dependency tree for relation classification". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 65–74.

Yang, Zichao et al. (2016b). "Hierarchical attention networks for document classification". In: *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pp. 1480–1489.

Yao, Xiaobai (2003). "Research issues in spatio-temporal data mining". In: *Workshop on Geospatial Visualization and Knowledge Discovery, University Consortium for Geographic Information Science, Virginia*, pp. 1–6.

Yi, Jeonghee et al. (2003). "Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques". In: *Third IEEE international conference on data mining*. IEEE, pp. 427–434.

Yin, Yichun, Yangqiu Song, and Ming Zhang (2017). "Document-level multi-aspect sentiment classification as machine comprehension". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2044–2054.

Yoo, Shinjae et al. (2009). “Mining social networks for personalized email prioritization”. In: *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 967–976.

Yu, Yang, Wenjing Duan, and Qing Cao (2013a). “The impact of social and conventional media on firm equity value: A sentiment analysis approach”. In: *Decision Support Systems* 55.4, pp. 919–926.

Yu, Yanwei et al. (2013b). “Online clustering for trajectory data stream of moving objects”. In: *Computer science and information systems* 10.3, pp. 1293–1317.

Zhang, Lei, Shuai Wang, and Bing Liu (2018). “Deep learning for sentiment analysis: A survey”. In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 8.4, e1253.

Zhang, Xiang, Junbo Zhao, and Yann LeCun (2015). “Character-level convolutional networks for text classification”. In: *Advances in neural information processing systems*, pp. 649–657.

Zhang, Ye and Byron Wallace (2015). “A sensitivity analysis of (and practitioners’ guide to) convolutional neural networks for sentence classification”. In: *arXiv preprint arXiv:1510.03820*.

Zheng, Yu (2015). “Trajectory data mining: an overview”. In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 6.3, p. 29.