Context-Aware Sentence Categorisation: Word Mover's Distance and Character-level Convolutional Recurrent Neural Network

by Xinyu FU

Thesis submitted to The University of Nottingham
for the degree of Doctor of Philosophy, January
2018

Contents

List of Figures

List of Tables

Publications

1. X. Fu, E. Ch'ng, U. Aickelin, and S. See, "CRNN: a joint neural network for redundancy detection," in *Smart Computing (SMARTCOMP), 2017 IEEE International Conference on*, pp. 1–8, IEEE, 2017

2. X. Fu, E. Ch'ng, U. Aickelin, and L. Zhang, "An improved system for sentence-level novelty detection in textual streams," in *Smart and Sustainable City and Big Data (ICSSC), 2015 International Conference on*, pp. 1–6, IET, 2015

Abstract

Supervised k nearest neighbour and unsupervised hierarchical agglomerative clustering algorithm can be enhanced through word mover's distance-based sentence distance metric to offer superior context-aware sentence categorisation performance. Advanced neural network-oriented classifier is able to achieve competing result on the benchmark streams via an aggregated recurrent unit incorporated with sophisticated convolving layer.

The continually increasing number of textual snippets produced each year necessitates ever improving information processing methods for searching, retrieving, and organising text. Central to these information processing methods are sentence classification and clustering, which have become an important application for natural language processing and information retrieval. This present work proposes three novel sentence categorisation frameworks, namely hierarchical agglomerative clustering-word mover's distance, k nearest neighbour-word mover's distance, and convolutional recurrent neural network. Hierarchical agglomerative clustering-word mover's distance employs word mover's distance distortion function to effectively cluster unlabelled sentences into nearby centroid. K nearest neighbour-word mover's distance classifies testing textual snippets through word mover's distance-based sentence similarity. Both models are from the spectrum of count-based framework since they apply term frequency statistics when building the vector space matrix.

Experimental evaluation on the two unsupervised learning data-sets show better performance of hierarchical agglomerative clustering-word mover's distance over other competitors on mean squared error, completeness score, homogeneity score, and v-measure value. For k nearest neighbour-word mover's distance, two benchmark textual streams are experimented to verify its superior classification performance against comparison algorithms on precision rate, recall ratio, and F1 score. Performance comparison is statistically validated via Mann-Whitney-U test. Through extensive experiments and results analysis, each research hypothesis is successfully verified to be yes.

Unlike traditional singleton neural network, convolutional recurrent neural network model incorporates character-level convolutional network with character-aware recurrent neural network to form a combined framework. The proposed model benefits from character-aware convolutional neural network in that only salient features are selected and fed into the integrated character-aware recurrent neural network. Character-aware recurrent neural network effectively learns long sequence semantics via sophisticated update mechanism. The experiment presented in current thesis compares convolutional recurrent neural network framework against the state-of-the-art text classification algorithms on four popular benchmarking corpus. The present work also analyses three different recurrent neural network hidden recurrent cells' impact on performance and their runtime efficiency. It is observed that minimal gated unit achieves the optimal runtime and comparable performance against gated recurrent unit and long short-term memory. For term frequency-inverse document frequency-based algorithms, the current experiment examines word2vec, global vectors for word representation, and sent2vec embeddings and reports their performance differences. Performance comparison is statistically validated through Mann-Whitney-U test and the corresponding hypotheses are tested to be yes through the reported statistical analysis.

Acknowledgements

I would like to express the deepest appreciation to my supervisors, Professor Eugene Ch'ng and Professor Uwe Aickelin who have supported and mentored me over three years in the pursuit of an area of interest I found fascinating many years ago.

Great thanks go to Mr. Liming Xu and Mr. Yichen Pan for their valuable comments on the theoretical investigation of the thesis. Thanks to all the members of the International Doctoral Innovation Centre (IDIC) group, Horizon Doctoral Training Centre at the University of Nottingham for the friendship and support.

I would also love to thank Dr. Simon See, Nvidia Technology Centre Asia Pacific and Japan, Ningbo Science and Technology Bureau, University of Nottingham Ningbo Campus, and UK campus for their generous financial support on my PhD lifespan and academic conferences attending.

My sincere gratitude goes to my Mum and Dad for encouraging me and guiding me in the pursuit of learning.

My thanks go to my girlfriend Chunan. Without her continuous support and patience this would not have been possible.

Chapter 1

Introduction

In this chapter, recent advancement on the natural language processing related fields and image classification motivate the present research. The scope and objectives of the work and defined research problems that the current thesis intends to solve are then presented. Derived research hypotheses are also given to direct benchmarking data-sets, comparison algorithms, and elements of experiments, which give supporting evidence to verify corresponding questions. Finally, an overview of the thesis structure and the chapter conclusion are shown.

Natural language mining and understanding has been a long-standing challenge and human language is a reflection of human intelligence and engages with a huge amount of grammatical rules. Such nature of human language implies that machine encoding of it will be a tough issue. Natural language processing involves natural language interpretation, semantics, sentence similarity, and information categorisation. This thesis devotes to a specific problem of natural language processing: sentence categorisation. It refers to classifying unlabelled sentences into pre-defined class or clustering forthcoming textual snippets into nearest centroid.

Natural language processing has various applications and strong linkage to machine learning, statistical analysis, information retrieval, language generation, generative modelling, and so on. Knowledge representation and comprehension, as a critical component for many machine learning tasks, has always been a central

problem in the field of natural language processing. As for a good representation of documents, an established form is to use bag-of-words-oriented vectorisation mechanisms. Another widely adopted method is generative topic models, such as latent semantic analysis and latent dirichlet allocation [1].

Within information retrieval, first story detection is a highly relevant topic towards sentence categorisation. First story detection research has shed some lights on the development of text mining, especially on sentence similarity/dissimilarity sector. Sentence similarity measure relies on spatial distance. Word mover's distance-based distance function has roots in information retrieval and first story detection. Traditional Euclidean distance, cosine distance, or Manhattan distortion metrics is deployed to many information retrieval applications such as context summarisation [2, 3], page ranking [4, 5], recommender systems [6, 7, 8], language modelling paradigms [9, 10], and event detection frameworks [11, 12, 13, 14]. Traditional approaches towards sentence classification rely on plain word counting statistics [14], n-gram techniques [15], and language modelling frameworks.

This present work proposes an utilisation of character-level sentence encoding with convolutional neural network to sift the most salient features because of its prevalence on deep neural network and wide applications to various domains like online text classification and clustering [16, 17, 10]. Character-level convolutional neural network also adds confidence to identifying discriminate classes by learning subword information, misspelling or grammatical errors. Furthermore, deep learning models has seen significant improvement on tackling hidden semantics or structure of sentences, paragraphs, and documents [18]. These issues are likely to happen in a human derived works such as newswire streams and social media data sources [16, 17, 10]. By feeding the most atomic characters in the deep neural model, grammatical errors and misspelling keywords can be interpreted in such a way that appropriate linguistic knowledge can be learned. The systematic comparison against the four challenging benchmarking data-sets demonstrates the high level of performance obtained through the proposed model in the current thesis.

Plain convolutional neural network, however, suffers from remembering long-term dependencies over sequences, as opposed to the traditional n-gram language modelling [19]. Nevertheless, according to Brown et al. [20], the system memory takes to store tri-gram phrases or above is enormous, which demands high spatial and time complexity. One possible solution to this is recurrent neural network [21, 16]. The embedded recurrent neural network hidden state and the non-linear activation function are capable of capturing long textual sequence probability distribution over iteration [21, 16]. By replacing the simple element-wise sigmoid activation function with a more complex long short-term memory, the integrity of expressing lengthy snippets can be further improved, in line with Hochreiter and Schmidhuber [22]. Cho et al. [21] proposed a gated recurrent unit which can dynamically reserve the state information through the pipeline of input signal. Convolutional recurrent neural network model benefits from Character-level convolutional neural network on the regularities feature-distilling and Character-aware recurrent neural network upon the long-term dependencies learning.

1.1 Motivation

With the growth of online data acquisition techniques and infrastructure, users have strong willingness to develop knowledge of the changing world. Research on sentence similarity identification began by traditional bag-of-words, term frequency-inverse document frequency matrix, and n-gram language modelling [14, 20]. Kusner et al. [23] and later Huang, Guo, and Kusner [24] propose the novel usage of word mover's distance distortion error on supervised sentence classification motivates the present thesis to investigates an emergent application on unsupervised environment. As hierarchical agglomerative clustering engages with pairwise geometric distance for building a dendrogram, which makes it a natural candidate for sentence similarity metrics [25]. It is thus a plausible solution to consider hierarchical agglomerative clustering with the latest word mover's distance distortion metrics.

Earth mover's distance is a distance function defined between probability distributions on a given metric space [26]. Earth mover's distance is normally used in image classification field [27] and gesture recognition [28]. Word mover's distance is a special variant of earth mover's distance particularly for sentence/document similarity measurement in the context of supervised learning [24, 23]. This thesis extends the practicality of word mover's distance to unsupervised environment by integrating it with the hierarchical agglomerative clustering algorithm. The clustering benchmarking data-sets are imported to evaluate its performance. Unlike the existing spatial distance based sentence dissimilarity metrics such as cosine, Euclidean, and Manhattan, word mover's distance offers more accurate precision with respect to semantics interpretation [23] and deeper interaction with dense word embeddings technique [29].

By involving the recent advances in fine-grained word embeddings like word2vec or global vectors for word representation with the advent novel word mover's distance metric, client users could compute the nuanced discrepancies through reflection on aggregating one word vector to its neighbours' dissimilarity. It produces accurate measurement when incorporating with the bag-of-words representations and imposes a positive influence on the sentence categorisation problem [11].

Following the success of deep neural networks in the ImageNet competition, there have been several inherited applications of deep architectures on tasks such as image classification, speech recognition, and natural language processing language modelling [30, 31, 32]. This paradigm has also been applied to text categorisation and sentence polarity analysis. In accordance with Krizhevsky et al. [33] and Collobert et al. [34], convolutional neural network has shown state-of-the-art performance on computer vision and varied natural language processing problems. Whilst traditional methods are good at encoding term association relationships (stock and market), neural models tend to preserve word similarity relationships (stock and security) [35]. The proposed convolutional recurrent neural network framework [36] is motivated by the aforementioned emergent applications of deep neural models on

text classification and natural language processing-related challenges.

1.2 Scope and Objectives

Text classification is the task of automatically assigning classes to sentences or documents. There exist several supervised classification algorithms that have achieved good results in text classification tasks (sentiment analysis, topic mining, etc.) such as SVM, latent dirichlet allocation, latent semantic analysis, locality sensitive hashing, or multi-layer perceptron [37, 38, 39, 40, 41]. While their primary use has been in image classification and speech recognition, deep learning techniques have recently been used for text classification and have achieved remarkable results. A text snippet is characterised by the words it contains, and consequently the representation of textual data is only based on its words. Thus, an important feature in text classification is the word vector representation of input data. Bag-of-words vectors representation is probably the simplest and most widely used representation where vectors indicate which words appear in the sentence without preserving word order. Bag-of-words word vectors subsumes three major variations, namely term frequency-inverse document frequency, count vectoriser, and binary one-hot vectorisation. Vectors from bag-of-words lack semantics and are usually huge and sparse. Alternative solutions have been proposed such as n-gram models, skip thoughts-based sentence vector [42, 43, 44] or wordnet [45, 46]. However, to be effective, models that use n-gram or skip-thought vectors require a huge dataset and sentences or words that are frequently observed [47]. The use of wordnet is also language dependant [47].

Word mover's distance-oriented sentence categorisation intends to investigate the helpfulness of word mover's distance in identifying the correct labels among the testing corpora. Other components of a traditional sentence categorisation including term representation, distance metrics, and word matrix dimensionality reduction techniques are taken into consideration when designing and evaluating the experiment. Standard evaluations on the participating algorithms are performed to testify

the effectiveness of word mover's distance on sentence classification or clustering.

Traditional count-based approaches to sentence categorisation rely on lots of feature engineering for designing, defining, and selecting the appropriate features [48]. Non-neural networks methods such as k nearest neighbour requires essential pre-defined word embeddings like global vectors for word representation or word2vec to formulate a vector space model matrix, which cannot scale well on unseen input data or biased stream. However, the proposed convolutional recurrent neural network model does not occupy a huge database for storing pre-defined word vectors, sparse sentence corpus, and is less sensitive to input language because of its instant training and optimisation. In other words, convolutional recurrent neural network is free from external knowledge base or dictionary when building the sentence representations. It instead decodes the input data through self-learning process.

As for the neural networks comparison and assessment, a list of algorithms including convolutional neural network, recurrent neural network, SVM, k nearest neighbour, and its variant are presented in the sentence classification experiment. Machine learning metrics such as precision rate, recall ratio, and F1 score are considered in the performance examination. System average runtime statistics is also provided in the estimation.

1.3 Research Problems and Derived Hypotheses

The broad problem domain this work tends to solve is twofold. First, this work studies the usefulness of word mover's distance on categorising unlabelled/unclassified sentences with respect to sentence similarity metric. Second, this thesis investigates deep neural model's effectiveness on improving sentence classification performance.

Research questions that this work raises and seeks to answer are summarised as follows.

(1) How does the expressiveness of term frequency-inverse document frequency differ from that of count vectoriser and binary one-hot vectori-

sation?

(2) Can word mover's distance-based distance function improve the accuracy of sentence similarity measurement over existing spatial distance functions?

(3) How does the performance of word mover's distance-based sentence categorisation models differ from that of the other competing algorithms?

(4) Can recurrent unit be leveraged to improve the convolutional neural network's performance in the context of sentence classification?

(5) How does the performance of long short-term memory differ from that of gated recurrent unit and minimal gated unit?

(6) How does the performance of convolutional recurrent neural network differ from that of the comparison frameworks for sentence classification?

In the following the stemmed research hypotheses are given with each research question deriving a list of corresponding hypotheses.

For the research question 'How does the expressiveness of term frequency-inverse document frequency differ from that of count vectoriser and binary one-hot vectorisation?', two hypotheses can be obtained.

(1) Given same quantity of sentences for vector space model matrix construction, binary one-hot vectorisation obtains the best time efficiency, followed by count vectorisation and term frequency-inverse document frequency.

(2) Term frequency-inverse document frequency vectoriser provides better vector space model representation than that of count vectoriser and binary one-hot vectorisation, leading to better sentence categorisation performance.

From the research question 'Can word mover's distance-based distance function improve the accuracy of sentence similarity measurement over existing spatial distance functions?', the following hypotheses can be generated:

(1) Given same benchmark data stream, word2vec dense embeddings offers better word vector representation than global vectors for word representation distributed embeddings scheme.

(2) Word mover's distance provides better sentence classification performance than earth mover's distance and relaxed word mover's distance over standard machine learning metrics on the specified supervised benchmark data-sets.

(3) Word mover's distance provides superior sentence clustering performance against earth mover's distance and relaxed word mover's distance over standard machine learning metrics on the specified unsupervised benchmark data-sets.

(4) Word mover's distance-based distance function improves the accuracy of sentence similarity measurement over cosine, Euclidean, and Manhattan on the specified benchmark textual streams.

The research question 'How does the performance of word mover's distance-based sentence categorisation models differ from that of the other competing algorithms?' implies the following two hypotheses:

(1) Given same testing sequences, k nearest neighbour-word mover's distance yields better performance than the other comparison frameworks with respect to precision rate, recall ratio, and F1 score.

(2) Given same testing sequences, hierarchical agglomerative clustering-word mover's distance achieves better mean squared error value, homogeneity score, completeness value, and v-measure than the competing algorithms in unsupervised learning.

The aforementioned three research questions and associated hypotheses will be tested or answered in Chapter 5. The next three presented research questions and derived hypotheses will be addressed in Chapter 6.

For the question 'Can recurrent unit be leveraged to improve the convolutional neural network's performance in the context of sentence classification?', the following hypothesis can be derived.

(1) Convolutional recurrent neural network offers better sentence classification performance than that of plain convolutional neural network.

For the question 'How does the performance of long short-term memory differ from that of gated recurrent unit and minimal gated unit?', the subsequent two hypotheses can be yielded.

(1) Long short-term memory, gated recurrent unit, and minimal gated unit offer statistically insignificant sentence categorisation performance when tested on same data-sets.

(2) Minimal gated unit requires lowest computational overhead, followed by gated recurrent unit and long short-term memory when constructing recurrent unit.

For the question 'How does the performance of convolutional recurrent neural network differ from that of the comparison frameworks for sentence classification?', the following hypothesis can be generated.

(1) Given identical hyper-parameters and benchmark data-sets, convolutional recurrent neural network model produces better sentence classification performance than that of the comparison frameworks.

1.4 Structure of the Thesis

An overview of the thesis is presented as follows. Existing literature in relation to sentence categorisation is presented in chapter 2. In chapter 3, the methodology

subsuming vector space model matrix compression techniques, generic sentence-level distortion functions, and two context-aware sentence categorisation frameworks is described. For the proposed novel sentence categorisation models, first the count-based sentence categorisation is presented including an unsupervised model and a supervised framework. Second, this chapter gives the model implementation details on neural network sentence classification, which includes an overview of several baseline models and the experimental configuration. Chapter 4 offers the corresponding benchmark data-sets with dedicated pre-processing approaches for count-based experiments and neural network performance evaluation respectively. Experimental setup for word mover's distance-based examination and convolutional recurrent neural network-oriented evaluation are shown respectively. Chapter 5 presents the results according to the word mover's distance oriented experiments, proves the related research hypotheses, offers several research findings, and discussions. Chapter 6 gives results and analysis on convolutional recurrent neural network-based performance evaluation, tests the correspondent hypotheses, and discusses the results. The final chapter includes summary, contributions, conclusions, application to the proposed novel sentence categorisation frameworks, limitations, and future directions of this work.

1.5 Conclusion

The chapter of introduction begins with an outlook of the related industry development for the elements of thesis motivation. The focused area of the present work and targeting application fields are described in the section of scope and objectives. Defined research questions and correspondent hypotheses are presented with linkage to the future chapters of presentation. Finally, the chapters distribution of the thesis is outlined. Next chapter is a review of literature, particularly on background of Topic Detection and Tracking, count-based sentence categorisation, neural network-based and probabilistic models for sentence classification, historical

view of language modelling, applications to vector space model, and context-aware recommender systems.

Chapter 2

Literature: A Review

In this chapter, various techniques, approaches, and frameworks relating to the field of sentence-level categorisation are presented. The chapter begins by the history of Topic Detection and Tracking, followed by three prominent components for developing sentence categorisation framework. The three ingredients are dense embeddings, feature weighting, and similarity calculation mechanisms. Plain first story detection based frameworks are normally unsupervised learning tasks which requires human annotation and exhaustive threshold learning [14, 11]. However, the work proposed in this thesis is free from manually labelling and threshold learning. A comprehensive view of neural networks and relevant natural language processing models is demonstrated. A detailed examination on neural network-based probabilistic language models is shown. Next, a collection of four efficient document representation manners are illustrated with application to sentence categorisation. This chapter is concluded by a summary on contribution to review of literature , followed by a description on possible research gap and how this work is going to address the gap.

First story detection was firstly introduced in signal processing research which seeks to identify novel signals. A report on methods for novelty detection can be acquired from the Signal Processing Journal by Karkali, Rousseau, Ntoulas, and Vazirgiannis [49]. The report has two separate sections: statistical methods and neural networks approaches.

Unlike the traditional first story detection or novelty detection, sentence categorisation is the process pairing text to preclassified labels through text similarity. Sentence categorisation often refers to supervised learning on pre-defined classes or unsupervised tasks on text similarity, whereas first story detection usually bases on novelty threshold learning. Effective document or sentence representation models are crucial to the real world sentence categorisation application. High dimensional vector space model occupies large memory space and therefore leads to high computational overhead. Many scholars and researchers have attempted various dimensionality reduction techniques and the thesis includes the ones related to sentence categorisation herein. For instance, vector quantisation tries to approximate the entire sequence with a subset of elements [50]. Non-negative Matrix Factorisation interprets a non-negative matrix with either multiplicative update or projected gradient methods [51, 52]. Because bag-of-words implies that every entry is non-negative, Non-negative Matrix Factorisation is suitable for supervised sentence classification tasks. Locality sensitive hashing [53] is an efficient and robust approximate nearest neighbor problem seeker which poses a positive impact on approximating high dimensional vector space model matrices. Locality sensitive hashing uses hyper-planes to identify sentence pairs with similar characteristics, describing similar topics or events for instance [53].

2.1 Background of Topic Detection and Tracking

An overview of the Topic Detection and Tracking evaluation, novelty detection specifically, in the field of information retrieval is investigated through the current section. The inaugural research activities, especially the pilot study initiative conducted during September 1996 through October 1997, under the Topic Detection and Tracking evaluation has established the Topic Detection and Tracking field. According to Allan et al. [54], Topic Detection and Tracking evaluation was also co-sponsored by the DARPA TIDES programme. The goal of Topic Detection and

Tracking is to pursue technological and technical advances on finding topically relevant snippets in a collection of newswire data [55]. There are five investigation options available in the Topic Detection and Tracking programme: namely first story detection, topic tracking, link detection, topic detection, and story segmentation [54]. First story detection, or novelty detection as an alternative, aims for identifying the first story that discuss a topic is the priority. Topic tracking focuses on detecting stories that describe a target topic. The goal of link detection is to detect whether a pair of stories discuss the same topic. Detecting clusters of stories that discuss the same topic is the goal of topic detection. For story segmentation, the task is to detect the boundaries of story.

2.1.1 Topic Detection and Tracking Corpus

Due to the ongoing demands on efficiently correlating or interpreting multitude informational sources, scholars had hosted seven consecutive open evaluations from Topic Detection and Tracking 1998 to Topic Detection and Tracking 2004. The Topic Detection and Tracking research also gleans and collates text corpora and they have been archived at the Linguistic Data Consortium platform. Multimedia sources and textual corpus have been demonstrated and distributed to the society in the annual Topic Detection and Tracking workshop in both English and Mandarin Chinese. For example, in accordance with Allan [56] and Gong and Xu [57], the Topic Detection and Tracking Pilot Study Corpus encompasses 15863 news events from July, 1994 until June, 1995. The newswire material from Reuters and the broadcast information from CNN constitute the integral part of the Topic Detection and Tracking Pilot Study Corpus, where the CNN source is formatted as the manual transcriptions. Moreover, the transcripts were generated by the Journal of Graphics Institute (JGI) and the stories are stored in chronological order with SGML format [54]. Topic Detection and Tracking Pilot Study Corpus defines a group of 25 events as the target topics. Later, an extended corpus called Topic Detection and Track-

ing2 corpus was organised to underpin the enlarged Topic Detection and Tracking community interest. The Topic Detection and Tracking2 corpus includes approximately 40000 individual stories and 1000 hours of audio based multimedia material, all recorded from January to June, 1998 [57]. The Topic Detection and Tracking2 corpus consists of 100 distinct classes or document labels, each represents a target news topic occurred in the first six months of 1998. The Topic Detection and Tracking2 corpus contains data contents from the two notable newswire agencies (AP WorldStream and New York Times Newservice), two renowned radio programme services provider PRI The World and VOA World News), two well-known television programmes (CNN Headline News and ABC World News Tonight), and three mainstream Mandarin news sources.

Furthermore, in line with Allan [56], the Topic Detection and Tracking3 corpus archives 45000 target news events from eight English newswire and broadcast agencies including three Chinese newswire inputs during the period of October to December 1998. The Topic Detection and Tracking3 corpus is annotated with 240 topics, 140 more classes when compared to the Topic Detection and Tracking2 corpora [14]. Each individual newswire event in either Topic Detection and Tracking2 or Topic Detection and Tracking3 corpus is labelled based on a triplet (YES, NO, BRIEF). YES simply means the story describes the specific topic. BRIEF is tagged if the story tells less than 10% of the topic, NO otherwise [56].

For the evaluation criteria employed by Topic Detection and Tracking researchers, several plans are worth considering. First, the normalised detection cost function: detection error trade-off curve was first appeared in a speaker recognition and language things detection scenario [58]. A detection error trade-off curve tries to cast the Topic Detection and Tracking tasks by measuring the probability of miss and false alarm rate (P_{miss} and P_{FA}), providing that the specified Topic Detection and Tracking task is evaluated as detection task. According to Alvin et al.'s evaluation [58], the detection error trade-off curve has significant advantages against the plain ROC curve rendering on the large vocabulary based speech recognition tasks and

language perception assessment. More specifically, a detection error trade-off curve explicitly involves the trade-offs of two error types whereas the traditional ROC curve does not [59].

Second, mutual information metric was developed for the Topic Detection and Tracking tasks evaluation by Gong and Xu [57]. Given two collections of documents D and D', their mutual information metric $MI(D, D')$ is represented as a joint logarithm probability function based on the cluster D and D'. The detailed definition can be accessed from [57]. A normalised variant was also mentioned in Gong and Xu [57] for a simplified comparison and is available at [57] as well.

For the participants attended in the Topic Detection and Tracking tasks evaluation, there were eleven candidates submitted their contest results during the 1998 Topic Detection and Tracking2 corpus evaluation competition [59]. GTE Internetworking's BBN Technologies (BBN), Columbia University (CIDR), Carnegie Mellon University (CMU), Dragon Systems (Dragon), General Electric (GE), IBM's T.J. Watson Laboratories (IBM), SRI International (SRI), University of Iowa (UIowa), University of Massachusetts (UMass), University of Maryland (UMd), and University of Pennsylvania (Upenn) were involved in the 1998 contest. Among those eleven participants, six are from the academic field, and the rest are corporational institutes. Every attending research site was given a group of Topic Detection and Tracking2 evaluation tasks including the segmentation, tracking, and detection.

2.1.2 Text Retrieval Conference and its History

Topic Detection and Tracking was devised under the DARPA TIDES program and the U.S government at the year 1998 and has emerged as the core of event oriented information retrieval framework. Another track named as NIST Text Retrieval Conference has focused on the efficient and effective knowledge retrieving relevant problem solving skills by providing consolidated datasets and test problems with over 25 years of evolving. Interestingly, Text Retrieval Conference has various tracks

such as the Clinical Decision Support Track, Contextual Suggestion Track, and Live QA Track and each of them has intrinsically different theme. These aforementioned tracks imply that a universally applied Text Retrieval Conference solver is not feasible at the current state of knowledge understanding. Text Retrieval Conference is involved in the following three areas: text summarisation, question answering, and quick machine translation [60]. Li and Roth [60] employ the question classification stream for text summarisation. This thesis utilises the same data-set for sentence classification and clustering. This thesis, convolutional recurrent neural network model and word mover's distance-based framework in particular, applies similar data pre-processing techniques and term weighting schemes for sentence interpretation and word vectors representation [11, 54].

At the annum of 2003, a record high of 93 groups of participants ranging from academia and independent institutions to industries attended the Text Retrieval Conference conference from over 20 different territorial countries. This shows the blossom of event detection and tracking research. The goal of the Text Retrieval Conference research is to provide infrastructure and necessary tool-kits for accurate or even precise manipulation and interpretation of multiple information sources, leading to the essence of informational knowledge.

Within the realm of event detection and tracking, a number of information retrieval models for novelty detection have been reported, according to Allan et al. [61]. For example, System for the Mechanical Analysis and Retrieval of Text is a well-known information retrieval system created by Cornell University in the 1960s [61]. Lucene system however, is an open source information retrieval library depended on Java delivering scalability performance by working coherently with Hadoop (Map-Reduce) in Amazon Book [62, 63]. Lucene library is also a generally feasible platform for searching and indexing with good granularity and wide coverage [63].

One distinguishable property of this work compared to existing Topic Detection and Tracking and Text Retrieval Conference is that it categorises sentences without traditional threshold learning using novelty score [11] and detection error trade-

off curve [64] for performance measuring. Plain information retrieval frameworks identify a first story with respect to novelty threshold[15].

2.2 Count Based Sentence Categorisation

Distributed word/sentence embeddings, feature terms weighting, and similarity calculation are three closely related concepts in count-based sentence modelling. Dense word embeddings and term weighting are stepping-stone for building vector space model framework. Vector space model requires similarity calculation for computing sentence pair dissimilarity. Count-based methods usually implement bag-of-words model to reflect statistical term frequency distribution in sentence categorisation.

2.2.1 English Word Segmentation

This thesis emphasises sentence categorisation in the context of English language and each involved benchmark data-set is written in English. One prominent characteristic of English is that a sentence is partitioned by natural space between each term. This makes word segmentation in English a trivial procedure. However, Chinese like word splitting is different in nature since Chinese phrases or words are gathered without space in between. According to Huang and Zhao [65], there are generally two approaches for Chinese word segmentation: dictionary look-up and statistical term frequency analysis.

2.2.2 Vector Space Modelling

Vector space model is the process of efficient storing and fetching terms frequency from a corpus [66, 67]. It captures statistical distribution of n-gram distribution. Traditional vector space model can be as simple as a $m \times n$ matrix where m denotes the number of sentences and n represents the vocabulary of the target corpus. Recently Reisinger and Mooney [68] propose a multiple prototype vector space model

where a structured vector container is constructed for homonymy and polysemy issues in natural language texts.

2.2.3 Dense Embeddings

In this section, several word embeddings and sentence level vectorisation in relation to this work are illustrated. For example, word2vec and global vectors for word representation are two popular pre-trained dense word representations [69, 70]. Word2vec was first introduced by Mikolov et al. [69] in their work on unsupervised skip-gram and CBOW neural network architecture. Specifically, word2vec is a simple and effective three-layer (input layer, hidden layer, and output layer) neural network to learn unsupervised word embeddings [69]. For skip-gram, one-hot vector representation of a single word is considered as the feed for the input layer and the output from the projection layer is its surrounding words. CBOW operates the opposite flow of the skip-gram framework. Input in CBOW algorithm is the context words and the output is the target word in one-hot vector.

Word2vec tries to learn optimal word embedding or distributed word vector through advanced optimisation mechanism such as hierarchical softmax or negative sampling [29]. Word2vec updates its learning parameters through stochastic gradient descent and backpropagation [71] algorithm. Word2vec is a fast and robust optimiser for which the computational time of training ten billion tokens on a single conventional computer is only half day, in accordance with Mikolov et al. [67]. Word2vec has proven to be a high quality word vector generator given a sufficient training corpora [71, 69, 24].

Under hierarchical softmax implementation, the three-layer neural model learns the optimal weights of the hidden layer by maximising the log-likelihood using a Huffman tree structure [72]. On the other hand, negative sampling method tries to minimising the log-likelihood of chosen negative samples. Often, frequent tokens are subsampled or deleted from the training texts in order to better capture the

surrounding contexts. The pre-trained word2vec corpus is trained on a large scale Google News archive with over ten billion tokens. The word2vec collection was consisted by three million word embeddings and each word vector occupies three hundred dimensions [23].

As an extension of word2vec, doc2vec or paragraph2vec was constructed to interpret the entire document [69] rather than a word. Rather than learning words and words relationship, paragraph2vec tries to associate labels and words. According to Le and Mikolov [71], paragraph2vec overcomes the weaknesses inherited from the traditional word2vec, mainly losing of the semantic ordering of words. Similar to the plain bag-of-words, doc2vec represents variable length documents with a fixed-length dense vector. In the word2vec architecture, the two learning algorithms are CBOW and skip-gram. For doc2vec, the corresponding mechanisms are distributed bag of words and distributed memory respectively. The commonly deployed gradient updating scheme in doc2vec bears resemblance to that of word2vec, i.e., stochastic gradient descent [71].

Global vectors for word representation was proposed by Pennington et al. [70] which incorporates corpus statistical information into the embeddings process. More specifically, global vectors for word representation effectively leveraged global matrix factorisation and local context window when being trained on nonzero elements in a word-word co-occurrence model. Pennington et al. [70] utilise latent semantic analysis algorithm [39] for global matrix factorisation and skip-gram or CBOW [67] for the local context window method. Global vectors for word representation embeddings includes three different pre-trained word embeddings in various dimensions: fifty, one hundred, and three hundred. For consistency, this thesis adopts three hundred dimensions on word2vec and global vectors for word representation. The global vectors for word representation word vectors used in the experiment was trained on Wikipedia 2014 and Gigaword 5 and the total number of tokens in the training corpus is six billion [70]. A noticeable contradiction between global vectors for word representation and word2vec is the size of the vocabulary. The pre-trained

global vectors for word representation corpora is 0.4 million and the number of word embeddings for word2vec is approximately seven times more, i.e. three million dense vectors.

Applications using word2vec and global vectors for word representation can be classified into four major categories: sentiment analysis or opinion mining [23, 73], text classification [74], word aware neural network construction [35, 75], and inspiring other social media platforms like Twitter for building specific tweet2vec dictionary [16].

Skip-thought vectors, aka sent2vec, is another type of distributed embeddings specifically for sentential snippet representations [42]. Fast text classifier, a recently devised architecture from Bojanowski et al. [76], can be applied to enrich word embeddings interpretation through subword information. It can also be used to efficiently classify texts or documents with bag of tricks [77].

2.2.4 Feature Weighting

In traditional information retrieval and machine learning oriented textual information classification problems, word weighting is a central issue to consider of. Term frequency-inverse document frequency is a feature weighting scheme where both the frequency of a target term in the document and its balanced inverse frequency are accounted for. Because of the IDF factor, a high frequency word or term is very unlikely to receive a heavy weight in the vector space model.

A standard term frequency-inverse document frequency includes the following integral units: term frequency (TF), document frequency (DF), inverse document frequency (IDF), and term weighting [11]. TF defines the frequency of a specified term t_j in the document D_i whereas DF denotes the value for the number of documents demonstrating the appearance of the target term t_j. IDF is a counter factor to restrict the high frequency word in the formulated vector space model. IDF is

expressed as

$$\log_{10}(\frac{d}{d_{f_i}}), \tag{2.1}$$

where d expresses the total quantity of documents in a corpus and d_{f_i} indicates the number of document contains the specified term t_j or the document frequency of the target term t_j. Term frequency-inverse document frequency has been prevalent since Topic Detection and Tracking-2002, and all four participants in the Topic Detection and Tracking-2002 evaluation competition used the term frequency-inverse document frequency component for the document processing and encoding [13].

Weighing a term is a critical ingredient for Topic Detection and Tracking based evaluation, the plain feature weighting methodology utilises the term frequency-inverse document frequency information to measure the importance of a specified token in the corpora. The traditional term weighting is summarised as follows:

$$w_{ij} = \frac{(\log_{10} tf_{ij} + 1.0) \times idf_j}{\sum_{j=1}^{t}[(\log_{10} tf_{ij} + 1.0) \times idf_j]^2}, \tag{2.2}$$

where w_{ij} defines the weight of term t_j in the document w_i, and idf_j denotes the inverse document frequency for term t_j.

Term frequency-inverse sentence frequency (TF-ISF) was first recommended by Li and Croft [78]. Unlike IDF, ISF score is computed based solely on sentence statistics. Compared to the term frequency-inverse document frequency scheme, TF-ISF was primarily designed for sentence level tasks. Recently Doko, Stula, and Stipanicev [79] proposed a recursive TF-ISF oriented sentence retrieval model with presence of the local context of sentences. The recursive TF-ISF incorporates local information to boost first story detection evaluation and sentence categorisation performance. Term frequency-inverse term frequency (TF-ITF) is another term based weighting mechanism appropriate for retrieving terms-phrases relationship [80]. Blake [80] conducted an experimental condition in which a three-dimension event space, namely term, sentence, and document, was constructed to compare the semantic interpretation difference of IDF, ISF, and ITF. The evaluation was experimented on a corpus of 100,830 full-text scientific papers. The experiment on

the document level language models reveals that raw IDF, ISF, and ITF are highly correlated [80]. Furthermore, IDF appears to be consistently robust on the entire corpus space as well as on a 10% sub-corpus. ISF and ITF based models result in slightly skewed vector space model statistics when tested on the same codition. This confirms that IDF is a robust schema across sentence categorisation tasks.

Okapi BestMatch25 (BM25) is a probabilistic relevance-based term weighting scheme which uses term frequency length normalisation [81]. Its normal usage includes ranking documents, web, newswire corpora, and n-gram language modelling [81]. It can also be used in collaboration with co-occurring query terms or proximity models [82]. BM25 ranking function can be adapted to deal with structured documents, according to Robertson, Zaragoza and Taylor [83]. Multiple weighted fields based experiment over Reuters volume 1 and the Text Retrieval Conference dotGov data reveals the superior performance of BM25 [83].

BM25F is a newer variant of BM25 which leverages document structure and anchor text [5] into the vector space model process. BM25F is often used in the off-the-shelf document retrieval system such as ranking linked data [84]. Semantic search engines employ BM25F for understanding documents structure and achieve better performance than the popular IR library Lucene [84]. BM25MF further extends BM25F by considering multi-valued attributes and entity normalisation [85, 4].

PL2F and PL2MF are another two field ranking formulas [85]. PL2F weighs entities' similarity according to the combination of three ingredients: the information gain, the Poisson randomness, and the normalised TF. PL2MF generalises PL2F by considering semi-structured data with multi-valued attributes. Evaluations on the Mean Average Precision (MAP) scores demonstrate better performance of PL2MF model over its predecessor PL2F [85].

Other term weighting schemes include the group average clustering (GAC) based hierarchical clustering algorithm, incremental clustering algorithm (also termed story-story algorithm or single pass clustering), story-cluster algorithm, and the LGT

scheme [13, 64, 86, 87]. The GAC oriented hierarchical clustering algorithm was specifically devised for retrospective first story detection. The incremental clustering algorithm works on both online and retrospective first story detection tasks. The LGT scheme combines the local element, global element, and topical features to effectively address polysemous and synonymous subjects on events [12].

Term frequency-inverse document frequency was adapted in this work simply because it is a robust and yet effective feature weighing schema which requires minimal time complexity during computation.

2.2.5 Minkowski Distance Family

Minkowski measures the distance between two points in a L_p norm [88]. The Minkowski distane between point x and y at space L_p is formed as:

$$d(x, y) = (\sum_{i=0}^{n-1} |x_i - y_i|^p)^{1/p}. \tag{2.3}$$

Manhattan distance and Euclidean distance are two special cases in the Minkowski distance family. Manhattan distance holds when $p = 1$ and $p = 2$ denotes Euclidean distance. When p reaches infinity or maximum, the distance is called the Chebyshev distance [89]. In other words, Minkowski distance is a generalisation of the plain Mahattan distance and Euclidean distance. Manhattan distance is defined in L_1 space and Euclidean distance is formulated at L_2 normed vector space. Minkowski distance is usually applied in clustering tasks such as symmetry analysis [90] and fuzzy clustering and its applications [91].

2.2.6 Similarity Calculation

Similarity calculation in the TREC and Topic Detection and Tracking tasks is usually interpreted as a spatial distance measurement [92, 93, 89]. Looking into the literature, several different methods have been proposed. A natural way of measuring distance between two probability distributions is to apply l_1-distance [94] or

total variance distance (i.e., half the l_1 distance). L_2 norm-based distance takes the square root of the l_1 one [95]. KL divergence with symmetrical smoothing has been used in many document to document distance scenarios [96, 97]. KL divergence has also applications in image similarity measure and images matching [96]. As KL divergence computation requires non-zero term frequency, symmetrical smoothing needs to be employed in the implementation. Often add-one smoothing is applied in the calculation [96].

Later, asymmetrical smoothing technique is developed to leverage possible impacts caused by seen terms and unseen words [97]. A larger smoothing factor is applied on the seen words and a small smoothing constant is set for unseen tokens.

Cosine distance can be an alternative for similarity calculation [11]. In cosine distance modality, query Q is compared to each individual host document of seen events. Suppose the document collection contains a list of n documents and the vector space model is a matrix of $t \times t$ terms, the difference between pivot document D_i and query Q is depicted as

$$\begin{aligned} \text{cosine}(Q, D_i) &= \frac{\sum_{j=1}^{t} w_{q_j} d_{ij}}{\sqrt{\sum_{j=1}^{t} (d_{i_j})^2 \sum_{j=1}^{t} (w_{q_j})^2}} \\ \text{label}(Q) &= \begin{cases} \text{unseen}, & \text{if}\, \theta < \epsilon \\ \text{seen}, & \text{otherwise} \end{cases} \\ \theta &= \min(\text{cosine}(Q, D_i),\ \forall i \in |D|)), \end{aligned} \tag{2.4}$$

where w_{q_j} represents the weight of term q_j and d_{i_j} denotes the weight of term d_j in D_i. In the denominator, the modulus of document D_i and query Q is computed. Practitioners can next identify the smallest spatial distance value accordingly by sorting the intermediate cosine similarity results to get the minimum θ value. By comparing θ with the pre-defined threshold ϵ, the cosine distance metrics is cast as a binary 'classification' on seen event or unseen event. This type of binary classification is a special case of novelty scoring on first story detection.

For cosine similarity, previous studies have reported that two major variations

are worth considering: max cosine similarity or 1-NN approach and mean cosine similarity [11]. For max cosine similarity measure, the maximum value is selected from a group of document to document comparisons. However, for mean cosine similarity measurement, the mean value from each document to document comparison is output.

Earth mover's distance generally measures spatial distance between two images in the context of computer graphics or image classification [98]. In mathematics, earth mover's distance aka the Wasserstein or Vasershtein metric is a distance function defined between probability distributions on a given metric space M [26]. Intuitively, if each distribution is viewed as a unit amount of 'dirt' piled on M, the metric is the minimum cost of turning one pile into the other, which is assumed to be the amount of dirt that needs to be moved multiplies the distance it has to be travelled. Due to this analogy, the metric is known in computer science as earth mover's distance [28]. It intuitively captures the minimum cost required to transform one distribution into another and is robust against outliers and slight probability shifting.

Computing Earth mover's distance was inspired from a solution to the well-known transportation problem [26, 98]. For instance, a group of suppliers needs to carry goods for several consumers, with each supplier having a constraint on capacity and the cost of transporting a single unit for each pair is pre-defined. The transportation problem is to seek an optimal goods flow which minimises the overall transportation cost and fulfil consumers' requirement. The Measuring Earth mover's distance in computer vision is typically solving a linear programming problem. Suppose $P = (p_1, w_{p_1}), \cdots, (p_m, w_{p_m})$ is the first distribution with m clusters. Each pair in P is composed by the cluster pivot p_i and the cluster weight w_{p_i}. Similarly, the second signature Q can be denoted as $Q = (q_1, w_{q_1}), \cdots, (q_n, w_{q_n})$ with n clusters. The ground distance between cluster p_i and q_j is defined by $D = [d_{ij}]$. Users seek to find a flow $F = [f_{ij}]$ that minimises the overall cost. This process can be formulated

as

$$\text{WORK}(P, Q, F) = \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} d_{ij}, \tag{2.5}$$

subject to the subsequent constraints:

$$\begin{aligned} f_{ij} &\geq 0 \quad 0 \leq i \leq m, 0 \leq j \leq n \\ \sum_{j=1}^{n} f_{ij} &\leq w_{p_i} \quad 0 \leq i \leq m \\ \sum_{i=1}^{m} f_{ij} &\leq w_{q_j} \quad 0 \leq j \leq n \\ \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} &= \min(\sum_{i=1}^{n=m} w_{p_i}, \sum_{j=1}^{n} w_{q_j}). \end{aligned}$$

The first restriction specifies that the goods can only move from P to Q and not in opposite direction. The second and third constraints restrict the quantity of suppliers which can be sent from P to Q and the amount of goods can be received by Q from P respectively. The fourth restriction forces the maximum possible number of suppliers being moved during transportation. Earth mover's distance can then be defined as the normalised work:

$$\text{EMD}(P, Q) = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} d_{ij}}{\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij}}. \tag{2.6}$$

The use of the earth mover's distance has been pioneered in the computer vision literature, according to Rubner et al. [26] and Ren et al. [28]. Several publications reported approximations of the earth mover's distance for image retrieval applications and computer vision. In text classification area, pre-trained word embeddings like word2vec [69] and global vectors for word representation [70] associates each word with a highly informative feature vector to form a basis for spatial distance measurement. Formally, under Kusner et al. [23] implementation, word mover's distance can be formalised to solve a linear optimisation problem by minimising:

$$\sum_{i,j=1}^{n} G_{ij} w(i, j) \tag{2.7}$$

subject to the following two constraints:

$$\sum_{i=1}^{j} G_{ij} = d_i \quad \forall i \in \{1, ..., n\}$$
$$\sum_{j=1}^{j} G_{ij} = d'_j \quad \forall j \in \{1, ..., n\} \tag{2.8}$$

with $w(i, j)$ being the word2vec embedding distance between word i and j which is usually a constant given two fixed terms and $G_{ij} \geq 0$ being the proportion of term i in the document d moves to term j in document d'. In other words, the optimum travelling distance between two given documents d and d' is defined as a transportation problem where the discrepancy between the normalised bag-of-words distribution of d and d' are to be computed and named as word mover's distance. Because each word carries a word2vec or global vectors for word representation word embeddings, word travel cost between any two tokens can be derived through a distance function such as Manhattan, cosine, or Euclidean in the corresponding vector space model. Plain earth mover's distance takes quadratic time complexity which may lead to high computational overhead [23]. A relaxed version RWMD replaces the normal word mover's distance with reduced hard constraint on the regular linear programming problem, following with Kusner et al. [23].

Latent dirichlet allocation was first described by Blei et al. around a decade ago [38]. His work considered the problem of modelling text corpora and other collections of discrete data. Latent dirichlet allocation is a generative probabilistic model for collections of discrete data such as text corpora. Latent dirichlet allocation is a three-level hierarchical Bayesian model, in which each item of a collection is modelled as a finite mixture over an underlying set of topics [55]. Each topic is, in turn, modeled as an infinite mixture over an underlying set of topic probabilities. In other words, each document d is formulated as a distribution over K topics, which are themselves characterised by distributions over words. The words in a document are generated by repeatedly sampling a topic according to the topic distribution and

then sampling a single word from the chosen topic. In the context of text modelling, the topic probabilities provide an explicit representation of a document. Latent dirichlet allocation has applications to text classification, document modelling, first story detection, and collaborative filtering.

An extended variation of latent dirichlet allocation is the Author-Topic Model (ATM) [99, 100] which relates author information to a mixture of topics for modelling and organising the topic distribution. Topic trends detection is another variant of latent dirichlet allocation which embeds temporal dynamics into the model [55].

The Hellinger distance is a classical sentence similarity calculation method firstly introduced by Hellinger at 1909 [101]. It was often used to quantify the dissimilarity between two probability distributions [101]. The Hellinger distance is expressed in terms of the Hellinger integral. Other than the traditional usages in sentence categorisation and first story detection, the Hellinger distance has wide application in other fields. For example, it can be applied in nearest neighbour classifier for relational database applications [102]. Entropy-based Exemplar Learner (EEL) was developed by Lee and Shin [102] to identify the nearest neighbours within relational datasets. EEL contains the following two phases: phase I: calculate attribute weights, and phase II: select k-similar instances.

2.3 Neural Networks and Probabilistic Language Models

Under this section, various deep/shallow neural network structures and probabilistic language modellings relating to this work are presented. Neural networks are relevant to sentence categorisation because they can be transformed to classify target label of a textual snippet [103, 104, 105, 106]. Language modellings have roots in sentence categorisation since estimating a model based on a set of sentences is analogous to feature vectorisation in sentence categorisation [107, 29, 9, 108, 109].

2.3.1 Deep Neural Networks

An artificial neural network is generally constructed by three elements [110]:

1. Architecture (layering design and neurons interconnection and message passing and conditional exposure)

2. The optimisation (backpropagation) algorithm that updates the weights of the interconnections

3. The activation function that mimic the biological neural networks' translating an input signal to its output information.

Until recently, scholars are aware that convolutional deep neural network (CDNN) can be applied to many classification tasks. The prior applications of convolutional deep neural network include traffic sign classification [111], robust multi-speaker speech recognition[112], visual deep reinforcement learning from crude 3D buffer information and experience replay [113], neural machine translation [114], and ImageNet classification [33]. Those architectures are not dealing with sentence categorisation.

The emergence of convolutional deep neural network has also shed some lights on the text mining society [75]. Kim [74] developed a two-channel CNN, a small variation of Collobert et al. model [34]. Kim's model was specifically used for sentiment polarity scoring and questions classification. Zhang et al. [17] offered an empirical investigation on the usefulness of character-level convolutional networks for binary textual classification. Further, Yin and Schutze [115] exhibited a multi-channel CNN for sentence classification. Zhang et al. [35] gave a multiple word embeddings based CNN model for sentence categorisation. Nevertheless, those frameworks were tested merely on simple classification or sentiment analysis.

Traditionally, pure recurrent neural networks were applied to fields like statistical machine translation [21], polyphonous music modelling, speech signal understanding [116], and Python programme evaluation [117]. Currently, recurrent neural networks

have seen promising results on text classification. Recurrent neural network is usually powered by the recurrent units such as *tanh* [116]. Advanced recurrent units such as gated recurrent unit cells [21], long short-term memory [22], and minimal gated unit [118] are released recently. Minimal gated unit is probably better than gated recurrent unit in that the gating mechanism in minimal gated unit not only halves the number of units when building up the model, but also reduces the size of the model by optimising the number of parameters during network interaction [116, 119]. Moreover, gated recurrent unit cell is different from the long short-term memory in that a separate memory cells is not presented in the former one. However, there is no clue which of them has the best performance without further investigation [116].

Aforementioned review demonstrates work on recurrent neural networks or convolutional neural networks. Ensembles of models of convolutional neural network and recurrent neural network is presented now. Lai et al. [120] proposed a recurrent convolutional neural network (RCNN). The recurrent structure learns the specific word with its left and right context or simply words representation in Lai et al. [120]. Their model did not take profits from the pre-trained word2vec [69] or GloVe [70]. Rather, sentence structure is obtained from its recurrent state information. Kim et al. [10] described a hybrid network with convolutional neural network-long short-term memory to capture syntactic information buried in the contextual data. Moreover, empirical survey in Zhang et al. [17] shows that deep convolutional neural network is less effective when the data volume is less than several million. Recently, Xiao and Cho [121] proposed a convolution-recurrent network for character-level document categorisation. Their model effectively utilised a hybrid of convolutional and recurrent networks which took advantages from both architectures. However, the convolutional recurrent neural network framework is different from [121] in that the proposed model augmented an aggregation layer before the logistic classifier to incorporate recurrent neural network encoder with convolutional neural network activations.

2.3.2 Objective Function

An objective/cost function is designed to quantify the quality a neural network can achieve with respect to its desired output. It often produces a scalar to show the error of a given neural network's performance.

Neural network needs to update its parameters iteratively when training, which requires parameters estimation on the difference between estimated and true values for each epoch of operation [122]. Loss/objective function optimisation is to quantify the training quality [123] after certain cycles of training.

A classical example of cost function is the mean squared error. MSE is aka. maximum likelihood or sum squared error [124]. Cross-entropy cost function or Bernoulli negative log-likelihood is another well-known cost function mostly applied to recent backpropagation enabled nerual networks [125]. Compared to MSE, cross-entropy has a more complex gradient computing formula which consumes more computational resources [125].

When combined with a backprogation algorithm, minimising a loss function via stochastic gradient descent [126, 127], AdaGrad [128], RMSProp [129], or adaptive moment estimation are suitable for training neural network. Other stochastic loss function optimisation schemas include vSGD [130], the natural Newton method [131], and AdaDelta [132].

Other popular objective functions subsume Hellinger distance [133], Kullback-Leibler divergence [134] and its generalised form [135], and Itakura-Saito distance [136]. Those distance-based cost functions need to have positive values, and ideally values between 0 and 1 [137].

2.3.3 Neural Network Optimiser

Optimiser aims to learn and tune the possibly optimum weight [138]. Objective/cost function is designed to quantify the quality a neural network can achieve and an optimiser can boost the journey of minimising or maximising the cost function [139].

An optimiser for neural networks can be as simple as the backpropagation schema [139]. Backpropagation is an abbreviation for backward propagation of errors. It often comes with stochastic gradient descent to collaboratively enhance a neural network's performance [139].

Backpropagation is a kind of mechanism for fast computing the gradient [140]. Each propagation involves the following five steps:

1. set the initial value, activation, and learning rate for the input layer;
2. forward propagate on a training network's input for generating the network's output values;
3. output error terms based on feedforward operation;
4. backward propagate the error to get the partial derivatives on the specific bias terms;
5. compute the gradient of the cost function.

For updating network's weight, the output error and old weight are taken into consideration. Neural network is trained iteratively until the mapping from input layer to output layer is sufficiently good for the given problem.

Gradient descent optimisation algorithms have seen increasing prevalence in the recent decades [141]. There are three direct variations of gradient descent, namely batch gradient descent or vanilla gradient descent, stochastic gradient descent, and mini-batch gradient descent [138]. The major difference among them is how much data is exposed to compute the slope of the given objective function. Practitioners have developed the following gradient descent-based optimisers so far: stochastic gradient descent (without or with momentum in two versions), AdaDelta [132], AdaGrad [128], Adam [141], Adammax [141], Nadam [142, 143], and RMSProp [144]. Technically speaking, Nadam is Adam RMSProp with Nesterov momentum [142, 143]. Gradient descent algorithms are known for seeking local optimum. Other

than the aforementioned gradient descent optimisers, genetic algorithm (GA) [145], simulated annealing [146], and adaptive tabu search [147, 148] have also gain reasonable attention in the domain of neural network optimisation.

2.3.4 Hyper-parameter Optimisation

Hyper-parameters in artificial neural network usually include learning rate, learning rate decay function, and dropout probability [149]. When seeking an array of optimum hyper-parameters, two possible options are available: grid search aka exhaustive search and random search. Bergstra and Bengio have empirically and theoretically show that a random strategy performs better than a grid search algorithm [149]. Grid search algorithm works better in non-neural network frameworks like K nearest neighbours [23, 24].

2.3.5 Activation Function

An activation function of a neural network serves as a gate to modulate the input-output interconnection [150, 75]. There are eight commonly used activation functions in practice. They are listed and described as follows:

- Identity function is probably the most basic form in the activation family. In other words, it denotes no activation function is applied during the operation.
- Sigmoid seems the simplest non-linear activation function. It squashes real numbers to range between 0 and 1. However, it saturates and kills gradients and the output of it is not zero-centered. If the input data is always positive, then the gradient on the weights is likely in a zig-zagging shape.
- Tanh is another well-known rectifier. Tanh stands for hyperbolic tangent. The output range is -1 to 1. Actually the tanh rectifier is a scaled sigmoid neuron.
- Softplus takes the primitive of sigmoid's first derivative to form a convex activation function [151].

- Rectified Linear Unit is a popular activator since it involves simple computation of activation and boosts the convergence of stochastic gradient descent [152, 33]. Activation in the Rectified Linear Unit is simply thresholded at zero.

- Leaky Rectified Linear Unit aims to alleviate the issue of hard zero activation of plain Rectified Linear Unit neuron by introducing a small negative slope when the input is smaller than zero [153].

- PReLU, aka parametric Rectified Linear Unit pushes leaky Rectified Linear Unit one step further through a learned coefficient for controlling the slope of the negative part [154]. PReLU has no extra overfitting problem because the number of additional parameters is equal to the total number of channels or weights.

- Maxout [155] generalises Rectified Linear Unit and its leaky variant. Maxout performs on a set of inputs and outputs the maximum value. It is optimised for dropout and doubles the number of training parameters for each activation step, resulting in a relatively slow updating mechanism.

In the above list of candidate activators, tanh and Rectified Linear Unit are reported with general robust performance across several application areas like text classification, machine learning, and image processing [75].

2.4 A History on Language Modelling

This section describes the history and current development of language modelling such as cross-language model and translation modelling approach. Generative relevance model and Naive Bayesian probabilistic classifier are also presented inclusive in the language modelling field.

The very first statistical language model was developed by Claude Shannon, in line with Berger and Lafferty [156]. In his application of the newly founded theory for

human language based information exploration, Shannon took language into consideration as a statistical source, examining the performance of the generative n-gram models in projecting or, equivalently, compressing natural text. More specifically, he estimated the entropy of English words through experiments with human subjects. The cross-entropy of the n-gram models on natural texts was also evaluated.

Since the underlying human generated language is so diverse and evolves so quickly, it demands much efforts on estimating the true entropy of language. Although this kind of language model is experimented or tested quantitatively, the predictive power of simple n-gram language models is much lower than that of plain vector space models [93].

On the basis of [157], language modelling approach has been advanced to cross-language retrieval about a decade ago. Because basic language models do not address issues of synonymy, a cross-language information retrieval model generates query words not in document via 'translation' to synonyms. This improved modelling approach is capable of translating a document into a query under a bilingual environment. More formally, through the attached translation probabilities to pairs of words with a parallel corpus or a pseudo-parallel semantics dictionary, the problem of out-of-vocabulary can be corrected to a certain extent.

According to Allan [56], a translation model can also accommodate with the problems of synonymy. Synonymy is done by learning a translation model through a dictionary or statistical machine translation. Language models provide a novel way of looking at the problem of text retrieval and text categorisation based on probabilistic language modelling. It is conceptually simple and explanatory. Since language models use collection statistics naturally, hence there is almost no heuristics or randomness within the experiments. As a consequence, the results generally require no replications in statistical analytics. Next, language models support high efficacy in information source retrieval and can be improved so that the following conditions can be met: a) language models are accurate representations of the data. b) users have some sense of term distribution.

Probabilistic semantics is examined in the relevance models for Topic Detection and Tracking [156]. The basic question for the relevance model is 'What is the probability that this document is relevant to this query?'. In a classical treatment of the probabilistic programme, a binary random variable R is introduced to denote relevance of the document D and the query Q. Notice R only has two possible options: relevant, represented by r and not relevant, denoted as $\bar{r}$. In probabilistic perspective, the basic question is then equivalent to evaluating the probability of relevance $p(R = r|D, Q)$. Consequently, this probability is treated as the basis for ranking documents, according to the Probability Ranking Principle by Lavrenko and Croft [157].

The probability of relevance is not computed directly from the generative relevance model. Rather, it is evaluated implicitly via applying the Bayes' rule [158]. The Bayes' rule can also be utilised to infer that the language modelling approach is within the same probabilistic framework, which is similar to the traditional probabilistic modelling method [81].

Naive Bayesian is applicable to first story detection, as noted by Manning et al. [159]. Naive Bayesian model can also be applied to filter spam in Emails and anti-spam programmes [160]. Parameters estimation in a Naive Bayesian classifier can either be Gaussian naive Bayes, multinomial naive Bayes, Bernoulli naive Bayes, or semi-supervised estimation [161, 162, 163]. Gaussian naive Bayes assumes the Gaussian likelihood of the features. Multinomial naive Bayes is mostly suitable for multinomially distributed data and has been widely used in text classification [164]. Bernoulli naive Bayes is applied to samples with binary-valued feature vectors and the data is distributed based on multivariate Bernoulli distributions [162]. In the case of text classification, Bernoulli naive Bayes requires word occurrence vectors rather than bag-of-words matrix [163]. Semi-supervised estimation [165] is an instance of the expectation-maximisation algorithm (EM). Expectation-maximisation algorithm can be deployed to perform semi-supervised text classification and sentence categorisation tasks [166]. Semi-supervised estimation works on data gener-

ated by a mixture model. Semi-supervised estimation runs a supervised learning algorithm on the labelled and unlabelled data in a recursive loop until convergence [134]. Recently, a semi-supervised multinomial naive Bayes was devised for large scale text classification [164].

A multinomial Naive Bayesian believe network tends to represent sequence of terms as it occurs in the document [167]. The fundamental question of it is 'what is the identity of the i'th query token?' and the observation is a sequence of events, one for each token. Bernoulli Naive Bayesian in the scenario of sequence learning attempts to produce a binary vector of dimensionality that indicates for each term whether it occurs in the document or not. The fundamental event of this original model is that 'does the word w occur in the query?' and the observation is a vector of binary events, one for each possible word.

2.4.1 Language Modelling

A statistical language model, or simply language model, is a probabilistic mechanism for modelling text [107]. Discriminative approaches model decision boundary [168]. For example, does this document belong to class X or Y is usually an issue addressed by a discriminative model. Language modelling approach is generative and it can be adopted to generate text. The goal of a language model is to estimate a model M from a sample text S [9].

The ongoing literature suggests that there are two types of widely used language models: unigram or higher-order models and multinomial or multiple-Bernoulli [9]. Unigram model does not consider the order of the words or assumes word independence. For example, 'brown dog' and 'dog brown' have same coding scheme in the unigram model. This indicates that unigram model tends to capture superficial form [20]. Higher-order models such as n-gram model with condition on preceding words, cache-like model with condition on a window, and grammar model with condition on parse tree resolve the problem of word independence, and are very likely to signal

prohibitively expensive parameter estimation in practice [20].

2.4.2 Scoring a Language Model

Documents ranking usually deals with issues like 'what is the probability to generate the given query, given a language model?' or 'what is the probability to generate the given document, given a language model?'. Scoring in language models can be evaluated with four different approaches, namely query-likelihood scoring, document-likelihood, likelihood ratio, and divergence of query and document models or model comparison scoring [159].

The standard query-likelihood approach evaluates a language model MD for every document D in the collection and ranks documents by the probability of 'generating' the query. The likely drawbacks of the standard query-likelihood approach are fourfold. First, there is no relevance in the model, i.e., everything is random sampling. Second, user feedback or query expansion is not part of the model. Third, examples of relevant documents cannot facilitate in improving the language model itself. Fourth, query-likelihood modelling approach does not directly allow weighted or structured queries [108].

Document-likelihood approach is the flip side of the query-likelihood method. More specifically, it constructs a language model MQ for the query Q and ranks documents D by the likelihood of being a random sample from MQ. MQ is expected to 'predict' a typical relevant document. A problem of document-likelihood approach is that the probabilities of different documents are not comparable because of distinct document lengths [158].

Likelihood ratio scoring method tries to fix the problem existed in the document-likelihood by allowing relevance feedback and query expansion combining with Probability Ranking Principle [158]. As a result, likelihood ratio can benefit from complex estimation of the query model. In terms of the model comparison score, it combines virtues of query-likelihood and document-likelihood via a naturally mea-

suring cross-entropy on the similarity score of the two models. Model comparison scoring computes a model of both the query MQ and the document MD with a direct resemblance comparison of the two models, so to speak. Divergence of query and document models is equivalent to Kullback-Leiblar (KL) divergence with respect to probabilities score [108].

2.4.3 Estimating Background Probability of a Word

For probability estimation in language model, maximum-likelihood estimator is often applied, according to Nallapati [169]. It counts the number of times each word occurs in the sample text S, divided by its length for normalisation purpose. Smoothing techniques are frequently deployed to avoid zero frequencies. A maximum-likelihood model assigns zero probability to those events which are not appeared in the initial observation. This happens very often when creating a model from short samples.

As stated in Parapar et al. [6], discounting methods such as Lidstone correction [170] and absolute discounting [107] enhance better interpolation with background probability of a word. Lidstone correction works by adding a constant to all counts and accomplished by re-normalisation [170]. Absolute discounting takes the approach of subtracting a constant from each nonzero n-gram count and redistributing the probability mass through an interpolation function [168]. Laplace smoothing attempts to count events in observed data by adding 1 to every count, followed by re-normalisation to obtain probabilities with correspondence to uniform priors [171].

In accordance with Nallapati and Allan [108], a vital problem with all discounting or smoothing methods is that they unconsciously treat unseen words equally by either adding or subtracting to all the counts, which may exert a negative impact on the original word frequency table. Using background probabilities to reflect expected frequency of events better accommodates with demanding estimation tasks [108]. Interpolation methods naturally inherits from the concept of background

probabilities through interpolating estimates with General English expectations. The overall estimate is adjusted as relative frequency for all events, both seen and unseen, in a large collection, so to speak.

For example, Jelinek-Mercer smoothing can provide reasonably good results for longer length documents, in accordance with Manning et al. [159]. Whilst dirichlet smoothing works comparably well with shorter length data stream via introducing two internal variables N as the length of sample or document length and μ as a constant to alleviate issues of different document length in a given corpora. Witten-Bell smoothing takes a step further by taking 'redundancy' of the example into smoothing [172]. Long and redundant example requires little smoothing, whereas short and sparse example desires a lot of smoothing. Two parameters N for the total number of events and V as the number of unique events are defined in the Witten-Bell smoothing technique, grounded on Federico, Bertoldi, and Cettolo [172]. Those two parameters assist in deriving estimate by considering the proportion of new events when seeking through the sample. Two-stage smoothing with explaining unseen words via Dirichlet prior (Bayesian) at stage 1 and eliminating noise in query through 2-component mixture at stage 2 has seen its place in the relevant surveys, according to Zhai and Lafferty [173].

2.5 Applications of Vector Space Model

According to Muhr, Zechner, Kern, and Granitzer [66], external plagiarism detection methodology shares some similarity to textual information retrieval. For instance, given a collection of query terms an information retrieval system yields a ranked set of documents from a data stream which matches best to the queries. More specifically, the inverted index in a vector space model is the most common framework for addressing such query terms. External plagiarism detection makes use of a reference inventory composed of documents where a block of sentences or a fixed amount of paragraphs are plagiarism-suspicious. A document is classified to be questionable

either because it duplicates with the reference corpus or near duplicates with the reference. An external plagiarism system then uploads these initial results to a human assessor who holds the final viewpoint whether the detected plagiarism are true or false. External plagiarism detection problem can also be defined as nearest neighbour finding [66].

Another approach for plagiarism checking is hashing scheme or fingerprinting, according to Muhr et.al. [66]. This type of methods yields one or more hashing values that identify the content of a document or sentence. A suspicious finding is then matched with a reference corpus based on their fingerprints. Duplications or near duplicates have relatively similar hashes, thus they share several common hashing factors. Since bag-of-words-based vector space model often introduces high dimensional workspace, the space partitioning techniques which relies on the triangle inequality is employed to perform dimensionality reduction.

Intrinsic plagiarism detection was first appeared in 2006, in line with Muhr et.al. [66]. It attempts to signal suspicious documents without the help of a reference corpus. For concreteness, a questionable document is first broke down into passages. A feature vector incorporated with a corresponding difference vector is built up to make a decision of which passages are plagiarised. Ground truth is a prerequisite to train the model with binary classification, i.e., either suspicious or not. A possible problem of the intrinsic plagiarism detection is that the model is likely to suffer from no prior knowledge on the ground truth derivation.

Some work has been carried out by Song et al. [174] to improve the efficiency of novelty detection systems by introducing a news indexing-tree. Luo, Tang, and Yu [175] design a framework for online new event detection used in the wild. The approach used is called 1-NN approach and the framework focuses on improving system efficiency by reducing the number of saved documents using indices, parallel processing, and etc. Comparing to prior research, the proposed method increases the efficiency of novelty detection by eliminating exhaustive comparison promoted in the one nearest neighbour approach and this is described in the methodology

section.

Previous studies have been primarily concerned with how a vocabulary can be individually weighted, and hence first story detection systems generally implement static term weighting scheme. This can be seen in the case of the two well-known Topic Detection and Tracking systems, namely the UMass and the CMU system [11, 53]. To the best of our knowledge, only few investigations have been done on appropriate yet efficient weighting of a token. One of these is the work by Brants et al. [13], the first story detection system is a typical instance of incremental term frequency-inverse document frequency weighting framework, the work reported an incremental performance on the standard TDT3 and TDT4 datasets.

Language models share some commonalities with the statistical vector space model. First, both of them weigh terms based on frequency. Second, terms often used as if they are independent, uni-gram language models for instance. Third, inverse sentence/document frequency is presented in the model construction. Fourth, some form of length normalisation is applied to cancel out issues of biased collection. Nevertheless, language models use the intuitions that probability measure is embedded, whereas vector space model is based on similarity calculation. Details of use of document length and term, document, corpora frequency differ as well. Smoothing strategy in language models has a role similar to IDF in vector space model.

Transfer learning from knowledge bases to topics extraction is another domain for first story detection tasks [176]. Empirical evaluations have been conducted on structure-guided category-level topics extraction to enriching semantics information. Enriched semantics enhances detecting emergence of events in the microblog stream, according to Huang et al. [176]. Word co-occurrences and transfer learning exert a positive influence on reducing noise when extracting topics in a pre-trained knowledge base.

2.6 Content-aware Recommendation Engine

Recommender platform has evolved in the interactive environment of the world wide web (WWW). Recommender system applies Knowledge Discovery in Databases (KDD) mechanism to the domain of making product recommendations under E-commerce environment [177] or scientific areas [178]. The core of a recommender engine is document similarity and top documents ranking [179, 178], which have same root in sentence categorisation work. Commercial applications are Amazon book suggestions engine [179], CDnow for CDs recommendation [179], YouTube video recommendation system [180], and Netflix recommendation system for movies, TV programmes, and video clips [177].

As the world's most popular video platform for sharing originally-created videos, YouTube has offered some unique chances and difficulties for video clips sharing and recommendations [180]. At present, the recommender engine for YouTube is based on a top-N list of videos [8]. In the case of YouTube ecosystem, a mapping from a seed video to a set of similar or relevant videos is constructed. This mapping process is analogous to pairwise documents dissimilarity comparison in sentence categorisation. Video meta-data includes title, description, and upload location is utilised to rank the top-N recommended videos, which conforms to the standard 2D (User $\times$ Item) space in collaborative filtering [181].

Recent development on the multidimensional space-based collaborative filtering encourages using effective dimension reduction techniques [181]. Contextual dimensions like time, place or location, and company information are available options for constructing a comprehensive recommendation platform. Methodologies discussed in Sect 2.2 are applicable in this scenario.

Scientific recommender platform facilitates client users identify most related papers out of sheer amounts of literature tank. Recently, Feyer et al. present an integration of the scientific recommender system Mr. DLib into the reference manager JabRef for intelligent bibliography management [178].

To date, there have been several new techniques devised for context-aware recommender systems. Oku et al. [182] show a SVM-based context-aware algorithm for generating high quality recommendations. Their method creates a boundary on like side and dislike hyperplane. The given item is recommended if it resides at the like side of the plane. Yu et al. [183] develop a joint model combined with Bayesian network, rule-based technique, and synthesising method. Hariri et al. [184] find that latent dirichlet allocation fits well in the recommendation systems. Personalised access model (PAM) has strong application on context-aware recommender engine as well because of its personalised or customised context service [185]. Recently, Youngki et al. offer a variant of traditional collaborative filtering method by reversing the process of finding k rated neighbours [186]. It is a rapid algorithm in that finding rated items requires less predictions compared to unrated ones. Youngki et al. also embeds a fast k nearest neighbour graph construction algorithm and term frequency-inverse document frequency to boost the recommendation process [186].

2.7 Contribution to Review of Literature

This chapter contributes to the field of sentence categorisation by the following points:

- A review of various Topic Detection and Tracking methods and their application to sentence categorisation work;
- A novel perspective on count based sentence categorisation model breakdown according to vector space model, distributed word embeddings, feature weighting, distance metrics, and similarity computation;
- An examination on recent neural network mechanism such as convolutional neural network, recurrent neural network, and their variants for constructing sentence categorisation framework;

- This chapter introduces a novel way on breaking-down neural network-based sentence categorisation framework according to cost function, optimiser, hyper-parameter optimisation, and activation function;
- The chapter on literature review lists popular activation functions, neural network optimisers, and cost functions respectively for deep neural network construction;

2.8 Chapter Discussion

Topic Detection and Tracking has seen prevalence since 1980s [54]. Many natural language processing related paradigms and evaluation techniques have originated from the Topic Detection and Tracking community. For instance, event detection related tasks and precise sentence representation are derived from first story detection [187, 53]. Plain first story detection based frameworks are normally unsupervised learning tasks which requires human annotation and exhaustive threshold learning [14, 11]. However, the word mover's distance-based sentence categorisation frameworks proposed in this thesis are free from manually labelling and threshold learning.

Classical spatial distance metrics or sentence-to-sentence similarity computation relies on cosine, Euclidean, Manhattan, or earth mover's distance distortion function [92, 27]. This work proposes a novel word mover's distance-based distance metrics which works jointly with advanced statistical term frequency approaches to effectively categorise testing sentences to a specific label or class. Those proposed models are then compared to strong baseline algorithms to verify their accurate categorisation performance on the unsupervised and supervised English language data-sets. We belief that this present research is the first one to present hierarchical agglomerative clustering-word mover's distance as an unsupervised sentence categorisation model.

For neural network-oriented frameworks, an observation can be yielded that

many previous work focus on singleton neural network architecture and word-level input sequences encoding. The proposed model offered in current work is the hybrid of a convolutional network and recurrent structure with character level encoding. The joint framework fits well with the four benchmarking data corpses and yields comparable performance.

The character-level convolutional neural network presented in this thesis was tailored from the original model to make it suitable for the present work [36]. The character-level convolutional neural network developed in the present research consists of a convolving layer, Rectified Linear Unit, and max-pooling layer. The classical convolutional neural network component from Vosoughi et al. [16] has multi-layer structure which seems inefficient when trained with the benchmarking data. The current thesis adapted character-aware recurrent neural network ingredient from Cho et al. [21] and Hochreiter and Schmidhuber [22]. The character-aware recurrent neural network provided in the present work unit has three different variations including gated recurrent unit [21], minimal gated unit [118], and long short-term memory [22]. The character-aware recurrent neural network is different from the tradition in that the input gate fits with the intermediate result from character-level convolutional neural network and the output gate follows a modulated activation [36].

Chapter 3

Methodology

This chapter presents various methods in relation to efficient sentence categorisation and performance evaluation measurement. For example, section 3.1 refers to a list of seven vector space model dimensionality reduction approaches. Five sentence distance functions and baseline algorithms for sentence classification are presented in this section as well. Sentence categorisation evaluation measurements are also offered in this chapter.

3.1 Dimensionality Reduction Techniques

This section covers various relaxed document representation methods. Often those kinds of mechanisms are employed to cope with high dimensionality data compression and fine bag-of-words representation. Due to the recent advance on dense word embeddings, bag-of-words representation requires even larger matrix storage in order to capture linguistic and semantic meaning of textual snippets. As a guidance to sentence dimensionality, this work presents a list of seven candidates. Empirical evaluation on those candidates indicates that reduced vector space model space complexity normally exerts a negative influence on sentence categorisation with a side effect of high computational overhead. The present thesis attributes this to the following two reasons. First, the benchmark data-set is a sparse collection with

regard to term features and reduced feature sets cannot fully represent the overall distribution. However, locality sensitive hashing works in a different way that each original feature is preserved in the model rather than a reduced set. Second, other than the runtime efficient locality sensitive hashing and k-term hashing model, the other participated techniques tend to reserve large computational resources and time during calculation. To this end, this thesis includes only the constant timing locality sensitive hashing in the context of supervised learning tasks.

Vector quantisation is an approximation approach to generate a subset of elements to represent the entire vector space [50]. Non-negative Matrix Factorisation has been widely applied as a useful decomposition for multivariate data analytics [51]. Fisher linear discriminant, aka linear discriminant analysis, is another strong candidate in data dimensionality reduction and is prevalent in pattern recognition [188]. PCA is an efficient data pre-processing technique. Latent semantic analysis or latent semantics indexing implements a singular value decomposition like algorithm to express the correlation between terms and sentences which explores implicit relationship like synonymy and polysemy [39]. Locality sensitive hashing is a fast approximate nearest neighbor problem solving algorithm which has promising accuracy and simplicity, especially in the first story detection-based tasks [53]. Unlike the traditional pairwise spatial distance measurements, locality sensitive hashing utilises hashing tables to reduce the number of straight document-to-document comparisons. The introduced hyperplane in locality sensitive hashing boosts the computation and also automatically classify the similar sentences into nearby 'districts'. K-term Hashing employs a hashed memory to store the most significant K features [189].

3.1.1 Vector Quantisation

Vector quantisation is a lossy approximation process which yields a subset of elements being encoded to represent the original vector space. Vector quantisation

was initially intractable to follow or apply in the real instances as it is practically infeasible to resolve the problem of multi-dimensional integration. Nevertheless, Linde et al. [50] proposed a training sequence based algorithm resolving the issue of multi-dimensional integration through iteratively substituting degree of distortion on the conditions of nearest neighbouring and centroid. The optimal combination of codebook and partitions of the space are achieved when the stopping criterion are satisfied [50].

3.1.2 Non-negative Matrix Factorisation

According to Lee and Seung [51], Non-negative Matrix Factorisation has been shown to be a useful decomposition for multivariate data and can be formulated as a minimisation problem with bound constraints. Traditional unsupervised learning algorithms such as PCA, vector quantisation, singular value decomposition, and its advanced variant latent semantic analysis can be understood as factorising a data matrix subject to different constraints. Depending upon the constraints utilised, the resulting factors have distinguishable properties in representation and expression. PCA enforces a weak orthogonality constraint, resulting in a very distributed representation that uses cancellations to generate variability. On the other hand, vector quantisation uses a hard winner-take-all constraint that results in clustering the data into mutually exclusive prototypes. However, non-negativity is a useful constraint for matrix factorisation that can learn a parts representation of the data [52]. The non-negative vectors learned are used in distributed, yet still sparse combinations to generate expressiveness in the matrix factors reconstructions. In other words, Non-negative Matrix Factorisation is useful for finding representations of non-negative data.

Given a non-negative matrix V, find two non-negative matrix factors W and H such that:

$$V \approx WH. \tag{3.1}$$

Non-negative Matrix Factorisation is usually applied to solve the following two popular methods: multiplicative update and projected gradient methods, with the latter one converges faster than the former one, based on the experimental results shown by Lin [190]. Because the bag-of-words matrix representation of sentences is composed by word counting or frequency thus every entry is non-negative by definition. Non-negative Matrix Factorisation is also suitable for document clustering, computer vision, and document classification scenarios.

3.1.3 Latent Semantic Analysis

Latent semantics indexing or latent semantic analysis explores correlation between terms and documents. Formally speaking, two terms are correlated or share similar semantic concepts if they often co-occur. Similarly, two documents are correlated (share similar topics) if they have certain many common words. Latent semantics indexing operates by associating each term and document with a small number of lexical concepts/topics. Latent semantics indexing uses a singular value decomposition of the matrix to identify a linear subspace in the space of term frequency-inverse document frequency features that captures most of the variance in the collection. This approach can achieve significant compression in large collections of texts. Furthermore, Deerwester et al. [39] argued that the derived features of latent semantics indexing, which are linear combinations of the original term frequency-inverse document frequency features, can capture some aspects of basic linguistic notions such as synonymy and polysemy. Latent semantics indexing is measured by the importance of concept and reflects error of approximating the original matrix of documents with terms. Singular value decomposition is a very popular technique for latent semantics analysis.

3.1.4 Fisher Linear Discriminant

Fisher linear discriminant is a supervised vector space based dimensionality reduction technique [188]. Fisher linear discriminant has application in face recognition, computer vision, pattern recognition, and artificial intelligence area [188]. Fisher linear discriminant attempts to distil most prominent features out from the original high dimensional vector space model. It then generates relaxed discriminate functions in which the projected distance within same class is minimal and between class distance is maximal [191].

3.1.5 Principal Component Analysis

PCA is an unsupervised data compression scheme normally applied in data preprocessing. It decomposes the original data matrix by projecting points onto a reduced plane with maximal covariance. Unlike PCA, fisher linear discriminant can only yield a matrix upto the number of distinct classes minus one. Fisher linear discriminant tries to reduce the high dimension data points with regard to optimal separation of target labels. However, PCA can result in a matrix upto the size of the given matrix's dimensionality.

3.1.6 Locality Sensitive Hashing

Many existing data compression methodologies discussed in the literatures are computationally expensive due to the brute force document to document comparison. Locality sensitive hashing has recently received due attention because of simplicity and its proximity in accuracy. Locality sensitive hashing was first introduced in 1998 [192] and has since demonstrated its superiority in increasing computational speed on large dimensional dataset by approximate nearest neighbor problem searching. Approximate nearest neighbor problem is often described by (c, r) where c simply stands for the approximation factor and is usually greater than 1 and r denotes the distance factor. Locality sensitive hashing has two forms with respect to data

dependency: i.e., the classical data independent partitioning scheme and the data dependent partitioning [40]. The improvement is addressed by escaping the curse of dimensionality with optimal space and query complexity time. Petrovic et al. [53] approximated one nearest neighbour with locality sensitive hashing.

It has been suggested that locality sensitive hashing is capable of improving computational time complexity on high dimensional data driven nearest neighbour searching [53]. Unlike the traditional one nearest neighbour based framework, locality sensitive hashing does not require exhaustive computation of the distance of each pair of documents. Locality sensitive hashing needs to only work out the cosine similarity values upon the trimmed hash buckets. The essence of locality sensitive hashing is that if two documents are far from each other in the original high-dimensional space, then the probability of them hashing into the same hash bucket is low. In other words, the possibility of two given documents hashing into the same bucket is proportional to their distance in the original high-dimensional space.

Signature bits are utilised to calculate the Hamming distance between the two hashed documents. Random projection produces signature bits by separating hyper planes. More specifically, in a vector space, if the value of a target term weight is underneath the corresponding hyper plane, the signature bit is assigned number 0 to the corresponding cell, otherwise signature bit 1 is assigned.

Experiments on single hashing table indicated that this configuration was not able to effectively identify the specificity of the training dataset. A new control variable L for multiple hash tables' construction is required.

$$\mathrm{L} = \log(\phi - p^k), \tag{3.2}$$

where $p = \frac{\theta(x,y)}{\pi}$ and ϕ and θ are two constants for the probability of missing a nearest neighbour.

Each of the L hash tables applies the same procedure as aforementioned and a union operator is deployed subsequently to get an overall average. By introducing

the variable L, the cosine similarity measurement generates much more coherent result as compared with a single hash table.

3.1.7 K-term Hashing

K-term hashing-based first story detection model stores the most salient K feature terms of previously encountered documents in a hashed memory [189]. K-term hashing implements feature reduction for processing large scale document base [187].

3.2 Sentence Distance Functions

A description on cosine distance [11], Euclidean distance [193], Manhattan distortion, earth mover's distance [26, 98], and word mover's distance [23, 24] is presented in this section.

3.2.1 Cosine Distance

Cosine distance is an off-the-shelf mechanism for sentence/word similarity calculation [54, 11]. In cosine distance modality, difference between sentence q and query d is depicted as

$$\text{cosine}(q, d) = \frac{\sum_{i=1}^{n} q_i d_i}{\sqrt{\sum_{i=1}^{n} q_i^2}\sqrt{\sum_{i=1}^{n} d_i^2}}, \tag{3.3}$$

where sentence q and d are bag-of-words-based vector space model in cosine space $\mathbf{R}^n$.

3.2.2 Euclidean Distance

K nearest neighbour-Euclidean and hierarchical agglomerative clustering-Euclidean algorithm employ regular Euclidean metric or distance as the similarity metrics [51]. Euclidean distance is aka Pythagorean metric [194]. For sentence categorisation tasks, n dimensional standardised Euclidean distance is often applied. It can be

formulated as follows:

$$\text{Euclidean}(s, m) = \sqrt{(s_1 - m_1)^2 + (s_2 - m_2)^2 + \cdots + (s_n - m_n)^2}, \tag{3.4}$$

where sentence s and m are bag-of-words-based vector space model in Euclidean space $\mathbf{R}^n$ [195]. Euclidean space is analogous to L_2 norm in the context of L space [196].

3.2.3 Manhattan Distance

Manhattan distance between sentence a and b is defined as the sum of horizontal and vertical paths or components.

$$\text{Manhattan}(a, b) = \sum_{i=1}^{n} h(a_i, b_i) + \sum_{i=1}^{n} v(a_i, b_i), \tag{3.5}$$

where h stands for horizontal path and v denotes vertical path. Sentence a and b are bag-of-words-based vector space model in Manhattan space $\mathbf{R}^n$.

3.2.4 Earth Mover's Distance

In mathematics, the Wasserstein or Vasershtein metric is a distance function defined on two probability distributions for a given metric space M [26]. Intuitively, if each distribution is viewed as a unit amount of 'dirt' piled on M, the Wasserstein metric tries to obtain the minimum cost of turning one pile into the other, which is assumed to be the amount of dirt that needs to be moved multiplies the distance it has to be travelled [27]. Due to this analogy, the metric is known in computer science as earth mover's distance. Earth mover's distance was originated from a solution to the well-known transportation problem [26, 98]. The use of earth mover's distance has been pioneered in the computer vision literature, according to Rubner et al. [26] and Ren et al. [28]. Several publications reported approximations of earth mover's distance for image retrieval applications. Earth mover's distance intuitively captures the minimum cost required to transform one distribution into another and is robust against outliers and slight probability shifting.

For example, a group of suppliers needs to deliver goods for several consumers, with each supplier having a constraint on capacity and the cost of transporting one unit for each pair is pre-defined. The transportation problem is then to fetch an optimal goods flow which minimise the overall transportation cost and fulfil consumers' requirement. Earth mover's distance in computer vision is typically solving a linear programming problem. Suppose $P = (p_1, w_{p_1}), \cdots, (p_m, w_{p_m})$ is the first distribution with m clusters. Each pair in P is composed by the cluster pivot p_i and the cluster weight w_{p_i}. Similarly, the second signature Q can be denoted as $Q = (q_1, w_{q_1}), \cdots, (q_n, w_{q_n})$ with n clusters. The ground distance between cluster p_i and q_j is defined by $D = [d_{ij}]$. Users seek to find a flow $F = [f_{ij}]$ that minimises the overall cost.

3.2.5 Word Mover's Distance

Word mover's distance extends the concept of earth mover's distance to textual mining by elaborating distributed word embeddings like word2vec and global vectors for word representation on a bag-of-words term matrix. Because of the associated highly informative word embeddings, document retrieval can measure the spatial similarity of two given sentences by adapting the classical earth mover's distance. Formally, under Kusner et al. [23] implementation, word mover's distance was formalised to solve a linear optimisation problem by minimising:

$$\sum_{i,j=1}^{n} G_{ij} w(i,j) \tag{3.6}$$

subject to the following two constraints:

$$\begin{aligned} \sum_{i=1}^{j} G_{ij} = d_i \quad \forall i \in \{1, ..., n\} \\ \sum_{j=1}^{j} G_{ij} = d'_j \quad \forall j \in \{1, ..., n\} \end{aligned} \tag{3.7}$$

with $w(i, j)$ being the word2vec embedding distance between word i and j which is usually a constant given two fixed terms and $G_{ij} \geq 0$ being the proportion of term i in the document d moves to term j in document d'. In other words, the optimum travelling distance between two given documents d and d' is defined as a transportation problem where the discrepancy between the normalised bag-of-words distribution of d and d' are to be computed and named as word mover's distance. Because each word carries a fixed word embedding in the vocabulary, word travel cost between any two tokens can be derived through the Euclidean/cosine/Manhattan distance representation in the corresponding embedding space. The normal word mover's distance takes quadratic time complexity which may lead to high computational overhead [24]. A relaxed RWMD reduces to one hard constraint on the regular linear programming problem and has better time complexity, following with Kusner et al. [23].

3.3 Baseline Methods for Sentence Classification

This section gives an overview of various models involved in the word mover's distance-based supervised sentence classification performance comparison.

- **SVM** is first introduced in COLT-92 by Boser, Guyon, and Vapnik and became rather popular since then [124]. Vector space model is a theoretically well motivated algorithm developed from statistical learning community and has empirically robust performance across applications in text classification/regression, image recognition, and bioinformatics [197]. A SVM is a discriminative classifier formally defined by a separating hyperplane. In other words, given labelled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorises new examples [198].

- **Locality sensitive hashing** is one of the first attempt to finding nearest neighbour within sublinear time or approximate nearest neighbour, according

to Indyk and Motwani [192]. It hashes each query point into the corresponding buckets in such a manner that the possibility of of collision was much higher for points that are near by. The naïve way of dealing with how many layers of hyperplanes for locality sensitive hashing was through a fixed constant, as in Petrović et al. [53].

The naïve way of dealing with how many layers of hyperplanes for locality sensitive hashing is through a fixed constant, as stated in Petrović et al. [53]. However this not captures the characteristic of the term-by-document vector space matrix and leaves term frequency-inverse document frequency weighting scheme undermined. Towards this end, in the proposed architecture, the current work sets a flexible variant h to alleviate the problem with fixed parameter setting in the plain locality sensitive hashing:

$$h = c \sum_{i=1}^{t'} \sum_{j=1}^{d'} w_i s_j, \tag{3.8}$$

where w_i depicts the term i in a specific sentence indexed as s_j and c is a light tuning constant serving as a scale [11].

- **Gaussian Naive Bayesian** is a supervised learning algorithm based on applying the Gaussian Naive Bayes algorithm for classification [167]. The likelihood of features is assumed to be Gaussian with maximum likelihood for the parameters [167].

- **Multinomial Naive Bayesian** is a supervised learning algorithm implements the Naive Bayes assumption of independence for multinomially distributed data, and is one of the two classical naive Bayes variants used in text classification [162]. It is typically used in the situation where the textual snippets are represented as word vector counts or term frequency-inverse document frequency vectors. The parameters are estimated by a smoothed version of maximum likelihood, i.e., relative frequency counting [162].

- **Bernoulli Naive Bayesian** employs the naive Bayes training and classification algorithms for data that is distributed according to multivariate Bernoulli distributions [160]. Each feature is assumed to be a binary-valued variable or boolean. In the circumstance of this work, this algorithm naturally suits a binary one-hot vectoriser vector space model. If handled with any other kind of data, a BernoulliNaive Bayesian instance can binarise its input according to the binarise parameter [160]. The decision rule for BernoulliNaive Bayesian differs from its sibling multimonialNaive Bayesian's in that it explicitly penalises the non-occurrence of a feature term. A feature is an indicator for class label. In other words, the multinomialNaive Bayesian simply omits a non-occurrence feature instead.

3.4 Sentence Categorisation Evaluation Metrics

This section presents various evaluation frameworks relating to this work. Popular supervised text classification evaluation methods such as precision rate, recall rate, F1 score, and accuracy are discussed here [36]. For unsupervised sentence categorisation learners, accuracy score, V-measure, completeness score, and homogeneity score are demonstrated [199]. Other evaluation bodies like neural network specific perplexity score is given as well. For training/testing data split strategy, the cross validation scheme is presented. For the statistical validation, a group of significance test are illustrated. For instance, Mann-Whitney U test is suitable for non-parametric two conditions significance evaluation where no assumption of normally distributed data is required. For sign test, it can be applied to assess a repeated-measures study and matched pairs comparison. Sign test is an instance of non-parametric statistical method as well.

3.4.1 Machine Learning Classification Performance Evaluation

This subsection describes standard machine learning based classification system assessment. Precision rate, recall rate, F1 score, and accuracy score are four widely applied techniques. Those aforementioned four metrics are analogous to the basic objective of any web search provider. The query intent on any Internet search engine is that users prefer only the most relevant material, while minimising the junk information. Precision ratio determines the percentage of predicted TP sentences out from the whole prediction pool, i.e., $\frac{\text{TP}}{\text{TP+FP}}$. Recall rate on the other hand, measures the proportion of retaining TP samples out from all true sentences, i.e., $\frac{\text{TP}}{\text{TP+FN}}$. F1 score is defined as the harmonic mean of precision ratio and recall rate. It can be formulated as $\frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision+Recall}}$. Accuracy score simply measures the difference between predicted values and the ground truth. Under binary and multi-class classification, accuracy score is equivalent to Jaccard index or Jaccard similarity coefficient [92]. It differs in the multi-label categorisation problems. Jaccard similarity coefficient is defined as the proportion of the size of the intersection within the union of the sample collections.

Precision rate and recall ratio work as a foundation for intuitive graphical machine learning classifier output quality mechanism such as precision-recall curve and ROC curve. Precision-recall curves shows the trade-off between precision ratio and reall rate. Precision-recall curves are typically used in binary classification to study the performance of a classifier. ROC curve is a kind of plot showing trade-off between TP rate or sensitivity on the Y axis and FP rate or 1-specificity on the X axis. Usually a larger AUC in ROC demonstrates better test accuracy on the diagnostic binary classification. The 'steepness' of ROC curves is another important metrics in classifier evaluation. It is ideal to maximise the TP rate while minimising the FP rate during the ROC.

Other than the above standard machine learning classifier measures, perplexity

score and detection error trade-off curve tend to be more sentence categorisation related metrics. Perplexity score has good coverage in the discrete or continuous probability distribution and probability model. When used in the domain of probability model, it bares much similarity to another error measurement called cross-entropy [200, 201, 202]. When applying the perplexity analysis to machine learning, three different interpretations can be derived under certain situation, namely how easy a probability distribution is to predict, how variable a prediction model is, and a quantitative scale of prediction error [202].

Topic Detection and Tracking evaluations employ a cost function to address the prior probability that a story belongs to a topic [203]. Normally the Topic Detection and Tracking cost function is engineered as a linear function of the probability on missing a target event and a false alarm [203]. Later, a normalised version was introduced to scale the cost function down to a value between 0 and 1 inclusive [11, 13]. Under the normalised situation of detection error trade-off, a perfect system would score 0, and a naive Topic Detection and Tracking evaluation system which always yields yes or no scores 1 correspondingly. Nevertheless, the fact that the Topic Detection and Tracking evaluation systems presume a constant cost for the C_{miss} and C_{false} is not appropriate in many circumstances. Therefore Manmatha, Feng, and Allan [203] proposed to leverage between providing a threshold leading to a consistent performance across events and generating a minimum cost function.

A detection error trade-off curve tries to cast the Topic Detection and Tracking tasks by measuring the probability of miss and false alarm rate (P_{miss} and P_{FA}), providing that the specified Topic Detection and Tracking task is evaluated as retrospective detection task. According to Alvin et al. [58], the detection error trade-off curve has significant advantages against the plain ROC curve rendering on the large vocabulary based speech recognition tasks and language perception assessment. More specifically, the detection error trade-off curve explicitly involves the trade-offs of two error types whereas the traditional ROC curve does not [59].

3.4.2 Cross Validation

Cross validation is a statistical analysis protocol normally employed to validate the system's robustness on sentence categorisation performance whilst mitigating overfitting issues [204]. This experiment follows a threefold cross validation where two-thirds are training set and the remaining one-third is the testing partition where applicable.

3.4.3 T-Distributed Stochastic Neighbour Embedding

The word mover's distance-specific evaluation metrics and visualisation methods are described now. More formally, the Mann Whitney U statistical test is applied to validate the cross-validated performances. System runtime or elapsed time is charged to produce a ranking for each participating algorithm. It also indicates the computational efficiency of the specific framework. For the system execution time comparison on k nearest neighbour-word mover's distance and plain k nearest neighbour, evaluation data size from the range of 600, 800, 1000, and 1200 were set to quantify the time complexity and performance difference across data size. Likewise, hierarchical agglomerative clustering-wmd and existing hierarchical agglomerative clusterings have the similar operational time difference.

For the visualisation on discrepant distance metrics, an award-winning nonlinear high dimensional data visualisation toolkit termed t-distributed stochastic neighbour embedding was employed [205, 206]. T-distributed stochastic neighbour embedding attempts to reduce a high spatial data points into a two/three dimensional plane without compromising the latent distance contained in the original space. A relatively low dimensional hyperplane facilitates intuitive view of distance functions difference [206]. It also offers the functionality to map and visualise the reduced space onto a graphical interface for interactivity and interpretation. Thus in the work, word mover's distance and {Euclidean, cosine, Manhattan} are paired each for the supervised categorisation and unsupervised tasks. Each pair is fed into the

t-distributed stochastic neighbour embedding framework for showing the superiority of word mover's distance method across various benchmarking data-set. For the t-distributed stochastic neighbour embedding specific parameters setting, the present work applied the same initialisation of embedding. The rest variables are defined as follows.

Table 3.1: The parameters setting for the T-SNE-specific distance visualisation. The balancing angle was configured as the trade-off between accuracy and speed.

Dimensionality of the embedded space	2
Perplexity	30
Gradient calculation algorithm	Barnes-Hut-sne [207]
Balancing angle	0.5

3.4.4 Significance Test

Because sentence categorisation tasks involve repeated experiments on cross-validated data-sets configuration, statistical test is required to validate the efficacy of participated sentence categorisation algorithms. The most relevant significance tests are non-parametric ones [208] since they make no assumptions on the distribution of system's performance in numeric values. As such Mann-Whitney U Test [209, 210] and Sign Test [211, 212] are two relevant candidates.

Mann-Whitney U test is a non-parametric or distribution free test where two independent groups or conditions of continuous values can be compared. It is ideal when employed to decide whether the null hypothesis satisfies, i.e., the medians of the two samples are identical [209, 210]. Mann-Whitney U test is based on numerical difference of elements in each of the two samples.

Sign test is also a distribution free statistical test that is usually used to test whether or not two conditions are equivalently sized [212]. It is applicable when the dependent variables are grouped in pairs or matched pairs. It intends to assign a plus

or minus sign between matched pairs based on the difference rather than numerical magnitude. Sign test is utilised to determine whether a binomial distribution has the equivalent chance of success and failure [211, 212].

3.5 Contribution to Methodology

This chapter contributes to the field of sentence categorisation methodology by the following points:

- Reviewing a list of relevant dimensionality reduction methods and their potential in the sentence categorisation tasks;
- A practical guide on how to assess sentence categorisation framework's performance based on machine learning mechanisms and statistical significance test;
- This chapter proposes a series of effective and efficient unsupervised sentence categorisation evaluation metrics which can be applied to future works for other practitioners;
- The methodology chapter gives a strong example of using t-distributed stochastic neighbour embedding as an intuitive approach of visualising sentence distortion function's efficacy and efficiency.

3.6 Conclusion

This chapter emphasises various methods, approaches, and evaluation metrics including a list of seven textual data dimensionality reduction methods, spatial distortion functions, baseline algorithms for supervised learning, and evaluation metrics for sentence categorisation. The next chapter will discuss experimental data-sets, correspondent pre-processing steps for textual snippets, and the configuration of

experimental setup for word mover's distance-based evaluation and convolutional recurrent neural network-oriented performance comparison respectively.

Chapter 4

Data-sets and Experimental Setup

In this chapter, the benchmark data sets and preparation on the experiments are presented. The associated pre-processing techniques for each data stream is also stated in the present work. In general, data pre-processing process is crucial in order to eliminate the potential impacts posed by human annotation, babbles, and stop-words. Sentential data pre-processing offered in the current chapter also leads to an efficient bag-of-words matrix as low frequency terms are removed from original data streams. The current chapter of research also provides the detailed software platform and test bed environment for reference purpose. The enclosed benchmarking data sets for the word mover's distance examination are split into supervised classification and unsupervised categorisation respectively. Deep neural network-based sentence classification bares much similarity in terms of benchmark data provision. Thus the two involved supervised data streams in the context of word mover's distance experiment are also applied in the convolutional recurrent neural network evaluation.

4.1 Datasets for Word Mover's Distance

In an experiment conducted in the present research, two categories of data streams are established: one for supervised experiment and the other for unsupervised task. The characteristics of each data set are shown in Table 4.1 below. Bag-of-words de-

notes the total quantity of terms contained in a corpora. The bag-of-words statistics reported in Table 4.1 was collected for the original data sets.

Table 4.1: Data sets characteristics. M.S.L stands for mean sentence length. Lan denotes the language used in the specific sentence of the data-set. En represents the English language.

Data	Lan	**Size**	**BOW**	**M.S.L**	**Labelled**	**Classes**
Movie-review	EN	10662	18765	21.0	No	-
Sms-spam collection	EN	5574	7543	15.6	No	-
Question classification	EN	5952	8634	5.4	Yes	50
Twenty-news-groups	EN	18000	31341	159.6	Yes	20

As can be seen from Table 4.1, the unsupervised branch subsumes two clustering data sets: movie review data [74] and the sms-spam collection [213]. The number of movie reviews content in the data set is 10662 with one sentence per review content. The bag-of-words feature vector spans the dimension up to 18765.

Sms-spam messages collection was originally collected by Almeida et al. [213] in a pioneering work for short messages spam filtering. The data set is gathered through legitimate online sources such as the Grumbletext Web site from a UK mobile forum, NUS SMS Corpus, and Tagg's PhD thesis [214]. The whole corpora is made up by a total of 5574 sms texts including both legal and spam messages. The bag-of-words statistics for the sms-spam collection is 7543.

Question classification was created by Li and Roth [60]. Question classification collection contains 50 classes of questions. The padded vector space size is 8634 which means the longest sentence has the length of 8634 or 8634 tokens. In the case of question classification data-set, a reliable ground truth collection is pre-mounted which eliminates the problem of human annotator as the leading decision maker.

Twenty-news groups data is another well-known text categorisation dataset which normally appears at various classification tasks. Twenty-news groups corpus is com-

posed by 18,000 news articles with up to 20 different categories. Because news industry usually produces relative long stories on a specific event of interest, the contextual data captured is more complete than other media sources, thus the bag-of-words column embraces a high 31341 word vector dimension. The present work chooses to serve the textual snippets filtering systems with twenty-news groups data as it is a hand-crafted premium stream with evenly distributed texts across 20 classification labels referred by document classification frameworks, image recognition tasks, NLP systems, and sequence learning oriented neural networks [215, 216].

Data pre-processing is critical in natural language processing as the quality of the returning tokens may reflect the predictive efficacy of the modelling process. For data cleaning or pre-processing, a set of operations were performed in order to discard the potential effects caused by unnecessary factors like human errors and stop-words. In this thesis, sentential snippets tokenisation, stop-words removing, tokens replacing by their stems [217], stems refining by lemmatisation facility, and inverse sentence frequency vectors generating on a cut-off vector space model were conducted in sequence. As a normal Porter stemmer [217] may not necessarily recognise the POS of a token, and therefore some duplicated instances of an identical stem are neglected. The process of lemmatisation, however, can further split the above omitted instances down to more precise groups of stem words and thus result in a more representative term-by-sentence matrix. For the cut off vector space matrix, the practitioners first obtained the global sentence frequency for each feature or term in the model. They then performed a distilling process to filter out the terms with their global term frequency less than a threshold, which can help reduce the dimensionality of the features vector and thus increase the implementation efficiency on the sentence categorisation system. The threshold for lowest term frequency in the present experiment is 10.

4.2 Experimental Setup for Word Mover's Distance

As the purpose of stacking up a sentence categorisation-based system is to assign label to the training set, suggesting that precision rate, recall ratio, and F1 score (aka balanced F-score or F-measure) on the supervised sentence categorisation represent efficacy of the system. For the unsupervised situation, performance measure is based on the accuracy score, homogeneity score, completeness score, and V-measure [199] instead. In line with Kusner et al. [23], word mover's distance is an expensive operation with time complexity $O(p^3 \log p)$ where p denotes the number of distinct feature words in a sentence, and even its relaxed version still suffers from strong computational overhead. To this end, the work presented in the thesis decided to randomly select part of the dataset in the experiments to save overall experimental runtime. For the supervised classification, the evaluation set contains 600 sentences, including 400 testing samples and the remaining 200 are training textual snippets. For the unsupervised tasks, the evaluation pool subsumes 1000 sentences in total without any training/testing split.

There are mainly two parts involved in the word mover's distance comparison, namely sentence categorisation performance evaluation on the participated systems and t-distributed stochastic neighbour embedding-oriented sentence dissimilarity metrics assessment. For the list of participating competitors, the work included the plain k nearest neighbour-{cosine, Euclidean, and Manhattan}, SVM, and locality sensitive hashing for the supervised classification tasks and hierarchical agglomerative clustering-{cosine, Euclidean, and Manhattan} for the unsupervised sentence categorisation tasks.

Table 4.2 below lists the major algorithmic comparison and its evaluation in details.

The earth mover's distance metrics implemented in this thesis follows [27, 218]. For multi-labelled classification, the F1 score is computed according to Godbole

Table 4.2: The list of designed experiments including algorithmic performance comparison based on benchmarking data-sets and distance metrics evaluation. The evaluation on performance difference is validated by the statistical family involving the Mann Whitney U test. TFIDF stands for term frequency-inverse document frequency, T-SNE denotes t-distributed stochastic neighbour embedding, WMD refers to word mover's distance, MSE denotes mean squared error, and LSH represents locality sensitive hashing. RWMD [23] refers to the relaxed version of its sibling word mover's distance. KNN refers to k nearest neighbour classifier. Naive Bayesian classifiers include three options: Gaussian Naive Bayes [167], Multinomial Naive Bayes [162], and Bernoulli Naive Bayes [160].

Comparison	**Evaluation Criteria**	Note
TFIDF, count vectoriser, and binary one-hot	Processing speed Vector space model distribution Sentence encoding T-SNE	Sentence representation comparison
Euclidean, cosine, Manhattan, EMD, WMD, and RWMD [23]	Elapsed time T-SNE	Distance metrics comparison
word2vec, glove, and sent2vec	Operational speed Encoding performance	Feature embeddings comparison
KNN-{WMD, Euclidean, cosine, Manhattan}, SVM, Naive Bayesian classifiers, LSH	Precision ratio Recall rate F1 score Best number of neighbours Mann-Whitney-U test	Supervised baseline comparison
HAC-WMD, and HAC-{cosine, Euclidean, and Manhattan}	MSE Completeness score Homogeneity score V-measure score Mann-Whitney-U test	Unsupervised baseline comparison

and Sarawagi [219]. For the K-neighbours classifier, a grid search algorithm [220] is employed to seek the optimum k neighbours value.

Previously the work presents the algorithm performance comparison, now the t-distributed stochastic neighbour embedding-sponsored sentence-level distance visualisation is described. More formally, word mover's distance-based and traditional {Euclidean, cosine Manhattan} distance metrics are to be compared on supervised learning and unsupervised sentence categorisation tasks respectively. The actual design and manipulation of this evaluation is demonstrated in Table 4.3 below.

Table 4.3: T-distributed stochastic neighbour embedding-based sentence distance function visualised comparison. QC stands for the question classification stream. RWMD [23] refers to the relaxed version of its sibling word mover's distance. KNN refers to k nearest neighbour classifier.

Distance Function	**Data**	**Algorithm**
word mover's distance RWMD earth mover's distance Euclidean	Twenty	KNN
word mover's distance RWMD earth mover's distance Euclidean	QC	KNN
word mover's distance RWMD earth mover's distance Euclidean	SMS	hierarchical agglomerative clustering
word mover's distance RWMD earth mover's distance Euclidean	Movie	hierarchical agglomerative clustering

4.2.1 Word Vectorisation Parameters and Configuration

The pre-processing is undertaken in prior the input features engineering. The standard natural language processing techniques such as removing stop-words and striping out the newsgroup-related meta-data including noisy headers, footers, and quotes for more realistic information presentation are performed in advance. Often raw bag-of-words feature vector for natural language processing related tasks has high dimensionality by nature, thus it seems reasonable to conduct some data cleaning techniques first [34]. Within the quantitative experiment, the low frequency tokens and high space complexity textual snippets are trimmed. More specifically, normalised bag-of-words oriented counting vector is constructed to generate a global uni-gram based dictionary mapping. Users can then get rid of low frequency terms and overly length sentences by querying on the uni-gram indexer. As a general rule-

of-thumb, it is usually good to set the boundary frequency to be ten and maximum length equals to five hundred for the specified benchmarking data source.

For the textual data encoding, necessary investigations are needed in order to determine the optimum sentence encoding scheme among the listed options. Term frequency-inverse document frequency, raw TF counting or plain count vectoriser, and binary one-hot vectorisation are considered for sentence vector space model construction. Details about term frequency-inverse document frequency scheme, count vectoriser, and binary one-hot encoding can be found at Table 4.4 and 4.5. Binary one-hot vectoriser applies almost identical parameters with the one used by the count vectoriser. The only exception is the value set and binary one-hot vectorisation only accepts 0 or 1. Latent semantics indexing, pure locality sensitive hashing, and latent dirichlet allocation-based encoding strategy are employed in the sentence matrix optimisation phase.

Later, the current thesis conducted an efficiency analysis on each of them with a random subset of benchmark data stream and observed that term frequency-inverse document frequency yielded balanced result with minimal time and space complexity. Term frequency-inverse document frequency has the robust performance because of the following two points. First, term frequency-inverse document frequency accurately captured the sentence structure and its semantic relatedness due to its inverse sentence architecture. Second, latent semantics indexing and latent dirichlet allocation are based on the topical information generation which caused negative influence on the dense word embedding. Simple arithmetic aggregation of an array of topical words not necessary represents the semantic and syntactical meaning of that theme. From the perspective of distributed word embeddings, bag of terms did not match with a single word embedding vector. As a consequence, often an arithmetic mean of each word2vec/global vectors for word representation vector had to be taken as the representative for the topical domain and the preserved word relationships were severely destroyed. For the enclosed pre-trained and real-valued word embeddings, word2vec and global vectors for word representation were deployed.

Table 4.4: The list of parameters with value for n-gram term frequency-inverse document frequency weighting scheme. **Smooth_idf** aims to prevent zero divisions by adding one to sentence frequencies. **Ngram_range** herein determines a vocabulary of uni-gram, bi-gram, and tri-gram inclusion formality within vector space model. Setting **lowercase** to True allows the programme to convert any uppercase letters to their lowercase version.

Encoding	'utf-8'
Strip accents	'unicode'
Feature analyser	'word'
Smooth_idf	True
Sublinear_tf	Replace tf with 1+log(tf)
Norm	L2
Ngram_range	$(1, 3)$
lowercase	True

Table 4.5: The list of parameters with value for count vectoriser weighting scheme. **Decode_error** gives instruction on what to do if a byte sequence is given to analyse that contains characters not of the given 'utf-8' encoding. 'Strict' means a Unicode Decode error will be thrown. **Ngram_range** herein determines uni-gram formality within vector space model. **Binary one-hot** sets to False to form an occurrences vector space model.

Encoding	'utf-8'
Strip accents	'unicode'
Feature analyser	'word'
Decode_error	'strict'
Ngram_range	$(1, 3)$

For the binary one-hot vectorisation, the parameters setting follow almost identical configuration as opposed to the count vectoriser. The only exception or difference is that word appearance is in a binary format (i.e., 0 for absence and 1 for presence).

As heuristics was implemented in the participating locality sensitive hashing algorithm [53], indicating that stochastics may occur during a single run, it is therefore a necessity to report the averaged result based on several repetitive run with identical environmental parameters setting. The same repetition strategy was taken for every other algorithm in the experiment for consistency. This repetition strategy also conforms with the statistical verification of the experiment result. The current experiment provided in the thesis applies twenty runs of repetition under this circumstance.

4.2.2 Hierarchical Agglomerative Clustering Parameters and Configuration

In accordance with Rosenberg and Hirschberg [199], a clustering result satisfies homogeneity if all of its clusters contain only data points which are members of a single class. The available range of homogeneity is between 0.0 and 1.0 and 1.0 means perfectly homogeneous labelling. A clustering result satisfies completeness if all the data points that are members of a given class are elements of the same cluster [199]. The completeness score is between 0.0 to 1.0 and 1.0 stands for perfectly complete labelling. The V-measure takes the form of harmonic mean between homogeneity and completeness as follows:

$$\text{V\_measure} = \frac{2 * (\text{homogeneity} * \text{completeness})}{\text{homogeneity} + \text{completeness}} \tag{4.1}$$

Similar to the usual automatic document classification tasks, the proposed novel sentence categorisation algorithm paces a way to identify their label amongst the incoming unclassified textual snippets. For the optimal hyper parameters seeking in the system, a fast cross-validated grid search algorithm inherited from the Python

Table 4.6: Shared parameters setting for hierarchical agglomerative clustering-word mover's distance and normal hierarchical agglomerative clusterings. Max_iteration refers to the maximum number of iterations of the hierarchical agglomerative clustering algorithm for a single run.

#clusters	$(10, 20, 30)$
Max_iteration	300

scikit-learn library was applied [220]. Likewise, for the unsupervised classification tasks, optimal parameters were chosen by the enclosed scikit-learn cross-validated grid search approach as well [220].

4.2.3 Test bed for Word Mover's Distance

For the test bed, the present research deployed the codebase on the popular python scikit-learn environment [220] for Ubuntu version 16.04 operating system. The codebase was executed on the Spyder IDE for the intensive programme evaluation. The hardware platform was configured on an Intel eight core CPU, 16GB RAM conventional desktop computer. The computational load of processing WMD calculation was handled by the multi-processing units built in the scikit-learn library [220].

4.3 Datasets for Character Recurrent Neural Network

As the major focus of this work is to develop convolutional recurrent neural network for enhancing the topics classification, the appropriate benchmark data-sets would be natural textual streams or raw sentential data. This thesis utilised four popular classification data streams from the Internet for this study. Each benchmark stream contains only English language-based sentences. They are listed below:

- **Google-news** is a small portion of excerpts from the online platform giant Google [49], specifically the Google news channel.

- **Twenty-news-groups** was originated from the well-known American newspaper publishers formed as twenty-news-groups, probably first appeared at Lang's work [221].

- **Brown Corpus** of Standard American English, often abbreviated as the Brown Corpus [222]. The Brown Corpus encompasses with one million tokens of American English texts sampled from 15 different textual categories. The Brown Corpus is created by Francis and Kucera at Brwon University in 1960s [222].

- **Question Classification** was created by Li and Roth [60]. The question classification collection contains 50 classes of texts. The specific data-set characteristic is shown at Table 4.7 below.

Table 4.7: Data-sets statistics. QC stands for Question Classification collection. EN denotes English language.

Data	**Google-news**	**Twenty-news**	**Brown Corpus**	**QC**
Size	2066	18000	57340	5952
Vocabulary	11194	252999	55528	8634
Total Words	49988	1806249	649408	32356
Mean Sent. Length	24.2	159.6	11.1	5.4
Meta-data	No	Yes	Yes	Yes
Classes	55	20	15	50
Language	EN	EN	EN	EN

Table 4.7 reports the summary statistics based on the benchmarking data corpus. The work addressed the specificity of data with a six-dimension schema:

1. the number of sentences in the data-set

2. vocabulary of the stream

3. the total number of terms in a stream

4. average sentential snippet length

5. meta-data inclusion

6. the number of labels or classes of the corpus

During pre-processing in prior input features feeding, the current work in the thesis adopted standard natural language processing techniques such as removing English stop words and striping off the newsgroup related meta-data, which includes noisy headers, footers, and quotes. The vocabulary processor for the word-level convolutional neural network and word-level recurrent neural network, SVM [198], k nearest neighbour-word mover's distance [23], and plain k nearest neighbour requires the normalised bag-of-words features when constructing the corpus vectors. The normalised bag-of-words generated a global uni-gram based dictionary mapping. With the presence of the uni-gram indexer, the framework could readily remove low frequency terms and lengthy snippets. As a general rule-of-thumb, the work sets the bottom frequency to 10 and the maximum length to 500. On the other hand, character-aware recurrent neural network and character-level cnn employ a byte processor to map snippets into sequence of identities for bytes.

The present thesis evaluated the model and the others according to a cross-validated training/testing data split as shown in Table 4.8. For Google-news, the full corpus is selected. For twenty-news-groups, an arbitrary 1,100 snippets were taken from the total 18,000. 5,500 out of 57,340 were selected from the Brown Corpus and 5,500 out of 5,952 sentences were picked out from the question classification stream. Batch size defines a small amount of sentences involved in each epoch of training. For example, the batch size for the Google-news data is 50, indicating that each training epoch contains 40 samples. Similarly, the quantity of snippets for the

twenty-news-groups, Brown corpus, and the question classification stream is all 20 uniformly.

Table 4.8: Experimental testing/training split. Batch size denotes the number of samples used in each training epoch.

Data	Training	Testing	Batch Size	Total Words
Google-news	2000	66	50	1032
Twenty-news	1000	100	50	655
Brown Corpus	5000	500	250	4568
QC	5000	500	250	7665

4.4 Experimental Setup for Convolutional Recurrent Neural Network

This section specifically describes which algorithms participate in the experiment and what types of evaluation metrics are engaged in it. The section also presents the detailed configuration of each comparison algorithm with its corresponding input sequence encoding. There are in total twenty different algorithms participated in the experiment for convolutional recurrent neural network.

For non encoding options in the experiment, each is given a description on how the input textual snippet is translated according to its codification system.

Word2vec embeddings is a pre-trained word vector database which effectively learns and assigns a numerical array of real-valued numbers to a specific term [71]. Word2vec is trained on a ten billion Google-news stream and the each word embedding is a three hundred dimension vector [71]. Global vectors for word representation is different from word2vec in that a learned vector space representation is captured from a combined solution of global matrix factorisation and local context window technique [70]. Statistical information extracted from a word-word co-occurrence

scenario is efficiently applied to the word vector space learning process. Global vectors for word representation employed in this thesis is trained on a six billion tokens corpus which contains a 2014 Wikipedia dump and Gigaword 5 stream [70]. The dimension of a word vector representation in global vectors for word representation is three hundred as well.

Sent2vec or skip-thought vectors is an unsupervised sentence-level encoder-decoder learner which recognises a sentence as a vector. Since sent2vec by its own right uses no knowledge base or external information about sentence semantics, the issue of unseen words has to be solved by a vocabulary expansion mapping. This mapping is resolved by a vocabulary transition from word2vec domain to recurrent neural network word embedding space.

Another worth noting point is the choice of random or sent2vec in the comparison algorithms as shown in Table 6.2. The random encoding only works on word-aware neural networks simply because sent2vec is a sentence-level embeddings mechanism.

Character-level convolutional neural network is a nine layer deep neural network designed by Zhang, Zhao, and LeCun [17]. It interprets the input sequences by character encoding. It composes of six convolving modules and three fully-connected layers. Two dropout layers with probability of 0.5 are also included in between the three fully-connected layers for dynamic regularisation and overfitting prevention [223].

Character-aware recurrent neural network follows the traditional long short-term memory-based architecture developed by Hochreiter and Schmidhuber [22]. Character-aware recurrent neural network introduces a forget gate to facilitate the long-term memory learning, which is not presented in [22]. Character-aware recurrent neural network learning adopts classical stochastic gradient descent and back propagation.

Word-level convolutional neural network is an implementation from Zhang and Wallace [75]. Their model deploys one convolution layer on the sentence matrix. Temporal 1-max pooling is applied over each map generated by convolving layer. A

univariate feature vector is recorded for later softmax layer.

Word-level recurrent neural network is adopted from the original long short-term memory model [22] with the only exception that the input encoding is in word embeddings.

SVM is an classical classifier with radial basis function kernel [198]. It is suitable for data within a couple of 10000 samples [224]. Table 4.9 gives the detailed parameters setting for SVM configuration.

Table 4.9: SVM parameters. Variable gamma defines the kernel coefficient. Shrinking heuristic is enabled here.

Name	**Value**
Penalty	1.0
gamma	1/n_features
Shrinking	True

K nearest neighbour-word mover's distance is a variant of the plain k nearest neighbour algorithm [23]. It measures the distance between two distributions by word mover's distance metric [23]. However, in the traditional k nearest neighbour framework, the distance metric is Euclidean.

Evaluation on the sentence categorisation performance is measured by the following criteria. Examination on precision rate, recall ratio, F1 score, and loss value were demonstrated for each of four data-sets. The experiment is conducted on a cross-validated training and testing data split. The participants for the experiment are listed in Table 6.2. There are 20 frameworks in total for the experimental evaluation. The token random indicates the arbitrary initialisation of word vector.

Table 4.10 below lists the major algorithmic comparison and its evaluation in details.

Table 4.10: The list of designed experiments including algorithmic performance comparison based on benchmarking data-sets and distance metrics evaluation. The evaluation on performance difference is validated by the statistical family involving the Mann Whitney U test.

Comparison	**Evaluation Criteria**	Hypothesis
Each algorithm	Precision rate Recall ratio F1 score	There is significant performance difference on encoding

4.5 Conclusion

This chapter discusses the data-sets and accompanied pre-processing operations for word mover's distance-based sentence categorisation performance evaluation and convolutional recurrent neural network-based sentence classification examination respectively. The next chapter first gives an overview of KMeans algorithm, plain hierarchical agglomerative clustering algorithm, and traditional k nearest neighbour classifier. Second, two novel word mover's distance frameworks are presented, namely k nearest neighbour-word mover's distance and hierarchical agglomerative clustering-word mover's distance for supervised classification and unsupervised learning respectively.

Chapter 5

Count-based Sentence Categorisation

This chapter covers two sentence-level categorisation frameworks, namely supervised k nearest neighbour-word mover's distance and unsupervised hierarchical agglomerative clustering-word mover's distance. This chapter is organised as follows. First, a preliminary introduction on nearest neighbour algorithm and a definition of earth mover's distance are given. The essential elements for building up a novel word mover's distance-based sentence categorisation system are also given. Second, the proposed two novel models are described in detail in the present chapter. Third, experimental results are shown with further discussions in the current work. This chapter is finished by listing contributions to sentence categorisation and a conclusion section. The research questions and corresponding hypotheses this chapter are going to answer and test are presented as follows. Those hypotheses are specifically tested through section 5.4 and 5.5.

For the research question 'How does the expressiveness of term frequency-inverse document frequency differ from that of count vectoriser and binary one-hot vectorisation?', two hypotheses can be obtained.

(1) Given same quantity of sentences for vector space model matrix con-

struction, binary one-hot vectorisation obtains the best time efficiency, followed by count vectorisation and term frequency-inverse document frequency.

(2) Term frequency-inverse document frequency vectoriser provides better vector space model representation than that of count vectoriser and binary one-hot vectorisation, leading to better sentence categorisation performance.

From the research question 'Can word mover's distance-based distance function improve the accuracy of sentence similarity measurement over existing spatial distance functions?', the following hypotheses can be generated:

(1) Given same benchmark data stream, word2vec dense embeddings offers better word vector representation than global vectors for word representation distributed embeddings scheme.

(2) Word mover's distance provides better sentence classification performance than earth mover's distance and relaxed word mover's distance over standard machine learning metrics on the specified supervised benchmark data-sets.

(3) Word mover's distance provides superior sentence clustering performance against earth mover's distance and relaxed word mover's distance over standard machine learning metrics on the specified supervised benchmark data-sets.

(4) Word mover's distance-based distance function improves the accuracy of sentence similarity measurement over cosine, Euclidean, and Manhattan on the specified benchmark textual streams.

The research question 'How does the performance of word mover's distance-based sentence categorisation models differ from that of the other competing algorithms?' implies the following two hypotheses

(1) Given same testing sequences, k nearest neighbour-word mover's distance yields better performance than the other comparison frameworks with respect to precision rate, recall ratio, and F1 score.

(2) Given same testing sequences, Hierarchical agglomerative clustering-word mover's distance achieves better mean squared error value, homogeneity score, completeness value, and v-measure than the competing algorithms in unsupervised learning.

5.1 Classical K-Means, Hierarchical Clustering, and K-Nearest Neighbour

Cluster analysis or clustering in textual data can be defined as the unsupervised learning task of gathering a set of textual snippets based on the paired dissimilarity. The objective of KMeans is to seek the observational pattern in the data and find the common mean centroid vector for the subsequent grouping operation in an iterative way. The KMeans problem is usually solved by applying LIoyd's vector quantisation algorithm [225]. Since KMeans tries to minimise within-cluster variance, which implies that it involves no explicit pairwise distance function. KMeans may stop converging with distortion functions although data points to a centroid is implicitly computed during centroid update [88].

Hierarchical clustering algorithms use either top-down or bottom-up strategy [25]. Bottom-up algorithms treat each sentence as a singleton cluster at the outset and then successively merge (or agglomerate) pairs of clusters until all pools have been merged into a single cluster that contains all sentences [25]. Bottom-up hierarchical clustering is therefore abbreviated as hierarchical agglomerative clustering. Top-down clustering, on the other hand, requires a method for splitting a cluster. It proceeds by splitting clusters recursively until individual sentences are reached [25]. Hierarchical agglomerative clustering appears more frequently in information

retrieval than top-down clustering and is the main subject of this thesis.

A typical loop of hierarchical agglomerative clustering works as follows. Given a set of N items to be clustered, and an $N \times N$ distortion or distance matrix:

(1). Start by assigning each item to its own cluster, so that if you have N items, you now have N clusters, each containing just one item. Let the distances (similarities) between the clusters equal the distances (similarities) between the items they contain.

(2). Find the closest (most similar) pair of clusters and merge them into a single cluster, so that now you have one less cluster.

(3). Compute distances (similarities) between the new cluster and each of the old clusters, possible distortion function subsumes Euclidean, cosine, and Manhattan.

(4). Repeat steps (2). and (3). until all items are eventually clustered into a single cluster of size N.

Step (3). can be proceeded in different linkage criteria, which is what distinguishes single linkage with complete linkage and average linkage [25]. In this context, the linkage criteria determines which distance to use between sets of observation. The algorithm will merge the pairs of cluster that minimize this criteria.

- The defining feature of the single or ward is that distance between groups is defined as the distance between the closest pair of objects, one from each group.

- Complete or maximum linkage is the opposite of single linkage where the distance between the most distant pair of objects is considered as the distance.

- Average uses the average of distances between all pairs of objects, where each pair is made up of one object from each group.

Classification often refers to a standard machine learning task where a classifier is learned through an analysis on an array of class labels [226]. Plain k nearest neighbour is a typical instance of non-parametric classifier where the categorisation result is determined by majority vote [227, 228]. Generally speaking, sentential distance measure takes the Euclidean/cosine/Manhattan distance metrics as the dissimilarity computation when a range of k neighbours need to be decided. K nearest neighbour requires no background information on data distributional probability and the conditional probabilities, according to Gao and Xu [228].

5.2 Embracing the Tradition with Word Mover's Distance

Within this section, the two proposed frameworks are discussed in details. The first one is a novel unsupervised sentence classification model termed hierarchical agglomerative clustering-word mover's distance. The second one is called k nearest neighbour-word mover's distance for supervised sentence categorisation tasks. The proposed k nearest neighbour-word mover's distance is different from Kusner et al. [23] in that this framework incorporates with three bag-of-words construction and two dense word embeddings. Additionally a system efficiency comparison for word mover's distance and relaxed word mover's distance is provided in the experiments section.

5.2.1 Supervised Sentence Categorisation

Figure 5.1 models the stepwise processes of the suggested k nearest neighbour-word mover's distance framework.

The supervised word mover's distance-based k nearest neighbour model construction can be split into the following three sequential steps.

1. Given a target textual snippet, compute the pairwise word mover's distance

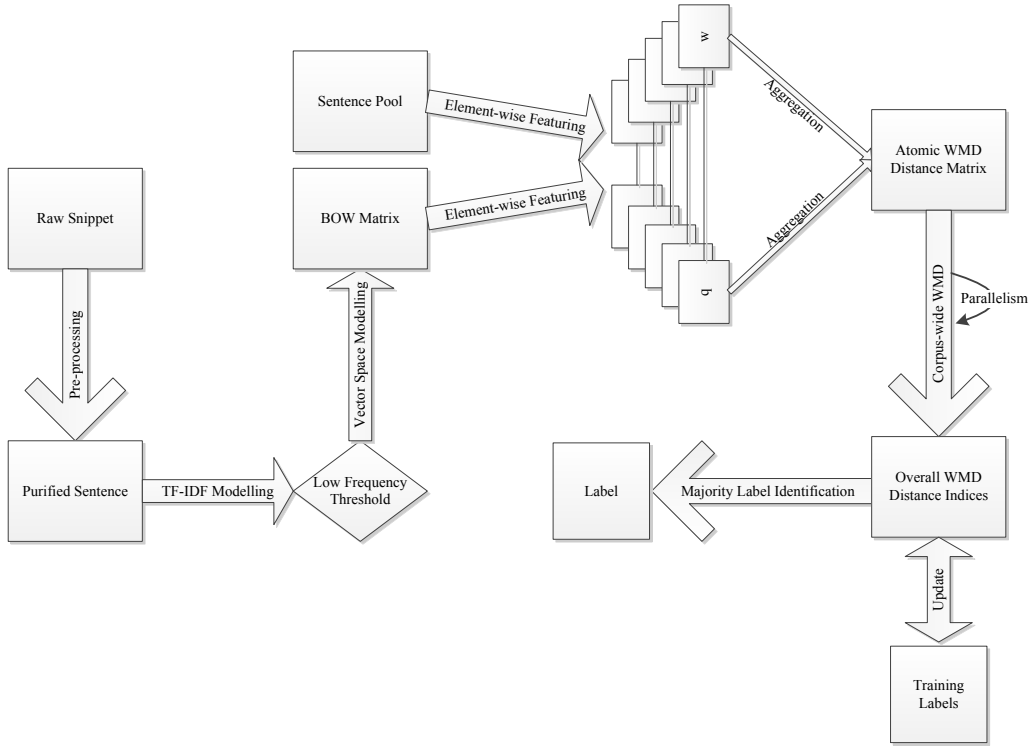

Figure 5.1: A pictorial illustration of k nearest neighbour-word mover's distance design.

distance between the target and each sample in the training set;

2. Identify k nearest neighbours in the training set and extract their corresponding class or label;

3. Count the number of instances for each distinct label and assign the target textual snippet to label with the highest occurrences.

Word mover's distance computation in the supervised sentence clustering is resemblant with that of the unsupervised tasks. Based on such an idea, Equation 5.1, 5.2 or 5.3, and 5.4 also apply to word mover's distance-based k nearest neighbour algorithm.

5.2.2 Hierarchical Agglomerative Clustering with Word Mover's Distance

hierarchical agglomerative clustering-word mover's distance bares some similarities against its traditional version. However, they have distinct distance updating rule which leads to different labelling.

The graphical model of the novel framework is illustrated in Figure 5.2. This pictorial explanation demonstrates the overall process of how each single sentence is pre-processed and stripped from the stop-words, fetched into bag-of-words matrix, element-wise featuring, parallelised word mover's distance distance computation, dendrogram updating scheme, and eventually target cluster identification phase.

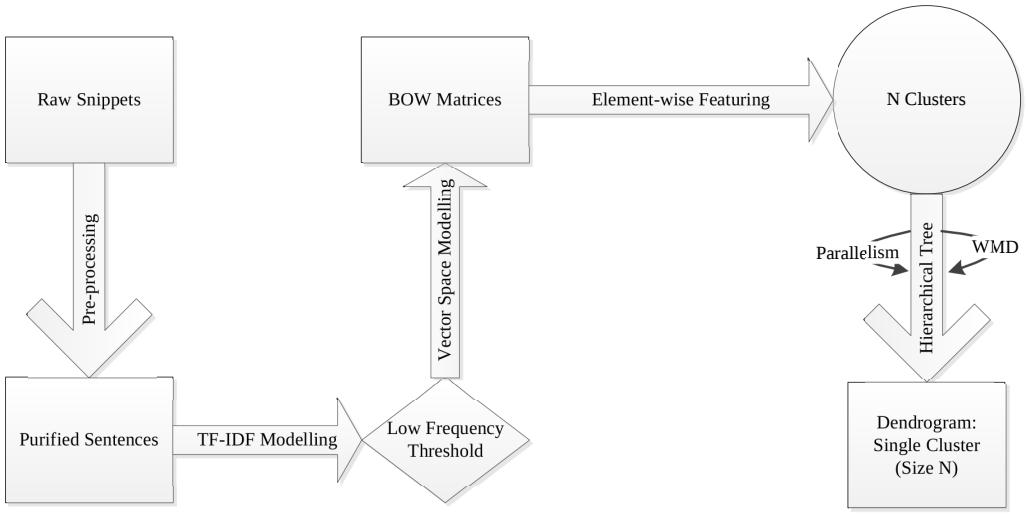

Figure 5.2: A pictorial illustration of hierarchical agglomerative clustering-word mover's distance design.

The proposed model works in an iterative fashion with five steps, which is classical for cluster analysis.

(1). Start by assigning each item to its own cluster, so that if you have N items, you now have N clusters, each containing just one item. Let the distances (similarities) between the clusters equal the distances (similarities) between the items they contain.

(2). Find the closest (most similar) pair of clusters and merge them into a single cluster, so that now you have one less cluster.

(3). Compute word mover's distance distances (dissimilarities) between the new cluster and each of the old clusters in the dendrogram. Since word mover's distance is a computational exhaustive procedure, the current work implements this step in a multi-threading fashion.

(4). Repeat steps (2). and (3). until all items are eventually clustered into a single cluster of size N.

The actual word mover's distance operation follows the four consecutive phases.

The smallest calculation unit w within the sentence categorisation framework is an element-wise operation which computes the unit value between word i and j.

$$w_{ij} = f(\vec{v_i}, \vec{v_j}) \tag{5.1}$$

where $\vec{v_i}$ and $\vec{v_j}$ denote the vectorised word2vec or global vectors for word representation representation of the term i and j. The function f operates in such a way that spatial distance of the input parameter pair is computed.

In the second phase, a proper wrapper of the atomic unit w is constructed to efficiently capture the word mover's distance similarity of a pivot row $\vec{r}$ in the testing samples against all training textual snippets τ.

$$wr_{i\tau} = \{f(\vec{v_i}, \vec{v_k}), \forall k \in \tau\} \tag{5.2}$$

or alternatively as

$$wr_{i\tau} = \{w_{ik}, \forall k \in \tau\} \tag{5.3}$$

In the above case, wr is a function which outputs a set of word mover's distance values and each instance of τ is indicated as k. It is assumed that $\vec{r}$ and every training sample in τ are normalised term frequency-inverse document frequency vectors summing up to unit 1, e.g., $\sum \vec{r} = 1$.

Third, a pairwise word mover's distance operation pw is defined to conduct corpus-wide distance measuring. The corpus-wide operation is analogous to the aforementioned second phase wrapper based assessment.

The system needs to deal with the high computational complexity caused by exhaustive distance comparison. A possible solution to relax the high computational overhead is through parallelism or multi-tasking rather than sequential indexing.

$$pw_c = \{m(wr_s), \forall s \in c\} \tag{5.4}$$

where c denotes the input feature vectors as a whole and function m is a multi-threading worker. The symbol s is a sample in the corpus c. The algorithm takes advantages of the efficient pw functionality to cope with the testing samples and training samples matching. The overall word mover's distance mapping of the whole corpora is readily generated because of the synchronised distance computation.

As previously mentioned in 5.1, classical k nearest neighbour and hierarchical agglomerative clustering fit traditional distance metrics. Performance limit on the sentence categorisation tasks is likely to occur due to the dissimilarity calculation and interpretation on semantic related information. Original linguistic or syntactical features is less likely to be preserved during sentence distance comparison. In the proposed work, the presence of the advanced word embeddings empowers effective decoding and expression of linguistic information with associated numerical word vectors and bag-of-words. Sponsored by the novel distance metrics word mover's distance, performance on the sentence categorisation evaluation can be further enhanced.

5.3 Performance Comparison on Count-based Sentence Categorisation

The supervised part of the sentence categorisation evaluation is measured on the standard precision rate, recall ratio, and F1 score. For unsupervised sentence cat-

egorisation metrics, accuracy score, completeness score, homogeneity score, and V-measure score are presented. Statistical metric like the Mann-Whitney-U is employed to validate the result with twenty repetitive runs of each algorithm. In other words, Mann-Whitney-U-based verification requires 20 runs of a specific algorithm with identical parameters. A p-value is calculated to proof the statistical significance of one algorithm against the other comparison models.

Referring to Table 4.2, term frequency-inverse document frequency, count vectoriser, and binary one-hot vectorisation are three distinct vector space model representation. There are three categories of evaluation for the sentence representation comparison: processing speed, vector space model storage efficiency, and the t-distributed stochastic neighbour embedding encoding for sentence distance visualisation.

5.3.1 TF-IDF, Count Vectoriser, and Binary One-hot Vectoriser

Below reports the vector space model processing time for the participated binary one-hot vectorisation, the traditional count vectoriser, and term frequency-inverse document frequency. The current chapter also presents the changing input corpus size's effect on the vectorisation processing speed. Table 5.1 shows the processing speed difference on the three vectorisation approaches. Figures are measured in milliseconds with two decimal precision. The evaluation was tested on the raw twenty-news-groups data, i.e., before stop-words removal. Column 'Size' denotes the number of sentences in the vector space model and 'Count' represents the generalised one-hot vectorisation as opposed to the binary one-hot version.

As can be seen from Table 5.1, binary one-hot vectoriser requires least computational time when generating a vector space model matrix, followed by count frequency and term frequency-inverse document frequency vectorisation. This is because detecting appearance/absence of a word is much faster than counting the

Table 5.1: Vector space model pre-processing speed comparison in milliseconds. TF-IDF denotes term frequency-inverse document frequency.

Size	**Binary one-hot**	**Count**	**TF-IDF**
1,000	276.31	322.24	327.76
3,000	808.69	862.76	881.27
6,000	1393.36	1604.72	1658.02
9,000	2155.49	2379.65	2448.04
11,000	2620.61	2883.87	2970.83

occurrences of a word in a sentence or corpus and word counting is much easier than computing term frequency and global inverse document frequency.

For changing input size's effect on the vector space model processing speed, binary one-hot, term frequency-inverse document frequency and count vectoriser all follow linear increasing pattern. Figure 5.3 reveals that the trend line for binary one-hot vectorisation can be captured as a linear ascending slope with respect to the input size increases. Likewise, bag-of-words and term frequency-inverse document frequency vectorisation obtain linear pattern with respect to its input corpus size. The hypothesis 'Given same quantity of sentences for vector space model matrix construction, binary one-hot vectorisation obtains the best time efficiency, followed by count vectorisation and term frequency-inverse document frequency.' is verified to be yes.

For vector space model distribution analysis, since term frequency-inverse document frequency vectorisation stores floating number in each entry, its space complexity is higher than its competitors count and binary one-hot vectoriser which maintain an integer value for each entry.

Sentence encoding efficacy for t-distributed stochastic neighbour embedding is effective when a sentence distortion function is employed [205]. It is necessary to consider sentence representation mechanism and distance function together as a joint

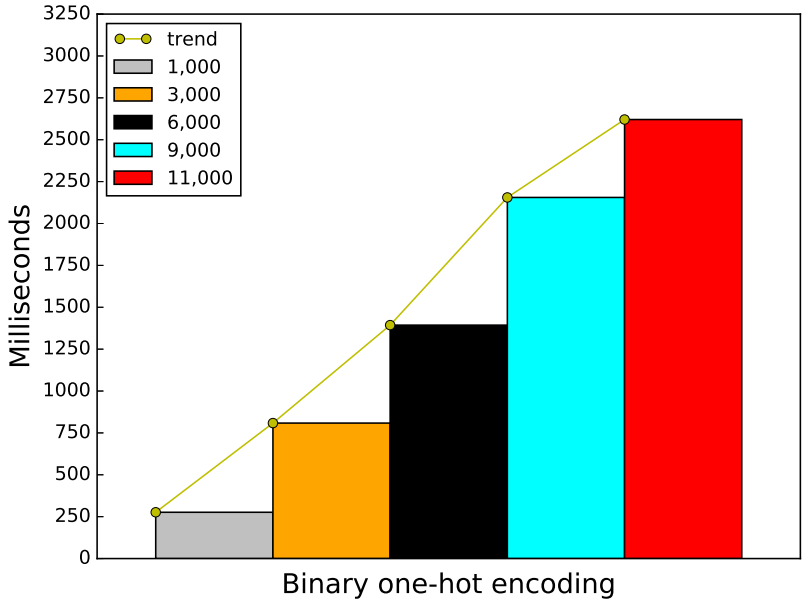

Figure 5.3: Binary one-hot vectoriser runtime statistics.

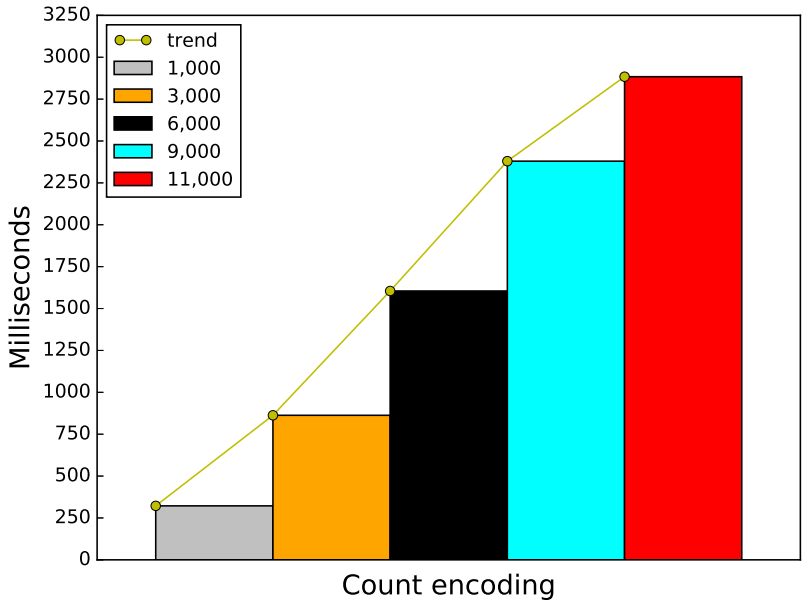

Figure 5.4: Count vectoriser runtime statistics.

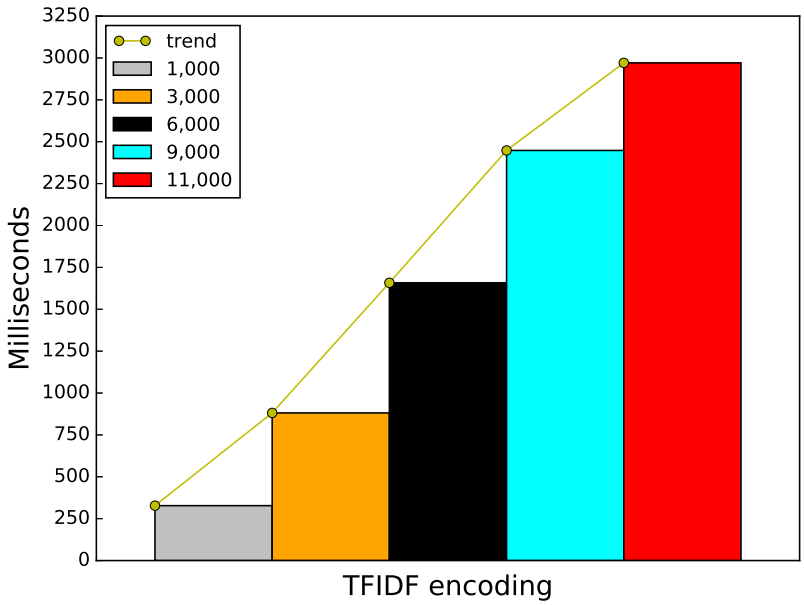

Figure 5.5: Term frequency-inverse document frequency vectoriser runtime statistics.

impact on the t-distributed stochastic neighbour embedding visualisation [206]. T-distributed stochastic neighbour embedding demonstrates a quantised distance metrics efficacy on sentence pool in such a way that better distortion function will have more dense point cloud where each rendered point in the defined hyperplain represents a sentence.[207]. More formally, sentences having resemblant semantic meaning tends to reserve closer spatial distance on the t-distributed stochastic neighbour embedding mapping. Thus t-distributed stochastic neighbour embedding visualises the distortion functions performance in an intuitive way.

5.3.2 Sentence Distortion Functions

Earth mover's distance is a distance function normally applied in image classification scenarios. This technique recently has been transferred to text classification field and termed word mover's distance for sentence similarity calculation [23, 24]. Relaxed word mover's distance [23] is a relaxed version of word mover's distance

where an approximated distance is computed based on reduced constraint. Since those aforementioned distortion functions rely on pre-defined word vector such as global vectors for word representation and word2vec embeddings, there are circumstances where a group of words not belongs to the vocabulary. Under this condition, an infinity variable is assigned to the distance entry.

5.3.3 Word Embeddings

Statistical natural language processing empowers the sentence vectors with word embeddings representation. In this thesis, a group of two options are available: global vectors for word representation[70] and word2vec [67]. During data preparation phase, vector space model matrix maintains a collection of statistical appearance of words in n-gram mode. Each token receives a corresponding word embeddings and a sentence vector is then preserved. Since stop-words removal effectively excludes unnecessary tokens which has no matching word embeddings, it appears rarely to encounter any out-of-vocabulary outliers in practice.

5.4 Word Mover's Distance Classification Results and Analysis

As can be observed from Table 5.2, there are eighteen different k nearest neighbour configurations applicable to the supervised sentence classification. They are configured through vector space model, word embeddings, and distance function. There are two benchmarking data-sets for the supervised environment, namely twenty-news-groups collection and question classification corpus.

5.4.1 Sentence Classification on Twenty-news-groups

The experiment for each configuration of k nearest neighbour utilised a fixed k value for neighbours and the value is set to be five. An observation from Figure

Table 5.2: The list of eighteen variations of the proposed k nearest neighbour model for supervised sentence categorisation. The numbering process conforms with the indices appear at the y-axis of in Figure 5.6, 5.7, 5.8, 5.9, 5.10, and 5.11. The numbering system follows the pattern si where s denotes supervised algorithm and i depicts the index of that specific participant. TFIDF stands for term frequency-inverse document frequency and glove denotes global vectors for word representation respectively.

Numbering	**VSM**	**Embeddings**	**Distance**
s1	TFIDF	word2vec	word mover's distance
s2			relaxed word mover's distance
s3			earth mover's distance
s4	Count	word2vec	word mover's distance
s5			relaxed word mover's distance
s6			earth mover's distance
s7	Binary	word2vec	word mover's distance
s8			relaxed word mover's distance
s9			earth mover's distance
s10	TFIDF	glove	word mover's distance
s11			relaxed word mover's distance
s12			earth mover's distance
s13	Count	glove	word mover's distance
s14			relaxed word mover's distance
s15			earth mover's distance
s16	Binary	glove	word mover's distance
s17			relaxed word mover's distance
s18			earth mover's distance

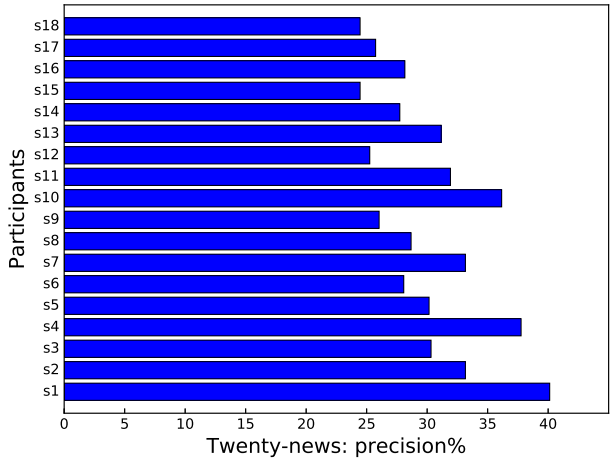

Figure 5.6: Precision ratio distribution for the eighteen configurations of k nearest neighbour-word mover's distance model. Testing data-set in this experiment is twenty-news-groups. Each precision rate is measured as an average of twenty cross-validated runs.

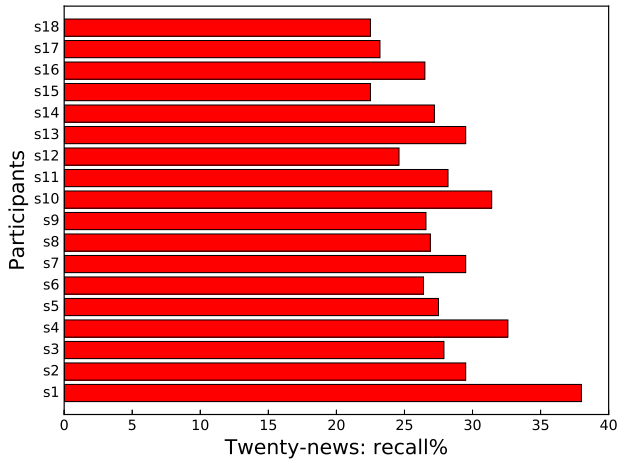

Figure 5.7: Recall rate distribution for the eighteen configurations of k nearest neighbour-word mover's distance model. Testing data-set in this experiment is twenty-news-groups. Every reported recall ratio is measured as an average of twenty cross-validated runs.

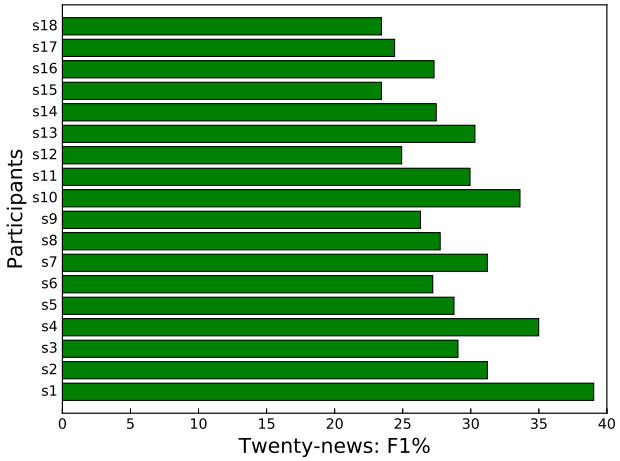

Figure 5.8: F1 score distribution for the eighteen configurations of k nearest neighbour-word mover's distance model. Testing data-sets is twenty-news-groups. Each F1 score entry is an average of twenty cross-validated runs.

5.6 is that configuration s1, s2, and s3 achieve best precision score. This is because term frequency-inverse document frequency vectorisation offers more global semantic information and balanced term frequency than the other two vectorisers. Another reason is that word2vec has a larger vocabulary than its competitor global vectors for word representation. Interestingly, earth mover's distance generally achieves worst precision across each category of comparison. This present work offered in the thesis attributes this to its primary use case on non text mining field. In other words, word mover's distance is designed for word sense distance which makes it a suitable candidate for text categorisation tasks. Relaxed word mover's distance is a relaxed version of word mover's distance and preserves less constraints when forming the words distance. It is expected that relaxed word mover's distance obtains less accuracy than its advanced sibling word mover's distance. The overall performance ranking is therefore word mover's distance, relaxed word mover's distance, and earth mover's distance in descending preference.

Likewise, for recall rate, compelling performance is maintained for word mover's distance. Since F1 score is the harmonic mean of precision and recall rate, and thus

the performance evaluation trend still follows the trend observed from the precision and recall ranking.

5.4.2 Sentence Classification on Question Classification Collection

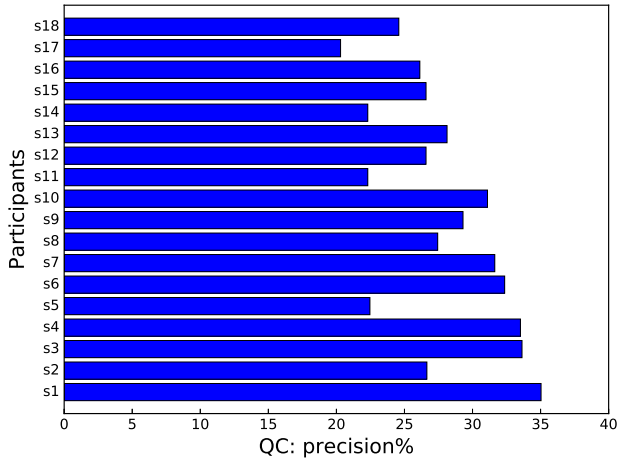

Figure 5.9: Precision ratio distribution for the eighteen configurations of k nearest neighbour-word mover's distance model. Testing data-set in this experiment is question classification collection. Each precision value is measured as an average of twenty cross-validated runs.

Performance on the question classification collection obtains a similar global ranking except that relaxed word mover's distance seems yields lower score on all three evaluation criteria. The experiment for each configuration of k nearest neighbour utilised a fixed k value for neighbours and the value is five. See the performance ranking for question classification in Figure 5.9, 5.10, and 5.11 for details. Algorithm s1, s2, and s3 obtain globally optimum performance. Relaxed word mover's distance, however, performs worse than traditional earth mover's distance. This is because question classification collection is a sparse stream with respect to relatively short sentence length and sentence length plays an important role in the relaxed word

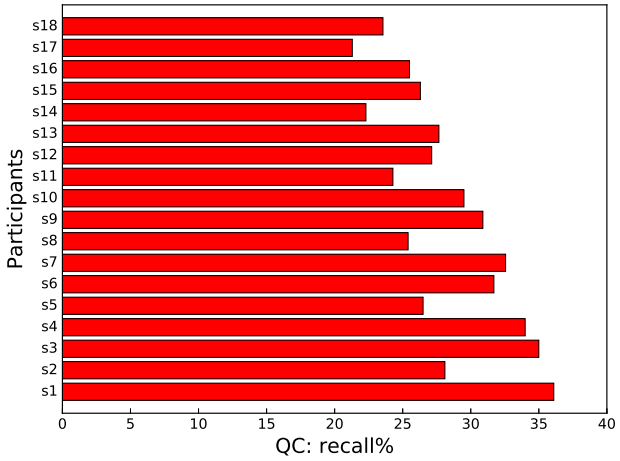

Figure 5.10: Recall rate distribution for the eighteen configurations of k nearest neighbour-word mover's distance model. Testing data-set in this experiment is question classification collection. Every recall entry rate is an average of twenty cross-validated runs.

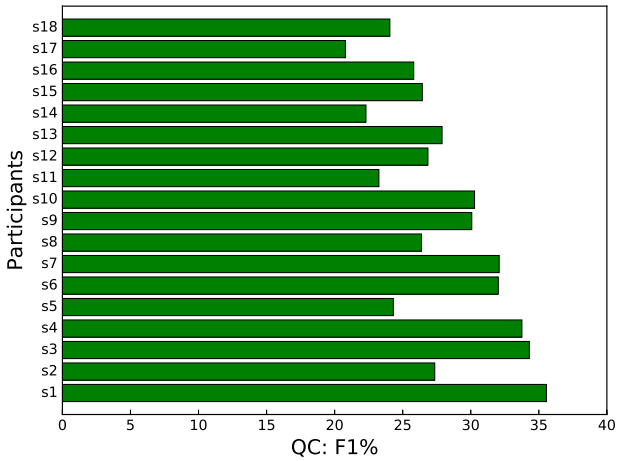

Figure 5.11: F1 score distribution for the eighteen configurations of k nearest neighbour-word mover's distance model. Testing data-sets is question classification stream. Each F1 score is measured as an average of twenty cross-validated runs.

mover's distance.

From the results distribution on two benchmark data-sets for the three vectorisation approaches, it can be observed that term frequency-inverse document frequency vectoriser-based model achieves the best sentence classification performance, followed by count vectorisation and binary one-hot vectoriser in the descending order. This implies that the answer to the hypothesis 'term frequency-inverse document frequency vectoriser provides better vector space model representation than that of count vectoriser and binary one-hot vectorisation, leading to better sentence categorisation performance' is yes.

Since word2vec-based k nearest neighbour models obtains better classification result than that of global vectors for word representation-based ones given other conditions identical in the context of two benchmark data-sets, the answer to the hypothesis 'Given same benchmark data stream, word2vec dense embeddings offers better word vector representation than global vectors for word representation distributed embeddings scheme.' is yes. Likewise, the answer to the hypothesis 'Word mover's distance provides better sentence classification performance than earth mover's distance and relaxed word mover's distance over standard machine learning metrics on the specified supervised benchmark data-sets.' is yes because for each group of comparison for example candidate tuple (s1, s2, s3), word mover's distance yields the optimum evaluation performance over relaxed word mover's distance and earth mover's distance.

5.4.3 Sentence Classification for Comparison Frameworks

Performance evaluation reflecting on Table 5.3 is shown in Figure 5.12, 5.13, and 5.14 for twenty-news-groups data-set and 5.15, 5.16, and 5.17 for question classification corpus. Each participant is compared to the proposed k nearest neighbour-word mover's distance framework and validated by Mann-Whitney-U statistical significance module. The experiment for each configuration of k nearest neighbour utilised

Table 5.3: The list of comparison algorithms for supervised sentence categorisation. The numbering system follows the pattern ci where c denotes comparison algorithm and i depicts the index of that specific algorithm. The term KNN stands for k nearest neighbour classifier and the suffix NB denotes Naive Bayesian

Numbering	**Algorithm**
c1	KNN-cosine
c2	KNN-Euclidean
c3	KNN-Manhattan
c4	Gaussain-NB[167]
c5	Multinomial-NB[162]
c6	Bernoulli-NB[160]
c7	SVM[37]
c11	KNN-word mover's distance

a fixed k value for neighbours and the value is set to be five.

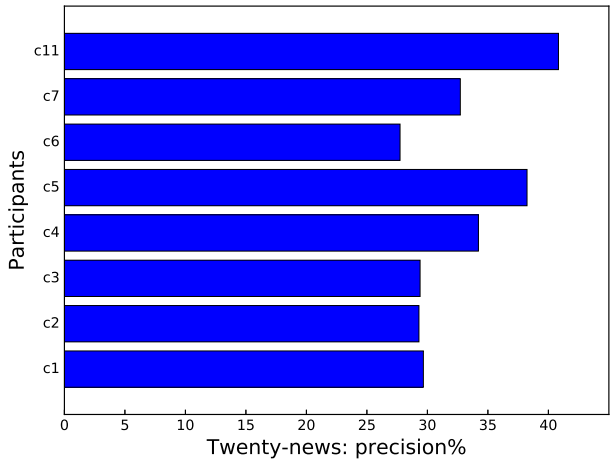

Figure 5.12: Precision rate distribution for every comparison algorithm and k nearest neighbour-word mover's distance model. Testing data-sets is twenty-news-groups data. Each precision score is measured as an average of twenty cross-validated runs.

From Figure 5.12, it can be observed that k nearest neighbour-word mover's distance obtains the highest precision rate on the testing twenty-news-groups stream. Among all the competitors, multinomial-Naive Bayesian achieves the next best precision ratio, followed by Gaussian-Naive Bayesian and SVM classifier. The worst performance is achieved by Bernoulli-Naive Bayesian. This is due to its enforced binary encoding of terms or sentences which harms the internal sentence semantics. For the k nearest neighbour family {word mover's distance, cosine, Euclidean, Manhattan}, word mover's distance achieves the optimal accuracy. The next optimum one is cosine, with Manhattan slightly lower than cosine, and Euclidean being the worst distance metrics.

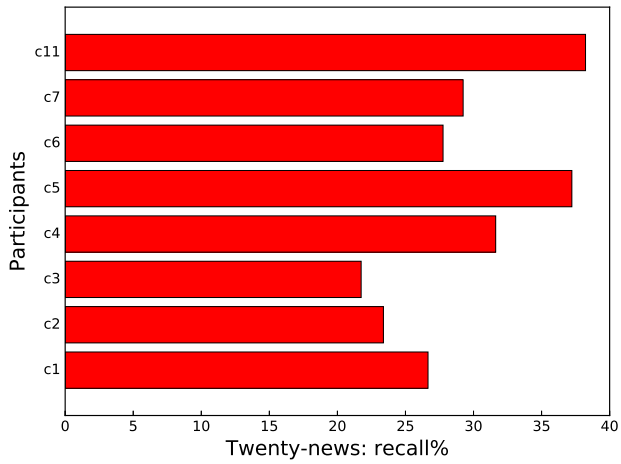

Figure 5.13: Recall ratio distribution for each comparison algorithm and k nearest neighbour-word mover's distance model. Testing data-sets is twenty-news-groups stream. Each F1 score is measured as an average of twenty cross-validated runs.

For recall ratio statistics on the list of comparison algorithms and k nearest neighbour-word mover's distance against the twenty-news-groups stream, refer to Figure 5.13. k nearest neighbour-word mover's distance still ranks the first place under sentence recall evaluation. Multinomial-Naive Bayesian is the next best due to its suitability on the bag-of-words configuration. Gaussian-Naive Bayesian is ranked third in this situation, followed by the SVM classifier, Bernoulli-Naive Bayesian, k

nearest neighbour-cosine, k nearest neighbour-Euclidean, and k nearest neighbour-Manhattan in descending order. Interestingly, cosine-based k nearest neighbour yields better result than its two other variants k nearest neighbour-Euclidean and k nearest neighbour-Manhattan. The current research attributes it to the fine grain computation of distance in cosine distortion. Euclidean distance uses only the magnitude of two sentence vectors and Manhattan distortion measures straight distance in two orthogonal axes. For the k nearest neighbour family {word mover's distance, cosine, Euclidean, Manhattan}, word mover's distance achieves the optimal accuracy, the next best is cosine, followed by Euclidean and then Manhattan.

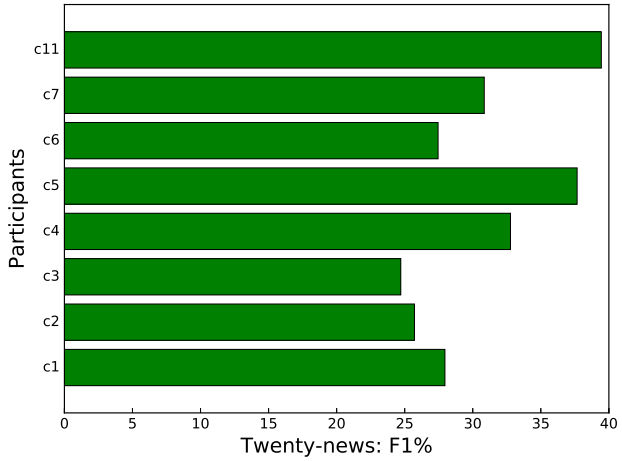

Figure 5.14: F1 score distribution for each comparison framework and k nearest neighbour-word mover's distance model. Testing data-sets is twenty-news-groups. Each F1 score is measured as an average of twenty cross-validated runs.

For the F1 score comparison on the twenty-news-groups data, readers refer to Figure 5.14 for details. In general, the performance statistics bares much similarity with that of precision rate and recall ratio ranking. K nearest neighbour-word mover's distance ranks the first place, Multinomial-Naive Bayesian being the second position, and Gaussian-Naive Bayesian the third place. The worst performance is obtained by k nearest neighbour-Manhattan. For the k nearest neighbour family {word mover's distance, cosine, Euclidean, Manhattan}, word mover's distance

achieves the optimum accuracy.

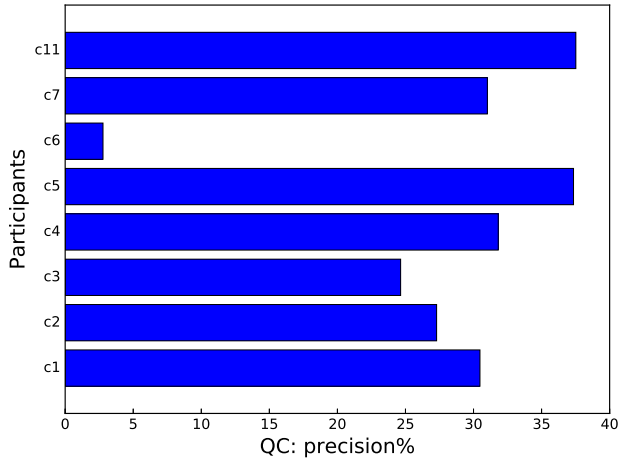

Figure 5.15: Precision rate distribution for every comparison algorithm and k nearest neighbour-word mover's distance model. Testing data-sets is question classification data. Each precision score is measured as an average of twenty cross-validated runs.

From Figure 5.15, it can be perceived that k nearest neighbour-word mover's distance obtains the highest precision rate on the testing question classification collection. Among all the competitors, multinomial-Naive Bayesian achieves the next best precision ratio, followed by Gaussian-Naive Bayesian and SVM classifier. The worst performance is achieved by Bernoulli-Naive Bayesian. For the k nearest neighbour family {word mover's distance, cosine, Euclidean, Manhattan}, word mover's distance achieves the optimal accuracy. The next optimum one is cosine, with Euclidean slightly lower than cosine, and Manhattan being the worst distance metrics. For Naive Bayesian series evaluation comparison, the performance difference is in the following decreasing order: Multinomial-Naive Bayesian, Gaussian-Naive Bayesian, and Bernoulli-Naive Bayesian. The lowest Naive Bayesian classifier is Bernoulli-Naive Bayesian because it enforces binary encoding of terms and sentences which amplifies noise when comprehending the internal sentence semantics.

For recall ratio statistics on the list of comparison algorithms and k nearest neighbour-word mover's distance against the question classification collection, refer

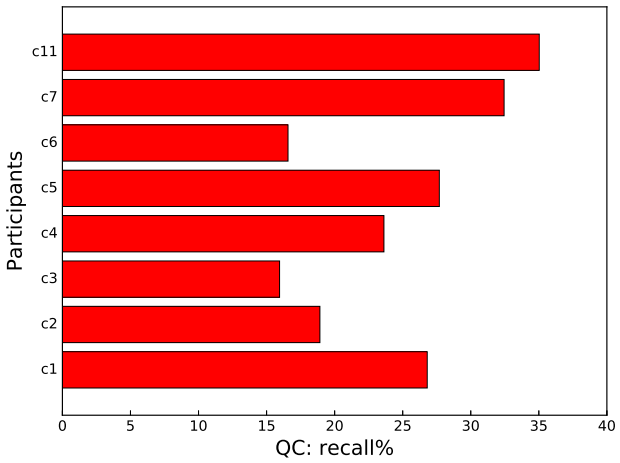

Figure 5.16: Recall ratio distribution for each comparison algorithm and k nearest neighbour-word mover's distance model. Testing data-sets is question classification collection. Each F1 score is measured as an average of twenty cross-validated runs.

to Figure 5.16. k nearest neighbour-word mover's distance ranks the first place under sentence recall evaluation. SVM is the next best, multinomial-Naive Bayesian ranks the third place. The lowest performance is acquired by k nearest neighbour-Manhattan. Interestingly, cosine-based k nearest neighbour yields better result than its two other variants k nearest neighbour-Euclidean and k nearest neighbour-Manhattan. The present work in this thesis attributes it to the fine grain computation of distance in cosine distortion. Euclidean distance uses only the magnitude of two sentence vectors and Manhattan distortion measures summed straight distance in two orthogonal axes. Naive Bayesian series subsumes multinomial, gaussian, and bernoulli. Multinomial-Naive Bayesian has the best recall ratio, followed by gaussian-Naive Bayesian and bernoulli-Naive Bayesian in descending order.

For the F1 score comparison on the question classification data, readers can refer to Figure 5.17. In general, the performance statistics bares much similarity with which of precision rate and recall ratio ranking. K nearest neighbour-word mover's distance still ranks the first place, Multinomial-Naive Bayesian being the second position, and SVM being the third place. The worst performance is obtained by

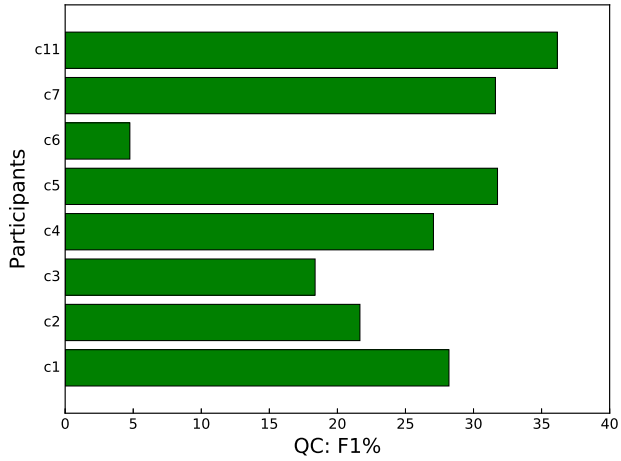

Figure 5.17: F1 score distribution for each comparison framework and k nearest neighbour-word mover's distance model. Testing data-sets is question classification collection. Each F1 score is measured as an average of twenty cross-validated runs.

Bernoulli-Naive Bayesian. For the k nearest neighbour family {word mover's distance, cosine, Euclidean, Manhattan}, word mover's distance achieves the optimum accuracy. Multinomial-Naive Bayesian yields the optimal F1 score across the -Naive Bayesian family.

5.4.4 Mann-Whitney-U Test on Supervised Learning

Mann-Whitney-U test takes two samples and computes a p-value to statistically verify the F1 score significance on a specific testing stream. From Table 5.4, it can be perceived that by comparing k nearest neighbour-word mover's distance to other competing algorithms, k nearest neighbour-word mover's distance obtains statistically significant result at specific significance level. Interestingly, p-value for each pair is much smaller than the specified significance level, indicating that at significance level .01, the result is significant. Additionally, for each comparison pair, 2-tailed hypothesis satisfies and the result is significant at the specified significance level. Overall, k nearest neighbour-word mover's distance generates robust perfor-

Table 5.4: Mann-Whitney-U test p-value for F1 score against the list of competing algorithm and k nearest neighbour-word mover's distance. The experiment data-set is twenty-news-groups.

Algorithm	**c11**	**Significance Level**	**1 or 2-tailed Hypothesis**
c1	<.000001	.01	2
c2	<.000001	.01	2
c3	<.000001	.01	2
c4	<.000001	.01	2
c5	<.000001	.01	2
c6	<.000001	.01	2
c7	.001314	.01	2

mance on the twenty-news-group data. Based on the observation on the Mann-Whitney-U statistical test from Table 5.4, it can be verified that the hypothesis 'Word mover's distance-based distance function improves the accuracy of sentence similarity measurement over cosine, Euclidean, and Manhattan on the specified benchmark textual streams.' is yes on twenty-news-groups. In addition, the answer to the hypothesis 'Given same testing sequences, k nearest neighbour-word mover's distance yields better performance than the other comparison frameworks with respect to precision rate, recall ratio, and F1 score.' is yes under the twenty-news-groups stream.

From Table 5.5, it can be perceived that by comparing k nearest neighbour-word mover's distance to other competing algorithms, it obtains statistically significant result at specific significance level. Interestingly, as algorithm c5 (Multinomial-Naive Bayesian) is the second best, the p-value for (c11,c5) pair indicates that at significance level .05, p-value .02926 < .05, therefore the result is significant at $p < .05$ and not $p < .01$. Likewise, for other comparison pairs, 2-tailed hypothesis satisfies and the result is significant at the specified significance level. The hypothesis holds

Table 5.5: Mann-Whitney-U test p-value for F1 score against the list of competing algorithm and KNN-WMD. The experiment data-set is question classification collection.

Algorithm	**c11**	**Significance Level**	**1 or 2-tailed Hypothesis**
c1	<.000001	.01	2
c2	<.000001	.01	2
c3	<.000001	.01	2
c4	<.000001	.01	2
c5	.02926	.05	2
c6	<.000001	.01	2
c7	.00132	.01	2

for optimum performance on the k nearest neighbour-word mover's distance against question classification collection. Similarly, from Table 5.5 it can be verified that the hypothesis 'Word mover's distance-based distance function improves the accuracy of sentence similarity measurement over cosine, Euclidean, and Manhattan on the specified benchmark textual streams.' is yes on question classification collection. In addition, the answer to the hypothesis 'Given same testing sequences, k nearest neighbour-word mover's distance yields better performance than the other comparison frameworks with respect to precision rate, recall ratio, and F1 score.' is yes under the question classification collection.

5.4.5 Runtime Efficiency for Comparison Algorithms and Distortion Functions

Based on Table 5.6, the most efficient and effective algorithm against the twenty-news-groups data is multinomial-Naive Bayesian and bernoulli-Naive Bayesian. K nearest neighbour-word mover's distance requires the maximum runtime or computational overhead. Similarly, for the question classification collection, resemblant

Table 5.6: System average runtime efficiency for the list of competing algorithm and k nearest neighbour-word mover's distance. Runtime is in seconds.

Algorithm	**Twenty-news Runtime**	**QC Runtime**
c1	0.28	0.38
c2	0.35	0.31
c3	1.02	2.57
c4	0.02	0.10
c5	0.01	0.01
c6	0.01	0.02
c7	0.08	0.11
c11	174.41	215.21

phenomenon can be observed. Multinomial-Naive Bayesian achieves the optimal system elapsed efficiency, followed by bernoulli-Naive Bayesian and gaussian-Naive Bayesian with a short lag. K nearest neighbour-word mover's distance demands a several magnitude more computational time to complete the operation in the question classification collection because of exhaustive sentence pair evaluation.

Table 5.7: Word mover's distance,relaxed word mover's distance, and earth mover's distance distortion function runtime comparison. Runtime unit is in seconds. For classification runtime comparison, twenty-news-groups data is involved, the distance computation runs on a set of 1000 textual snippets. RWMD stands for relaxed word mover's distance.

Distance Function	**Classification**
word mover's distance	249.52
relaxed word mover's distance	205.80
earth mover's distance	221.51

Word similarity based on word mover's distance distortion metrics is a computa-

tionally expensive operation which requires long execution time. This current work provides a runtime comparison on word mover's distance, relaxed word mover's distance, and earth mover's distance. This can offer an insight which word distance function is most computationally efficient.

From Table 5.7, it can be observed that with the perspective of classification evaluation, relaxed word mover's distance achieves the best runtime efficiency, earth mover's distance obtains the next best, word mover's distance being the last one with the most exhaustive computational overhead.

5.4.6 T-distributed Stochastic Neighbour Embedding for Distance Functions

Using t-distributed stochastic neighbour embedding [205, 206] as inspiration, the current work gives a visualised distance function comparison on the 200 testing sentences of twenty-news groups data upon the four involved distance metrics: word mover's distance, cosine, Euclidean, and Manhattan. For consistency, this work employs the identical initialisation of t-distributed stochastic neighbour embedding embedding, i.e., dimension of the embedded space: 2; perplexity: 30; gradient calculation algorithm (i.e. Barnes-Hut-sne [207]) and the balancing angle: 0.5 as the trade-off between accuracy and speed. Generally speaking, a dense instances distribution indicates the superiority of a specific sentence distortion function.

One observation from Figure 5.18 is that the whole cluster is dense and instances of each label are close to each other. The visualised t-distributed stochastic neighbour embedding graph demonstrates the precision word mover's distance can offer to the sentence similarity calculation.

For cosine-based t-distributed stochastic neighbour embedding visualisation, see Figure 5.19. It can be perceived that the core distribution of 200 sentences is smaller than that of word mover's distance function. The number of outliers is also greater than the competitor word mover's distance.

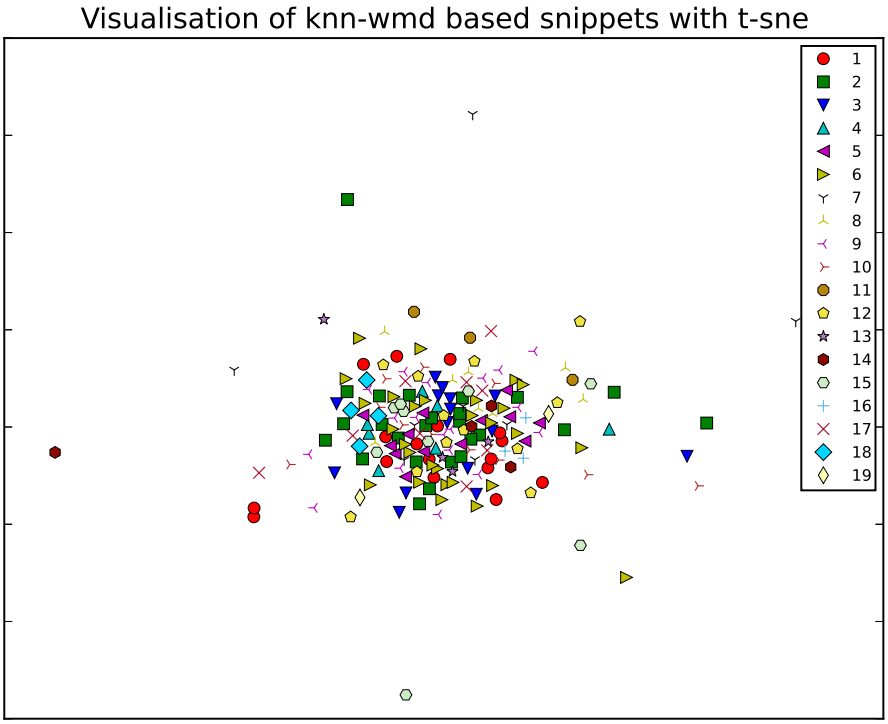

Figure 5.18: K nearest neighbour-word mover's distance distance distribution translated into two-dimensional space via the latest t-distributed stochastic neighbour embedding [205, 206]. The present work applies scikit-learn package for delivering t-distributed stochastic neighbour embedding in the visualisation. The 200 testing sentences represent 19 news groups or labels. Each marker shape represents 1 out of 19 different class labels in the twenty-news groups.

Similarly, for euclidean-based distance function from Figure 5.20and Manhattan-based distortion function from Figure 5.21, the cluster centroid and outliers statistics bare much resemblance with the cosine-based t-distributed stochastic neighbour embedding distribution. Thus it can be argued that word mover's distance-based distance function has advantages over the euclidean and Manhattan sentence dissimilarity function.

Intuitively, the cluster density of a specific class in Figure 5.18 is higher than

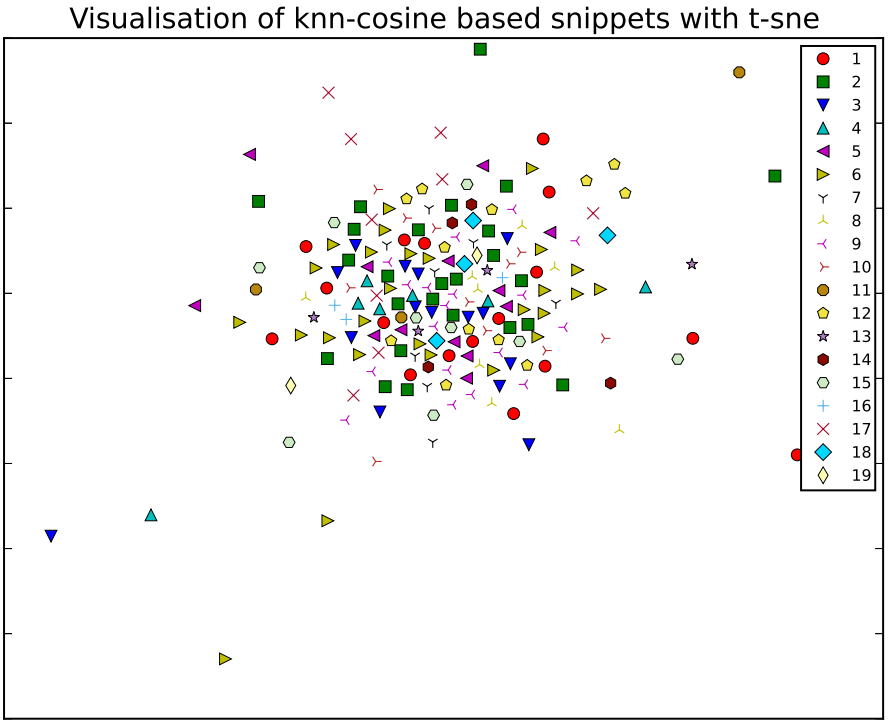

Figure 5.19: K nearest neighbour-cosine distance distribution translated into two-dimensional space via the latest t-distributed stochastic neighbour embedding [205, 206]. The current thesis applies scikit-learn package for delivering t-distributed stochastic neighbour embedding in the visualisation. The 200 testing sentences represent 19 news groups or labels. Each distinct marker represents 1 out of 19 different class labels in the twenty-news groups.

that of Figure 5.19. For instance, for class 1, the within-class distance distribution for word mover's distance better contrasts with the between-class distance distribution when considering the scaled factor. In other words, the within-class distance distribution for cosine function is likely to be skewed given that each instance of the class 1 seems far from each other.

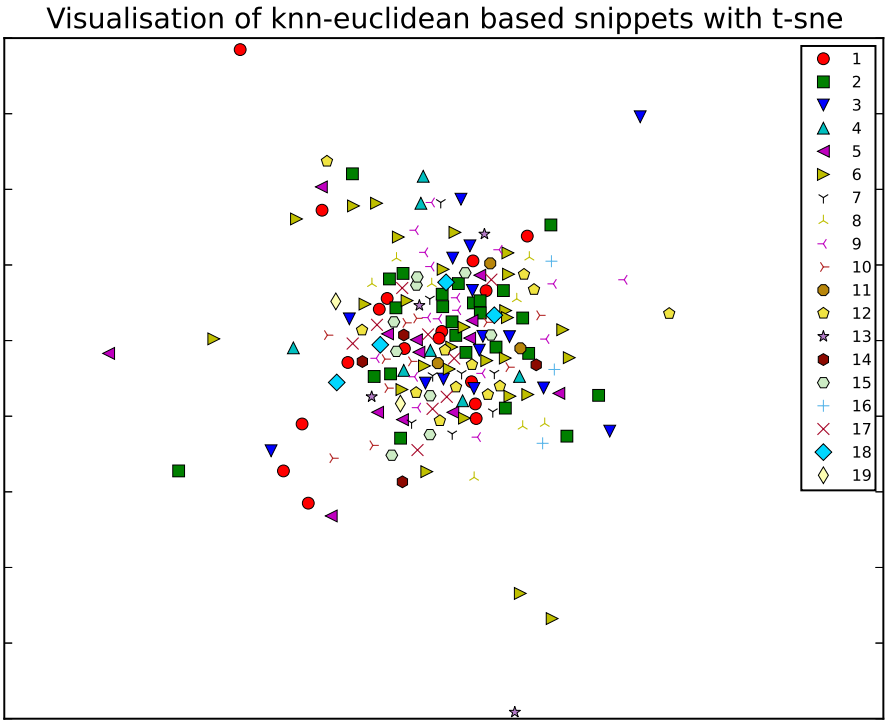

Figure 5.20: K nearest neighbour-Euclidean distance distribution translated into two-dimensional space via the latest t-distributed stochastic neighbour embedding [205, 206]. The experiments presented in this thesis applies scikit-learn package for delivering t-distributed stochastic neighbour embedding in the visualisation. The 200 testing sentences represent 19 news groups or labels. Each distinct marker represents 1 out of 19 different class labels in the twenty-news groups.

5.5 Word Mover's Distance Clustering Results and Analysis

For hierarchical agglomerative clustering-based sentence clustering performance assessment, four classical evaluation metrics are applied, namely mean squared error, completeness score, homogeneity score, and v-measure value.

To explain the twelve variations of the proposed hierarchical agglomerative clus-

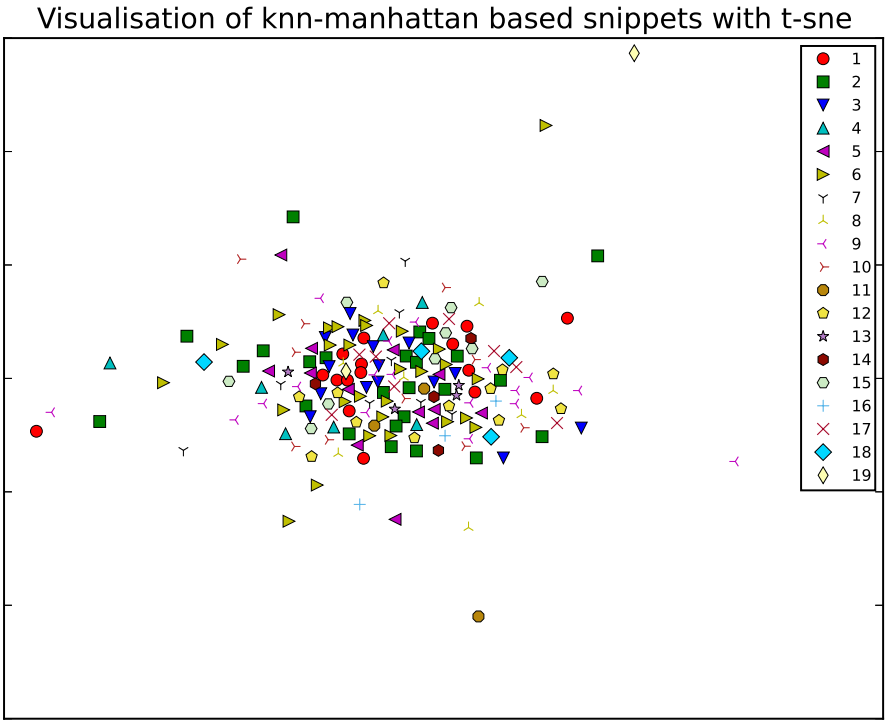

Figure 5.21: K nearest neighbour-Manhattan distance distribution translated into two-dimensional space via the latest t-distributed stochastic neighbour embedding [205, 206]. The thesis applies scikit-learn package for delivering t-distributed stochastic neighbour embedding in the visualisation. The 200 testing sentences represent 19 news groups or labels. Each distinct marker represents 1 out of 19 different class labels in the twenty-news groups.

tering model, three building ingredients are taken into consideration, namely vector space model, embeddings type, and distortion option. The detailed configuration is shown in Table 5.8.

Table 5.8: The list of twelve variations of the proposed hierarchical agglomerative clustering model for unsupervised sentence categorisation. The numbering system follows the pattern ui where u denotes unsupervised algorithm and i depicts the index of that specific participant. The term glove denotes global vectors for word representation.

Numbering	VSM	Embeddings	Distance
u1	Count	word2vec	word mover's distance
u2			relaxed word mover's distance
u3			earth mover's distance
u4	Binary	word2vec	word mover's distance
u5			relaxed word mover's distance
u6			earth mover's distance
u7	Count	glove	word mover's distance
u8			relaxed word mover's distance
u9			earth mover's distance
u10	Binary	glove	word mover's distance
u11			relaxed word mover's distance
u12			earth mover's distance

5.5.1 Sentence Clustering on SMS Spam Stream

From Figure 5.22, one observation is that for the group of u1, u2, and u3, u1 achieves the optimal mean squared error. For other three groups of algorithms, same performance ordering can be perceived. This is because word mover's distance distortion function seems the most accurate distance metrics among word mover's distance, relaxed word mover's distance, and earth mover's distance family. For the global-wise comparison, u1 yields the best mean squared error. This present work attributes this to word2vec, which has larger word embeddings vocabulary and better representation of word vectors compared to its competitor global vectors

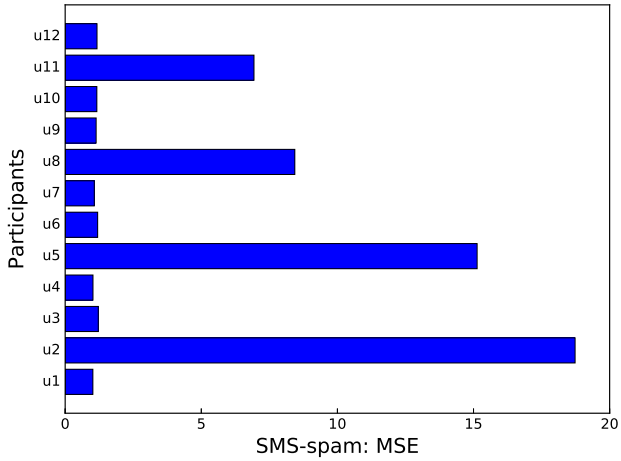

Figure 5.22: Sentence clustering mean squared error evaluation on the twelve different configurations of hierarchical agglomerative clustering-WMD. Low mean squared error value represents low error when evaluating the performance. Testing data-set is sms-spam collection. The numbering system on the y-axis of this figure conforms with Table 5.8. Evaluation is based on average value of clustering experiment with cluster size 10, 20, and 30.

for word representation.

For completeness score, higher value denotes better clustering performance. Overall, u1 achieves the best completeness score, suggesting that count vectoriser, word2vec, and word mover's distance distortion combination works optimally under the sms-spam corpus. The next best performer is u7, followed by u2. The worst model is u3. For sub-group comparison on three distance functions against the tuple (u1, u2, u3), word mover's distance-based framework ranks the first place, the next best is relaxed word mover's distance-based model, and the third place is earth mover's distance-oriented algorithm. This phenomenon applies to other three tuples.

For homogeneity value evaluation, practitioners can refer to Figure 5.24 for details. Algorithm u1 achieves the overall optimal homogeneity score, followed by u7 and u4. This is because count vectoriser better interprets the sentence structure and semantic meaning than that of binary vectorisation and word2vec embeddings

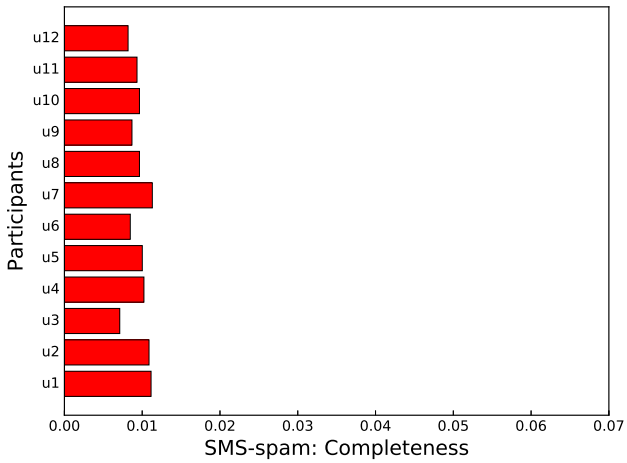

Figure 5.23: Sentence clustering completeness score evaluation on the twelve different configurations of hierarchical agglomerative clustering-WMD. Testing data-set is sms-spam collection. The numbering system on the y-axis of the figure conforms with Table 5.8. Evaluation is based on average value of clustering experiment with cluster size 10, 20, and 30.

has less out-of-vocabulary ratio in practice. The lowest performance is obtained by u5. For each sub-group of comparison, word mover's distance yields the best performance, earth mover's distance ranks the next best, and third place is obtained by relaxed word mover's distance. Other three sub-groups of comparison follow similar pattern.

Since v-measure is the harmonic mean between homogeneity and completeness, therefore the v-measure evaluation bares much resemblance with homogeneity and completeness distribution. As can be seen from Figure 5.25, u1 obtains the best v-measure score, followed by algorithm u7 and u4. The worst performance is achieved by u5. For the sub-group of (u1, u2, u3), word mover's distance yields the best performance, earth mover's distance ranks the next best, and the third place is obtained by relaxed word mover's distance. Other three sub-groups of comparison follow the similar pattern.

From the above performance evaluation upon the sms-spam corpus, the answer

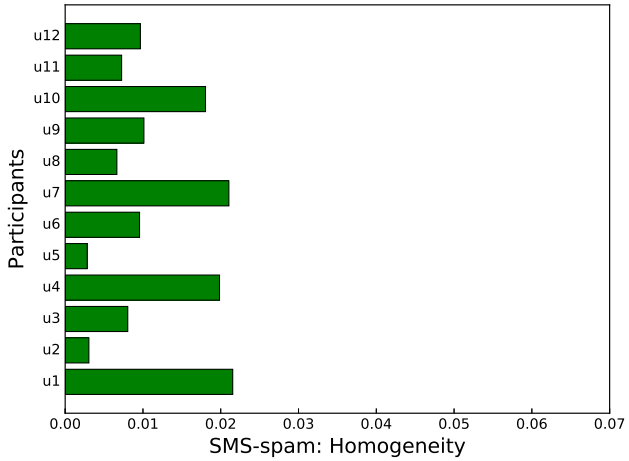

Figure 5.24: Sentence clustering homogeneity score assessment on the twelve different configurations of hierarchical agglomerative clustering-WMD. Testing data-set is sms-spam collection. The numbering system on the y-axis of each figure conforms with Table 5.8. Evaluation is based on average value of clustering experiment with cluster size 10, 20, and 30.

to the hypothesis 'Word mover's distance provides superior sentence clustering performance against earth mover's distance and relaxed word mover's distance over standard machine learning metrics on the specified unsupervised benchmark data-sets.' is yes.

5.5.2 Sentence Clustering on RT Polarity Corpus

From Figure 5.26, one observation is that for the group of u1, u2, and u3, u1 achieves the optimal mean squared error. For other three groups of algorithms, same performance ordering can be perceived. This is because word mover's distance distortion function seems the most accurate distance metrics among word mover's distance, relaxed word mover's distance, and earth mover's distance family. For the global-wise comparison, u1 yields the best mean squared error. The current thesis attributes this to word2vec, which has larger word embeddings vocabulary and

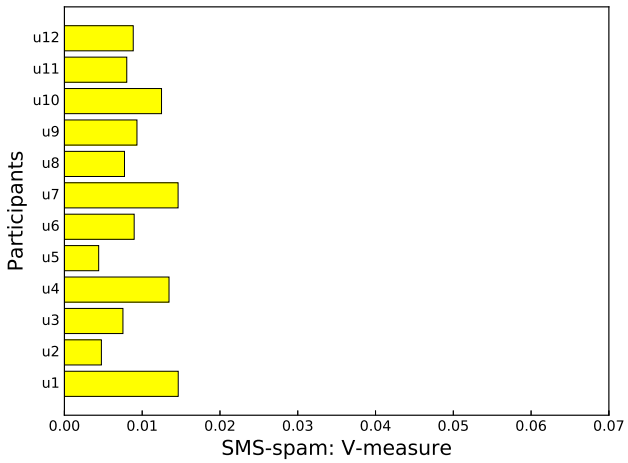

Figure 5.25: Sentence clustering performance evaluation on the twelve different configurations of hierarchical agglomerative clustering-WMD. Testing data-set is sms-spam collection. The numbering system on the y-axis of each figure conforms with Table 5.8. Evaluation is based on average value of clustering experiment with cluster size 10, 20, and 30.

better representation of word vectors compared to its competitor global vectors for word representation. Relaxed word mover's distance is worse than earth mover's distance in this experiment because of its relaxed constraint on the distance computation. Noise is amplified through reduced constraint on the propagated sentence pair comparison [23].

For completeness score from Figure 5.27, higher value denotes better sentence clustering performance. Overall, u1 achieves the best completeness score, indicating that count vectoriser, word2vec, and word mover's distance distortion combination works optimally under the rt-polarity corpus. The next best performer is u4, followed by u7. The worst model is u3. For sub-group comparison on three distance functions against the tuple (u1, u2, u3), word mover's distance-based framework ranks the first place, the next best is relaxed word mover's distance-based model, and the third place is earth mover's distance-oriented algorithm. This phenomenon applies to other three tuples of comparison.

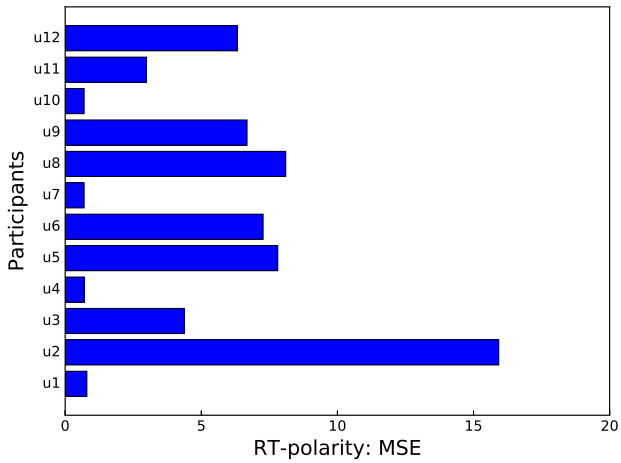

Figure 5.26: Sentence clustering mean squared error evaluation on the twelve different configurations of hierarchical agglomerative clustering-WMD. Testing data-set is rt-polarity. The numbering system on the y-axis of each figure conforms with Table 5.8. Evaluation is based on average value of clustering experiment with cluster size 10, 20, and 30.

For homogeneity value evaluation from Figure 5.28, algorithm u4 achieves the overall optimal homogeneity score, followed by u7 and u10. The lowest performance is obtained by u3. For each sub-group of comparison, word mover's distance yields the best performance, relaxed word mover's distance ranks the next best, and third place is obtained by earth mover's distance. Other three sub-groups of comparison follow same pattern of ranking.

Since v-measure is the harmonic mean between homogeneity and completeness, therefore the v-measure evaluation bares much similarity with homogeneity and completeness distribution. As can be seen from Figure 5.29, u4 obtains the best v-measure score, followed by algorithm u10 and u7. The worst performance is achieved by u3. For the distance function evaluation on tuple (u1, u2, u3), word mover's distance yields the best performance, relaxed word mover's distance ranks the next best, and the third place is obtained by earth mover's distance. Other three sub-groups of comparison follow the similar pattern.

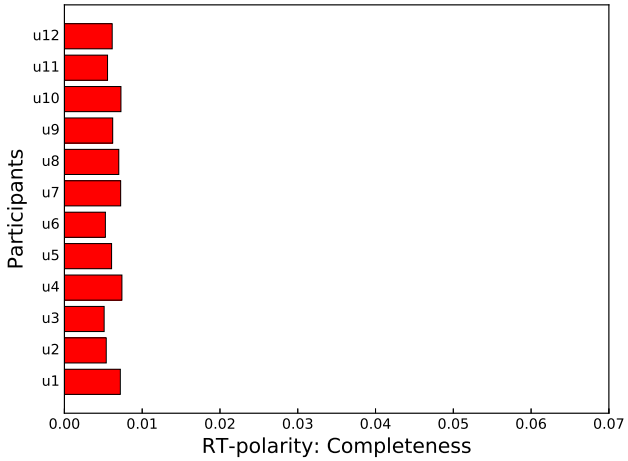

Figure 5.27: Sentence clustering completeness score evaluation on the twelve different configurations of hierarchical agglomerative clustering-WMD. Testing data-set is rt-polarity. The numbering system on the y-axis of each figure conforms with Table 5.8. Evaluation is based on average value of clustering experiment with cluster size 10, 20, and 30.

From the above performance evaluation upon the sms-spam corpus, the answer to the hypothesis 'Word mover's distance provides superior sentence clustering performance against earth mover's distance and relaxed word mover's distance over standard machine learning metrics on the specified unsupervised benchmark data-sets.' is yes on the rt-polarity data-set.

5.5.3 Sentence Clustering for Comparison Algorithms

Table 5.9 shows the list of all 5 unsupervised frameworks for the two benchmarking streams. This list also includes a complete series of distance functions namely {word mover's distance, cosine, euclidean, manhattan}.

Figure 5.30 demonstrates mean squared error evaluation on a list of five clustering algorithms. It can be observed that the proposed hierarchical agglomerative clustering-word mover's distance produces the best mean squared error value. The

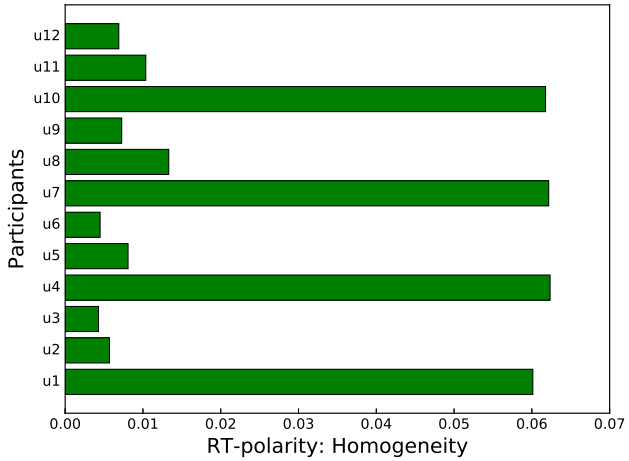

Figure 5.28: Sentence clustering homogeneity score assessment on the twelve different configurations of hierarchical agglomerative clustering-WMD. Testing data-set is rt-polarity. The numbering system on the y-axis of each figure conforms with Table 5.8. Evaluation is based on average value of clustering experiment with cluster size 10, 20, and 30.

Table 5.9: The list of participating algorithms and hierarchical agglomerative clustering-word mover's distance model for unsupervised sentence categorisation. The numbering process in this table follows the pattern uci where uc denotes unsupervised clustering and i depicts the index of that specific algorithm.

Numbering	**Algorithm**
uc1	hierarchical agglomerative clustering-word mover's distance
uc2	hierarchical agglomerative clustering-cosine
uc3	hierarchical agglomerative clustering-Euclidean
uc4	hierarchical agglomerative clustering-Manhattan
uc5	KMeans

next best is obtained by hierarchical agglomerative clustering-cosine, followed by hierarchical agglomerative clustering-manhattan being the third place. The worst

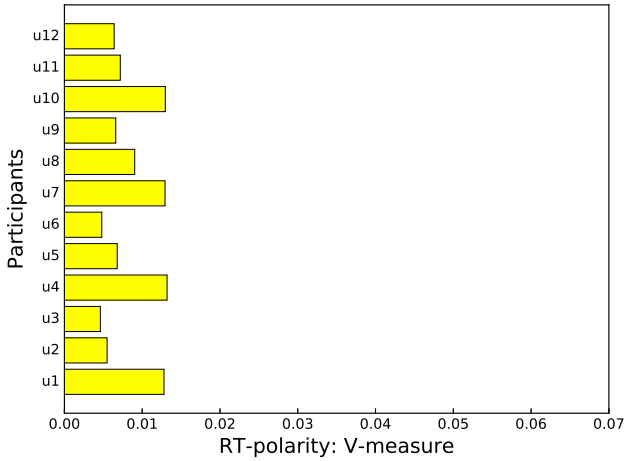

Figure 5.29: Sentence clustering performance evaluation on the twelve different configurations of hierarchical agglomerative clustering-WMD. Testing data-set is rt-polarity. The numbering system on the y-axis of each figure conforms with Table 5.8. Evaluation is based on average value of clustering experiment with cluster size 10, 20, and 30.

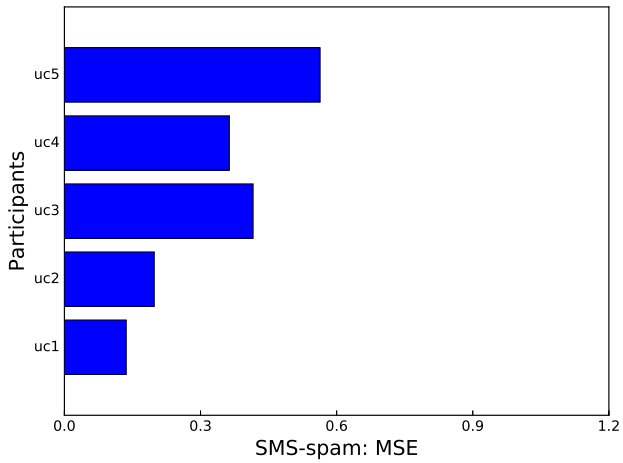

Figure 5.30: MSE for the list of competing algorithms and HAC-WMD. Evaluation is based on average value of clustering experiment with cluster size 10, 20, and 30.

performance is yielded by kmeans algorithm.

Completeness score distribution is shown in Figure 5.31. Algorithm uc1 (HAC-

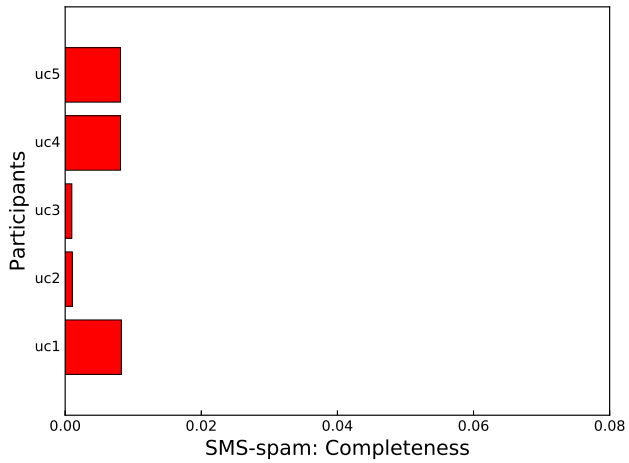

Figure 5.31: Completeness score for the list of competing algorithms and HAC-WMD. Evaluation is based on average value of clustering experiment with cluster size 10, 20, and 30.

WMD) achieves the highest completeness score, meaning the best clustering performance is preserved by uc1. Performance ranking for the rest four frameworks is uc4, uc5, uc2, and uc1 in descending preference.

Homogeneity statistics distribution is demonstrated in Figure 5.32. The overall performance order is in the following decreasing preference uc2, uc5, uc4, uc1, and uc3. For the distortion function family comparison, cosine-based hierarchical agglomerative clustering produces the optimum accuracy than its competitors. The next best distortion function is word mover's distance-based, followed by Manhattan-based framework and the fourth place being the euclidean function.

Figure 5.33 presents v-measure value for the five participating models. Model uc1 achieves the best v-measure score. The next best is uc4, followed by the third place uc5, fourth position uc2, and the last one is uc3. For distance function family evaluation, word mover's distance-based algorithm performs best. The rest three competitors is in the following preference, i.e., Manhattan, cosine, and euclidean.

Mann-Whitney-U test usually takes two samples and computes a p-value to statistically verify the result significance. From Table 5.10, it can be perceived that

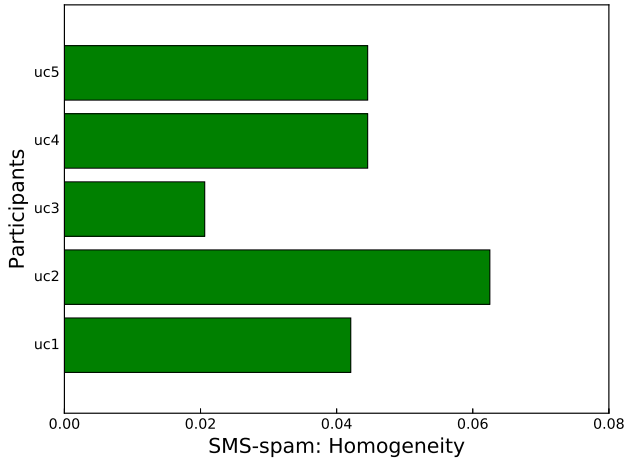

Figure 5.32: Homogeneity value for the list of competing algorithms and HAC-WMD. Evaluation is based on average value of clustering experiment with cluster size 10, 20, and 30.

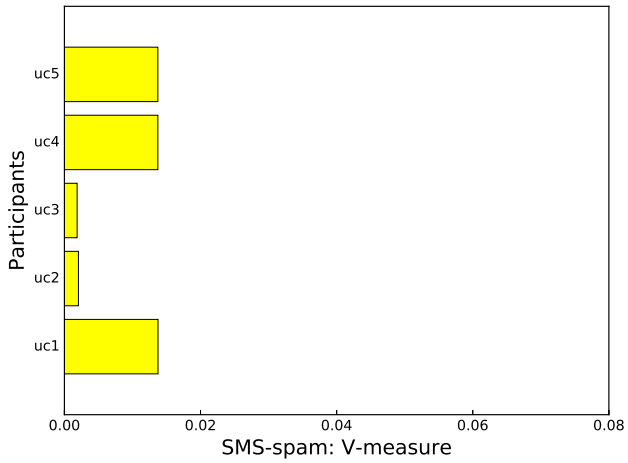

Figure 5.33: V-measure value for the list of competing algorithms and HAC-WMD. Evaluation is based on average value of clustering experiment with cluster size 10, 20, and 30.

by comparing hierarchical agglomerative clustering-word mover's distance to other comparison algorithms, it obtains statistically significant result at $p < .01$ on 2-tailed hypothesis. Thus hierarchical agglomerative clustering-word mover's distance

Table 5.10: Mann-Whitney-U test p-value for validating mean squared error against the list of competing algorithm and hierarchical agglomerative clustering-word mover's distance. The data-set is sms-spam. Evaluation is based on average value of clustering experiment with cluster size 10, 20, and 30.

Algorithm	**uc1**	**Significance Level**	**1 or 2-tailed Hypothesis**
uc2	<.000001	.01	2
uc3	<.000001	.01	2
uc4	<.000001	.01	2
uc5	.02926	.05	2

achieves the statistically best mean squared error on sms-spam collection.

Table 5.11: Mann-Whitney-U test p-value for validating v-measure score against the list of competing algorithm and hierarchical agglomerative clustering-word mover's distance. The data-set is sms-spam.

Algorithm	**uc1**	**Significance Level**	**1 or 2-tailed Hypothesis**
uc2	.00328	.01	2
uc3	<.000001	.01	2
uc4	<.000001	.01	2
uc5	<.000001	.01	2

From Table 5.11, it can be observed that by comparing hierarchical agglomerative clustering-word mover's distance to other competing algorithms on sms-spam, it obtains statistically significant result at the specified .01 significance level. The hypothesis is in 2-tailed. It can be confirmed that hierarchical agglomerative clustering-word mover's distance achieves the statistically best v-measure score on sms-spam collection.

From Table 5.10 and 5.11, it can be perceived that the answer to the following two hypotheses is yes on the sms-spam collection.

(1) Word mover's distance-based distance function improves the accuracy of sentence similarity measurement over cosine, Euclidean, and Manhattan on the specified benchmark textual streams.

(2) Given same testing sequences, Hierarchical agglomerative clustering-word mover's distance achieves better mean squared error value, homogeneity score, completeness value, and v-measure than the competing algorithms in unsupervised learning.

5.5.4 Mann-Whitney-U Test on Unsupervised Tasks

Table 5.12: Mann-Whitney-U test p-value for validating mean squared error against the list of competing algorithm and hierarchical agglomerative clustering-word mover's distance. The data-set is rt-polarity stream.

Algorithm	**uc1**	**Significance Level**	**1 or 2-tailed Hypothesis**
uc2	<.000001	.01	2
uc3	<.000001	.01	2
uc4	<.000001	.01	2
uc5	<.000001	.01	2

From Table 5.12, it can be perceived that by comparing hierarchical agglomerative clustering-word mover's distance to other competing algorithms, it obtains statistically significant result at .01 significance level in each paired comparison. The hypothesis is a 2-tailed one. Therefore it can be confirmed that hierarchical agglomerative clustering-word mover's distance achieves the statistically optimum mean squared error on rt-polarity.

From Table 5.13, it can be observed that by comparing hierarchical agglomerative clustering-word mover's distance to other competing algorithms, it obtains global-wise significant result at specified significance level .01 upon a 2-tailed hypothesis. It

Table 5.13: Mann-Whitney-U test p-value for validating v-measure score against the list of competing algorithm and hierarchical agglomerative clustering-word mover's distance. The data-set is rt-polarity collection.

Algorithm	**uc1**	**Significance Level**	**1 or 2-tailed Hypothesis**
uc2	.00056	.01	2
uc3	<.000001	.01	2
uc4	<.000001	.01	2
uc5	.0001	.01	2

can be confirmed that hierarchical agglomerative clustering-word mover's distance achieves the statistically best v-measure score on the rt-polarity stream.

From Table 5.12 and 5.13, it can be perceived that the answer to the following two hypotheses is yes on the rt-polarity stream.

(1) Word mover's distance-based distance function improves the accuracy of sentence similarity measurement over cosine, Euclidean, and Manhattan on the specified benchmark textual streams.

(2) Given same testing sequences, Hierarchical agglomerative clustering-word mover's distance achieves better mean squared error value, homogeneity score, completeness value, and v-measure than the competing algorithms in unsupervised learning.

5.5.5 Distance Functions Elapsed Time Statistics

From Table 5.14, it can be observed that with the perspective of clustering evaluation, relaxed word mover's distance achieves the best runtime efficiency, earth mover's distance obtains the next best, word mover's distance being the last one with the most exhaustive computational overhead.

Table 5.14: Word mover's distance, relaxed word mover's distance, and earth mover's distance distortion function runtime comparison. Classification and clustering tasks are evaluated separately for integrity and clarity. Here rt-polarity stream is evaluated and the distance computation runs on a set of 1000 textual snippets. Runtime unit is in seconds.

Distance Function	Clustering
word mover's distance	353.47
relaxed word mover's distance	286.61
earth mover's distance	346.86

5.6 Chapter Discussion

This section provides some research findings from the perspective of vector space model vectorisers, sentence distance functions, word embeddings, k nearest neighbour-word mover's distance, and hierarchical agglomerative clustering-word mover's distance respectively.

5.6.1 Findings from Vector Space Modelling Vectoriser

One observation from Table 5.1 is that term frequency-inverse document frequency mechanism occupies most computational resources among the three involved bag-of-words techniques. This is due to its sophisticated inverse document frequency calculation. Another interesting finding is that term frequency-inverse document frequency scheme in collaboration with word2vec dense embeddings and advanced word mover's distance distortion function is able to deliver robust and effective supervised sentence classification performance.

For unsupervised sentence clustering on top of hierarchical agglomerative clustering algorithm, count vectorisation mechanism and binary one-hot vectoriser are participated in the present experiment of research. It is perceived that count vec-

torisation scheme yields better clustering performance in a range of four evaluation metrics, namely mean squared error, completeness score, homogeneity value, and v-measure score. Considering the joint performance evaluation based on the distributed word embeddings and distortion function, count vectoriser still produces stronger performance than that of plain binary one-hot vectorisation. The present work attributes this to the fine grained term frequency mechanism in the count vectoriser.

5.6.2 Findings from Sentence Distance Functions

The research findings for sentence distance functions are in twofold. First, a group of experiments carried out in the present thesis on the tuple word mover's distance, earth mover's distance, and relaxed word mover's distance suggest that word mover's distance provides overall best sentence distance accuracy with most expensive computational overhead. Second, word mover's distance is compared to other strong competitors like cosine distance, Euclidean distortion, Manhattan dissimilarity function on both supervised and unsupervised learning tasks and its competing sentence categorisation accuracy is observed through the corresponding evaluation metrics.

For supervised learning, one research finding is that plain spatial similarity functions like cosine, Euclidean, and Manhattan offer worse performance than word mover's distance. The current research attributes this to the fact that word mover's distance takes word-level distance into account and considers sentence semantic differences between target and query textual snippet. Likewise, for unsupervised learning, statistical Mann-Whitney-U test reveals that word mover's distance-based hierarchical agglomerative clustering model achieves the best v-measure score and mean squared error value.

5.6.3 Findings from Word Embeddings

The experiments presented in this chapter offer two dense word embeddings candidates, word2vec and global vectors for word representation mechanism. In general, word2vec offers better word vector in bag-of-words representation since it has a larger vocabulary or look up dictionary to query when generating a target term matrix. Another reason is that the training pool for generating the dictionary of word2vec and global vectors for word representation is different, which exerts an influence on the resulting word vector, although both candidates have same word vector dimension in 300.

5.6.4 Findings from K Nearest Neighbour-Word Mover's Distance

Extensive supervised experiments results and evaluation on the list of comparison algorithms and k nearest neighbour-word mover's distance show that the novel algorithm implemented in the current thesis obtains overall optimal result on the two benchmark streams. This is because word mover's distance-based sentence distance computation precisely reflects semantic meaning difference between two sentential snippet. T-distributed stochastic neighbour embedding visualisation intuitively shows the spatial distance among testing data on a reduced plain. Another interesting finding is that the MultinomialNaive Bayesian achieves best performance across precision rate, recall ratio, and F1 score among its siblings like GaussianNaive Bayesian and BernoulliNaive Bayesian. The current work attributes this to the multinomially distributed term frequency assumption embedded in the MultinomialNaive Bayesian. In other words, BernoulliNaive Bayesian and GaussianNaive Bayesian are less accurate in terms of the term frequency distribution assumption.

5.6.5 Findings from Hierarchical Agglomerative Clustering-Word Mover's Distance

For hierarchical agglomerative clustering-word mover's distance based sentence clustering performance, one observation is that on the homogeneity score evaluation for sms-spam data-set, the proposed hierarchical agglomerative clustering-word mover's distance algorithm achieves worse than that of hierarchical agglomerative clustering-cosine. This indicates that there is no free lunch in the unsupervised sentence categorisation. There is no single algorithm performing robustly good in every benchmarking collection. For hierarchical agglomerative clustering-oriented performance comparison, word mover's distance function yields generally best results on the four evaluation measurements. This is because word mover's distance function operates in a 'asymmetrical' manner where term frequency statistics in target and query sentence are account. Optimised word pair similarity computation is also performed to dynamically measure spatial distortion between the target and query sentence.

5.7 Contribution to Sentence Categorisation

The major contributions to the domain of sentence categorisation are listed as follows.

- This present work proposes a novel sentence level distance metric for similarity computation under unsupervised learning;
- The proposed two sentence categorisation frameworks incorporates with three distinct sentence encoding schemes;
- The current thesis designs and develop a novel unsupervised sentence categorisation framework hierarchical agglomerative clustering-word mover's distance which delivers superior clustering performance on the benchmarking data streams;

- The work presented in current chapter extends the usual supervised word mover's distance-based k nearest neighbour [23] to its unsupervised version, i.e., word mover's distance-based hierarchical agglomerative clustering algorithm;

- Both supervised and unsupervised word mover's distance-oriented framework work on sentence level textual snippets, where the previous models only worked on the document level tasks;

5.8 Conclusion

This chapter emphasises sentence categorisation from the perspective of statistical term frequency. Word mover's distance is proposed in this work for developing two sentence categorisation models namely k nearest neighbour-word mover's distance and hierarchical agglomerative clustering-word mover's distance for supervised and unsupervised learning respectively. The following research questions and associated hypotheses are addressed in the results and analysis section.

For the research question 'How does the expressiveness of term frequency-inverse document frequency differ from that of count vectoriser and binary one-hot vectorisation?', two hypotheses can be obtained.

(1) Given same quantity of sentences for vector space model matrix construction, binary one-hot vectorisation obtains the best time efficiency, followed by count vectorisation and term frequency-inverse document frequency.

(2) Term frequency-inverse document frequency vectoriser provides better vector space model representation than that of count vectoriser and binary one-hot vectorisation, leading to better sentence categorisation performance.

The answer to the hypothesis 'Given same quantity of sentences for vector space model matrix construction, binary one-hot vectorisation obtains the best time efficiency, followed by count vectorisation and term frequency-inverse document frequency.' is yes because binary one-hot vectoriser consumes least computational time among the competitors. The answer to the hypothesis 'Term frequency-inverse document frequency vectoriser provides better vector space model representation than that of count vectoriser and binary one-hot vectorisation, leading to better sentence categorisation performance.' is also yes since term frequency-inverse document frequency offers statistically stronger sentence categorisation performance against count vectorisation and binary one-hot vectoriser given same testing sentential snippets.

From the research question 'Can word mover's distance-based distance function improve the accuracy of sentence similarity measurement over existing spatial distance functions?', the following hypotheses can be generated:

(1) Given same benchmark data stream, word2vec dense embeddings offers better word vector representation than global vectors for word representation distributed embeddings scheme.

(2) Word mover's distance provides better sentence classification performance than earth mover's distance and relaxed word mover's distance over standard machine learning metrics on the specified supervised benchmark data-sets.

(3) Word mover's distance provides superior sentence clustering performance against earth mover's distance and relaxed word mover's distance over standard machine learning metrics on the specified supervised benchmark data-sets.

(4) Word mover's distance-based distance function improves the accuracy of sentence similarity measurement over cosine, Euclidean, and Man-

hattan on the specified benchmark textual streams.

The answer to each of the hypotheses derived from the research question 'Can word mover's distance-based distance function improve the accuracy of sentence similarity measurement over existing spatial distance functions?' is yes.

The research question 'How does the performance of word mover's distance-based sentence categorisation models differ from that of the other competing algorithms?' implies the following two hypotheses

(1) Given same testing sequences, k nearest neighbour-word mover's distance yields better performance than the other comparison frameworks with respect to precision rate, recall ratio, and F1 score.

(2) Given same testing sequences, Hierarchical agglomerative clustering-word mover's distance achieves better mean squared error value, homogeneity score, completeness value, and v-measure than the competing algorithms in unsupervised learning.

The answer to the above two hypotheses yielded from the question 'How does the performance of word mover's distance-based sentence categorisation models differ from that of the other competing algorithms?' is yes based on the statistical evaluation on the sentence classification/clustering performance for each benchmark data-set. The next chapter shows neural network-based sentence classification including an introduction to classical convolutional neural network and traditional recurrent neural network model, how the proposed convolutional recurrent neural network framework is constructed with ingredients and aggregator, and the actual performance evaluation on a list of comparison algorithms and convolutional recurrent neural network against the four evaluation data-sets.

Chapter 6

Neural Network-based Sentence Classification

This chapter presents a neural network-based sentence classification model called convolutional recurrent neural network. The model composes of three ingredients: character-level convolutional neural network (Char-CNN), character-aware recurrent neural network (Char-RNN), and a modular aggregator. A summary of traditional deep neural networks such as convolutional neural network and recurrent neural network is displayed. Deep neural models have witnessed successful applications on image classification, natural language processing, sentiment analysis [33, 229]. Experimental results and discussions for convolutional recurrent neural network and the other competing frameworks are shown in this chapter as well.

The research questions and corresponding hypotheses this chapter are going to answer and test are listed as follows. Those hypotheses are specifically tested through section 6.7.

For the question 'Can recurrent unit be leveraged to improve the convolutional neural network's performance in the context of sentence classification?', the following hypothesis can be derived.

(1) Convolutional recurrent neural network offers better sentence classifi-

cation performance than that of plain convolutional neural network.

For the question 'How does the performance of long short-term memory differ from that of gated recurrent unit and minimal gated unit?', the subsequent two hypotheses can be yielded.

(1) Long short-term memory, gated recurrent unit, and minimal gated unit offer statistically insignificant sentence categorisation performance when tested on same data-sets.

(2) Minimal gated unit requires lowest computational overhead, followed by gated recurrent unit and long short-term memory when constructing recurrent unit.

For the question 'How does the performance of convolutional recurrent neural network differ from that of the comparison frameworks for sentence classification?', the following hypothesis can be generated.

(1) Given identical hyper-parameters and benchmark data-sets, convolutional recurrent neural network model produces better sentence classification performance than that of the comparison frameworks.

6.1 Preliminary to Convolutional Neural Network

In this section, a presentation of preliminary work related to the model is given. More specifically, two major categories of text-based neural networks are described in details here: word-level neural networks including convolutional neural network and word-level recurrent neural network and character-aware networks such as character-level convolutional neural network and character-aware recurrent neural network.

6.1.1 Convolutional Neural Network

A conventional word-level CNN is usually trained on top of the pre-trained word embeddings such as global vectors for word representation [70] and word2vec [67, 69]. It takes a sentential text s of length n as the input sequence [74, 230, 76]. Let $s_i \in \mathbb{R}^d$ be the d dimensional word embedding vector of the i-th word index mapping in the corresponding sentence. The specific sentence is denoted as

$$s_{1,2,\ldots,n} = s_1 \oplus s_2 \oplus \cdots \oplus s_n, \tag{6.1}$$

where $\oplus$ is the matrix concatenation operator. Often sentence padding is involved to deal with variable length sentences [231].

A convolving layer introduces a filter to operate on a sentence or a list of words [104]. For example, a specific feature is applied to a subset of the sentence pool to extract a raw feature vector. A non-linear activation function is then employed to produce a fine feature vector. A feature map is constructed once the filter has scanned each possible window size of the target sentence. In other words, a feature map is a collection of feature selector. Variable length filters yield useful features for the subsequent max-over-time pooling. Alternatively, a model can define variable length sliding windows to obtain multiple feature maps for the later concatenation operations [30].

An proper activation function like Rectified Linear Unit or tanh is needed to modulate a piece of input information for the output gate [150].

A max-over-time pooling layer is the subsequent operation next to the convolution layer [232]. The maximum element of a given feature map is selected as the value of the current filter. The output from the max pooling is denoted as the most prominent feature. Commonly the max pooling layer is the penultimate structure before the fully connected softmax layer. A softmax layer generates a probability distribution on the classification labels. This completes a classical cycle of a word-level convolutional neural network.

6.1.2 Recurrent Neural Network

A classical recurrent neural network is developed from a conventional feedforward neural network [22]. Recurrent neural network has roots in continuous speech recognition [233]. A typical recurrent neural network consists of an internal recurrent hidden state h_t and an optional variable length output y. More formally, the variable length input sequence x can be defined as:

$$x = (x_1, x_2, ..., x_T). \tag{6.2}$$

A recurrent neural network modifies h_t based on a non-linear activation function ϕ. ϕ can be as simple as an element-wise logistic sigmoid with an affine transformation [22]. It can also be a smooth and bounded hyperbolic tangent function. Sophisticated activation neuron such as long short-term memory, gated recurrent unit, or minimal gated unit cell are also applicable.

A recurrent neural network learns a probability of an arbitrary sequence length to the input recurrent hidden entity as a probability chain function:

$$p(x_1, x_2, ..., x_T) = p(x_1)p(x_2|x_1) \cdots p(x_T|x_1, ..., x_{T-1}). \tag{6.3}$$

The very last element of the sequence probability is a dedicated end-of-sequence value. This thesis can then encodes each component of the probability chain with

$$p(x_t|x_1, ..., x_{t-1}) = \phi(h_t). \tag{6.4}$$

The above computation completes the information flow of a plain recurrent neural network. From this modelled distribution, it is straightforward to encode a new sequence through an iteration over a specific symbol x at each time stamp.

6.2 Ingredients of the Model

Character-level convolutional neural network was tailored from the original model to make it suitable for this work. The character-level convolutional neural network

consists of a convolving layer, Rectified Linear Unit, and max-pooling layer. The classical Character-level convolutional neural network from Vosoughi et al. [16] has multi-layer structure which seems inefficient when trained with the benchmarking data. The present work adapted character-aware recurrent neural network from Cho et al. [21] and Hochreiter and Schmidhuber [22]. Character-aware recurrent neural network model has three different variations including gated recurrent unit [21], minimal gated unit [118], and long short-term memory [22]. This thesis's character-aware recurrent neural network is different from the tradition in that the input gate fits with the intermediate result from character-level convolutional neural network and the output gate follows modulated activation.

6.2.1 Character-Aware Convolutional Neural Network

In the proposed model, C represents the container of character quantisation and d is the character embedding size. The model was developed to support the same 70 characters selected by Vosoughi et al, which are presented below [16]. Suppose that

abcdefghijklmnopqrstuvwxyz0123456789
-,;.!?:'"/\|_#$%ˆ&*∼'+-=<>()[]{}

Figure 6.1: Illustration of 70 characters supported by convolutional recurrent neural network model.

a term $t \in C$ is comprised of a sequence of characters with length k, and then the character-level word representation of t is denoted as $C^k \in \mathbb{R}^d \times k$. Each character is encoded as a binary vector $v \in \{0,1\}^d$. Because the character quantisation is composed of 70 non-space tokens, d is equal to 70 in this case.

Given the above input sequence, the 1-D convolving layer discrete kernel function $f(i) \in [1,m] \to \mathbb{R}$ and the proper padding algorithm p, the output feature map is

$$h^k(i) = RELU(\sum_{i=1}^{m} C^k f(i \times tk + o)) \tag{6.5}$$

where $o = m - k + 1$ is a free constant, m depicts the output size, and RELU stands for the Rectified Linear Unit [152].

The temporal max-pooling layer operates in the following manner. Suppose that stride is highlighted by s,

$$y(i) = max_{i=1}^{m} C^k f(i \times s - k) \tag{6.6}$$

Notice that padding scheme is not applied at the max-pooling layer.

6.2.2 Character-Aware Recurrent Neural Network

Compared to the traditional implementation on gated recurrent unit [21], minimal gated unit [118], and long short-term memory [22], this work implanted an aggregator to produce combined result from two ingredients of convolutional recurrent neural network.

The long short-term memory is composed by five parametric arguments: memory cell, input gate, forget gate, output gate, and aggregated output state for each element.

Gated recurrent unit cell is different from the long short-term memory in that the memory content exposure is not presented. Memory information inside the gated recurrent unit cell is fully exposable [116]. Besides, the location of the input gate in the long short-term memory unit and the corresponding reset gate in the gated recurrent unit cell are different.

Gated recurrent unit cell consists of a reset gate, an update gate, and an aggregated output gate. The activation at timestep t of the j-th gated recurrent unit cell is processed as a linear transformation on the previous output and the current output state.

The recent minimal gated unit has shed some insights on the model. Zhou et al.'s minimal gated unit cell has around 33% less training hyper-parameters and has equivalent performance with gated recurrent unit [118]. The proposed minimal gated unit consists of a forget gate and an aggregated hidden recurrent output.

6.3 Designing Convolutional Recurrent Neural Network

In this section, the actual model design and construction is presented. This current chapter shows a graphical view of how the convolutional recurrent neural network is implemented. Three different recurrent units, namely long short-term memory, minimal gated unit, and gated recurrent unit are embedded into the neural network model. Share hyper-parameters for convolutional recurrent neural network and applicable comparison neural frameworks are given as well.

Table 6.1: The list of hyper-parameters with value.

Filters (F)	400
Hidden Size (H)	400
Window Size (Λ)	20
Pooling Window (P)	2
Stride	1
Padding Algorithm	VALID
Learning Rate	0.01
Training Steps	1000
Optimiser	Adam [141]

The model refers to Table 6.1 as the global training parameters configuration. The 'VALID' padding refers to no artificial padding, in contrast with the 'SAME' padding. The initial learning rate is set to 0.01. Moreover, the global training process is optimised through the Adam optimiser [141]. The learning rate decay is defined by Adam optimiser [141].

$$\alpha_t = \mathrm{lr}_{t-1} \times \frac{\sqrt{1-\beta_2^t}}{1-\beta_1^{(t-1)}}, \tag{6.7}$$

where lr_{t-1} is the learning rate for previous training step. β_1 is the exponential decay

rate for the 1st moment estimates and β_2 is for the 2nd moment estimates [141].

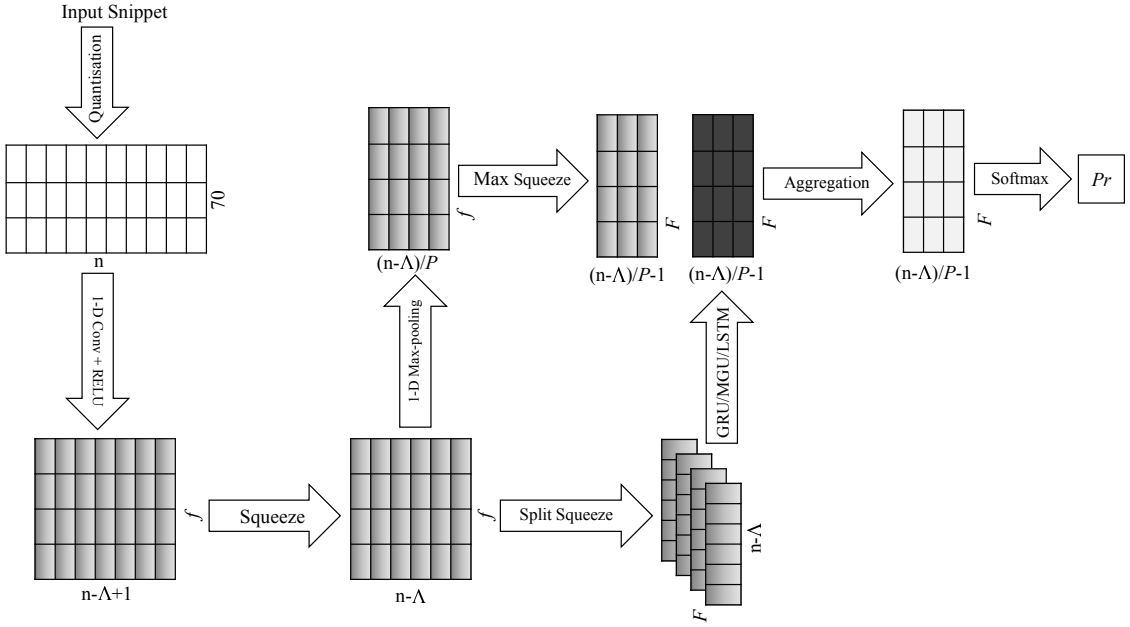

Figure 6.2: Illustration of the convolutional recurrent neural network model design.

In Figure 6.2, n stands for the padded input sequence length. Λ indicates the kernel window size. F represents the number of filters applied at the convolving layer and P denotes the pooling window. The default stride is 1. The operation on the 1-D temporal convolving layer, followed by the Rectified Linear Unit activation operation, leads to the high level features with frame size $n - \Lambda + 1$. A squeeze function is performed to get a reduced matrix formation with filter size $n - \Lambda$. Subsequently, two separate branches of feature computation are conducted. On one hand, the temporal max-pooling layer of character-level convolutional neural network yields an output with feature size $(n - \Lambda)/P$. In the following layer of character-level convolutional neural network, a maximum reduction operation is operated over the sentence dimension to obtain an intermediate matrix with frame size $(n - \Lambda)/P - 1$.

On the other hand, as the gated recurrent unit/minimal gated unit/long short-term memory based character-aware recurrent neural network only recognises input

snippets in a list of sequences, the model thus splits and squeezes the matrix retrieved from the previous layer along with the F dimension to achieve a list of corresponding sequences. A gated recurrent unit/minimal gated unit/long short-term memory unit embedded with H as the hidden size is constructed to acquire a matrix encoding. For the aggregation operation, it takes the output from the gated recurrent unit/minimal gated unit/long short-term memory and the result matrix from the maximum reduction operation upon the character-level convolutional neural network and modulates the two encodings accordingly. Given the probability distribution Pr over the possible labels of choice from the penultimate layer, the softmax will predict the target label. This completes a full training cycle for the proposed architecture.

Towards the final output, the number of frames from the softmax layer adapts to the classification data itself. For instance, if it is a 55-class problem, then the output frame is 55. The experiment offered in the present work evaluated the different conditions pair (i.e., $(0.9, 0.1), (0.8, 0.2) \ldots (0.1, 0.9)$) on the aggregation layer. This thesis found that a large proportion assigned to the character-level convolutional neural network (i.e., 0.7) and a small distribution on the character-aware recurrent neural network (i.e., 0.3) achieves the best performance. Additionally, because drop-out layer does not always appear to be effective in practice, the work therefore did not include any drop-out layers in this regard [75].

So far, this present work has constructed a novel convolutional recurrent neural network model which employs prominent feature-filtering from character-level convolutional neural network and long-term sequence understanding from character-aware recurrent neural network. The framework automatically learns grammatical errors and misspelling through subword information. The developed architecture also benefits from the latest development on gated recurrent unit, which reduces algorithmic runtime without compromise of the performance.

6.4 Comparison Frameworks for Convolutional Recurrent Neural Network

This section specifically describes which algorithms participate in the experiment and what types of evaluation metrics are engaged in it. The section also presents the detailed configuration of each comparison algorithm with its corresponding input sequence encoding. There are in total twenty different algorithms participated in the experiment for convolutional recurrent neural network.

Table 6.2 below shows the competing algorithms involved in the experimental evaluation. More specifically, the first three numberings (i.e., a1, a2, and a3) are the variants of convolutional recurrent neural network. The rest algorithms form the comparison family. Since traditional character-level convolutional neural network uses no explicit encoding except character-level interpretation, the cell for its encoding is marked '–'. Word-aware neural networks utilise word embeddings as the input, therefore popular pre-trained distributed word vectors like word2vec and global vectors for word representation are available for usage. As a baseline comparison, random initialisation of word vectors is implemented. In the random encoding, a Gaussian distribution is employed as a probability seed for distributing word indices. For plain count based competitors such as SVM, k nearest neighbour-word mover's distance, and classical k nearest neighbour, a sentence-based vectorisation technique dubbed sent2vec or skip-thought vector is also provided.

6.5 Experimental Configurations for Convolutional Recurrent Neural Network

This section specifically describes which algorithms participate in the experiment and what types of evaluation metrics are engaged in it. The section also presents the detailed configuration of each comparison algorithm with its corresponding input

sequence encoding. There are in total twenty different algorithms participated in the experiment for convolutional recurrent neural network.

For non encoding options in the experiment, each is given a description on how the input textual snippet is translated according to its codification system.

Word2vec embeddings is a pre-trained word vector database which effectively learns and assigns a numerical array of real-valued numbers to a specific term [71]. Word2vec is trained on a ten billion Google-news stream and the each word embedding is a three hundred dimension vector [71]. Global vectors for word representation is different from word2vec in that a learned vector space representation is captured from a combined solution of global matrix factorisation and local context window technique [70]. Statistical information extracted from a word-word co-occurrence scenario is efficiently applied to the word vector space learning process. Global vectors for word representation employed in this thesis is trained on a six billion tokens corpus which contains a 2014 Wikipedia dump and Gigaword 5 stream [70]. The dimension of a word vector representation in global vectors for word representation is three hundred as well.

Sent2vec or skip-thought vectors is an unsupervised sentence-level encoder-decoder learner which recognises a sentence as a vector. Since sent2vec by its own right uses no knowledge base or external information about sentence semantics, the issue of unseen words has to be solved by a vocabulary expansion mapping. This mapping is resolved by a vocabulary transition from word2vec domain to recurrent neural network word embedding space.

Another worth noting point is the choice of random or sent2vec in the comparison algorithms as shown in Table 6.2. The random encoding only works on word-aware neural networks simply because sent2vec is a sentence-level embeddings mechanism.

Character-level convolutional neural network is a nine layer deep neural network designed by Zhang, Zhao, and LeCun [17]. It interprets the input sequences by character encoding. It composes of six convolving modules and three fully-connected layers. Two dropout layers with probability of 0.5 are also included in between the

three fully-connected layers for dynamic regularisation and overfitting prevention [223].

Character-aware recurrent neural network follows the traditional long short-term memory-based architecture developed by Hochreiter and Schmidhuber [22]. Character-aware recurrent neural network introduces a forget gate to facilitate the long-term memory learning, which is not presented in [22]. Character-aware recurrent neural network learning adopts classical stochastic gradient descent and back propagation.

Word-level convolutional neural network is an implementation from Zhang and Wallace [75]. Their model deploys one convolution layer on the sentence matrix. Temporal 1-max pooling is applied over each map generated by convolving layer. A univariate feature vector is recorded for later softmax layer.

Word-level recurrent neural network is adopted from the original long short-term memory model [22] with the only exception that the input encoding is in word embeddings.

SVM is an classical classifier with radial basis function kernel [198]. It is suitable for data within a couple of 10000 samples [224]. Table 6.3 gives the detailed parameters setting for SVM configuration.

K nearest neighbour-word mover's distance is a variant of the plain k nearest neighbour algorithm [23]. It measures the distance between two distributions by word mover's distance metric [23]. However, in the traditional k nearest neighbour framework, the distance metric is Euclidean.

Evaluation on the sentence categorisation performance is measured by the following criteria. Examination on precision rate, recall ratio, F1 score, and loss value were demonstrated for each of four data-sets. The experiment is conducted on a cross-validated training and testing data split. The participants for the experiment are listed in Table 6.2. There are 20 frameworks in total for the experimental evaluation. The token random indicates the arbitrary initialisation of word vector.

Table 6.4 below lists the major algorithmic comparison and its evaluation in

details.

6.6 Datasets Statistics for Neural Network Frameworks

As the major focus of this work is to develop convolutional recurrent neural network for enhancing the topics classification, the appropriate benchmark data-sets would be natural textual streams or raw sentential data. This thesis utilised four popular classification data streams from the Internet for this study. Each benchmark stream contains only English language-based sentences. They are listed below:

- **Google-news** is a small portion of excerpts from the online platform giant Google [49], specifically the Google news channel.
- **Twenty-news-groups** was originated from the well-known American newspaper publishers formed as twenty-news-groups, probably first appeared at Lang's work [221].
- **Brown Corpus** of Standard American English, often abbreviated as the Brown Corpus [222]. The Brown Corpus encompasses with one million tokens of American English texts sampled from 15 different textual categories. The Brown Corpus is created by Francis and Kucera at Brwon University in 1960s [222].
- **Question Classification** was created by Li and Roth [60]. The question classification collection contains 50 classes of texts. The specific data-set characteristic is shown at Table 4.7 below.

Table 6.5 reports the summary statistics based on the benchmarking data corpus. The work addressed the specificity of data with a six-dimension schema:

1. the number of sentences in the data-set

2. vocabulary of the stream
3. the total number of terms in a stream
4. average sentential snippet length
5. meta-data inclusion
6. the number of labels or classes of the corpus

6.7 Convolutional Recurrent Neural Network Results and Analysis

This section presents the sentence classification results and analysis of them for convolutional recurrent neural network model and the comparison algorithms. The competing algorithms include character aware convolutional neural network, recurrent neural network, word level convolutional neural network, recurrent neural network, and non-neural network models. There are twenty comparison frameworks in total. Experimental benchmark corpus subsume Google-news stream, twenty-news-groups data-set, brown corpus, and question classification collection.

In this work, a novel convolutional recurrent neural network model is constructed which employs prominent feature-filtering from character-level convolutional neural network and long-term sequence understanding from character-aware recurrent neural network. The framework automatically learns grammatical errors and misspelling through subword information. The architecture also benefits from the latest development on gated recurrent unit, which reduces algorithmic runtime without compromise of the performance.

In general, this work shows the results independently for each benchmarking data stream. Evaluation on precision rate, recall ratio, and F1 score are demonstrated for each of data-sets. The experiment is conducted on a cross-validated training and testing data split. The participants for the experiment are listed in Table 6.2. There

are twenty frameworks in total for the experimental evaluation. The token random indicates the arbitrary initialisation of word vectoriser.

6.7.1 Sentence Classification on Google-news Data

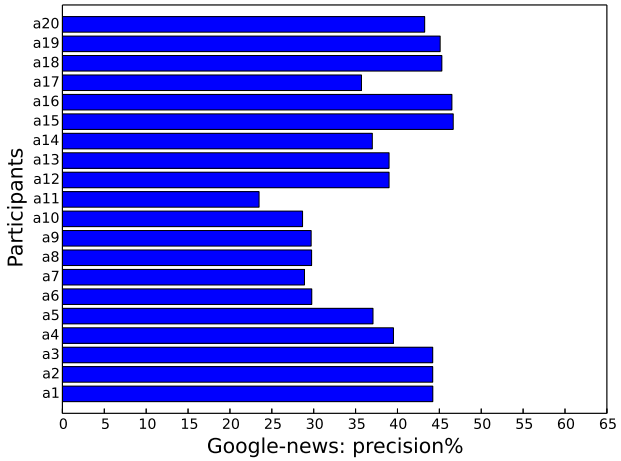

Figure 6.3: Precision rate for the list of competing algorithms and convolutional recurrent neural network on Google-news data-set. Evaluation is based on average value of twenty experiment with same hyper-parameters.

As shown in Figure 6.3, 6.4, and 6.5 (Google-news), convolutional recurrent neural network model ranks fifth on precision rate, with 2.44% less than the best one. For the recall rate and F1 score, the proposed model yields first and third respectively. Convolutional recurrent neural network architecture achieves 2.62% more recall and 0.57% less F1 score compared with the best one. From the above presentation on the Google-news data-set, the answer to the hypothesis "Convolutional recurrent neural network offers better sentence classification performance than that of plain convolutional neural network." is yes.

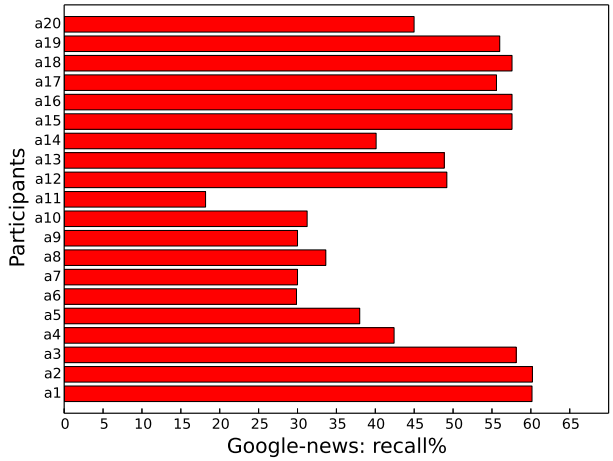

Figure 6.4: Recall rate for the list of competing algorithms and convolutional recurrent neural network on Google-news data-set. Evaluation is based on average value of twenty experiment with same hyper-parameters.

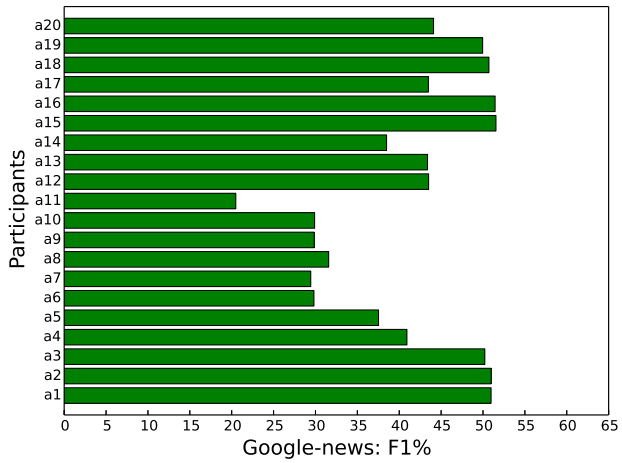

Figure 6.5: F1 score for the list of competing algorithms and convolutional recurrent neural network on Google-news data-set. Evaluation is based on average value of twenty experiment with same hyper-parameters.

6.7.2 Sentence Classification on Twenty-news-groups Corpora

For twenty-news-groups collection, users can refer to Figure 6.6, 6.7, and 6.8. The algorithm achieves the optimal precision rate, recall ratio, and F1 score, leading

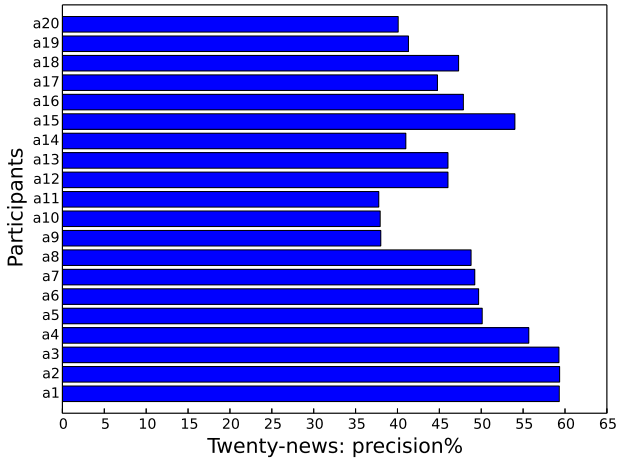

Figure 6.6: Precision rate for the list of competing algorithms and convolutional recurrent neural network on twenty-news-groups data-set. Evaluation is based on average value of twenty experiment with same hyper-parameters.

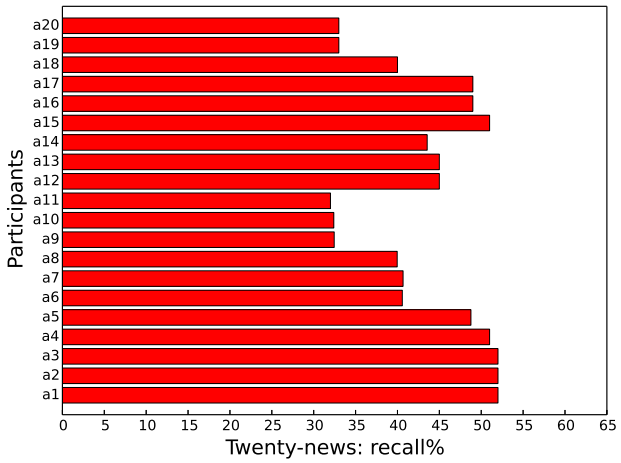

Figure 6.7: Recall rate for the list of competing algorithms and convolutional recurrent neural network on twenty-news-groups data-set. Evaluation is based on average value of twenty experiment with same hyper-parameters.

the next best with 3.68%, 1.00%, and 2.20% respectively. Convolutional recurrent neural network algorithm yields a similar precision rate and recall rate. For the F1 score, the lowest score 34.65% is from word-level recurrent neural network and

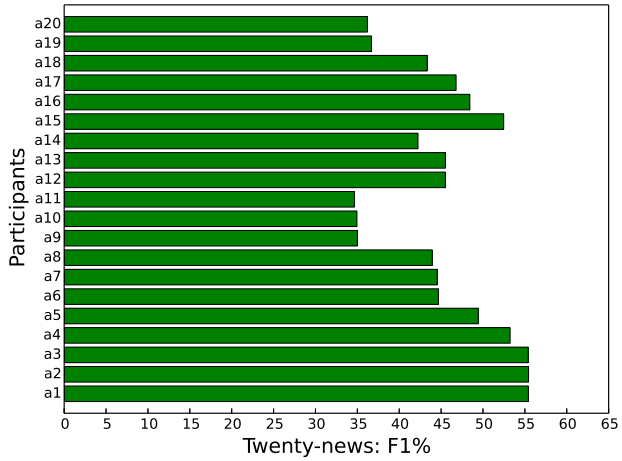

Figure 6.8: F1 score for the list of competing algorithms and convolutional recurrent neural network on twenty-news-groups data-set. Evaluation is based on average value of twenty experiment with same hyper-parameters.

random embeddings combination. The proposed algorithm obtains much better precision rate, recall ratio, and F1 score than word-level recurrent neural networks and word-level convolutional neural networks. Experimental results on twenty-news-groups stream reveals that the answer to the hypothesis 'Convolutional recurrent neural network offers better sentence classification performance than that of plain convolutional neural network.' is yes.

6.7.3 Sentence Classification on Brown Corpus

For Brown Corpus, observations can be derived from Figure 6.9, 6.10, and 6.11. In general, the proposed framework achieves the next best on precision rate, 0.54% less than the character-level convolutional neural network. Convolutional recurrent neural network ranks third on recall rate, having 1.40% difference to the optimum one. The model obtains the best F1 score, with 0.05% more than the next best. From Figure 6.9, 6.10, and 6.11, word level neural networks perform worse than convolutional recurrent neural network model. From the sentence classification per-

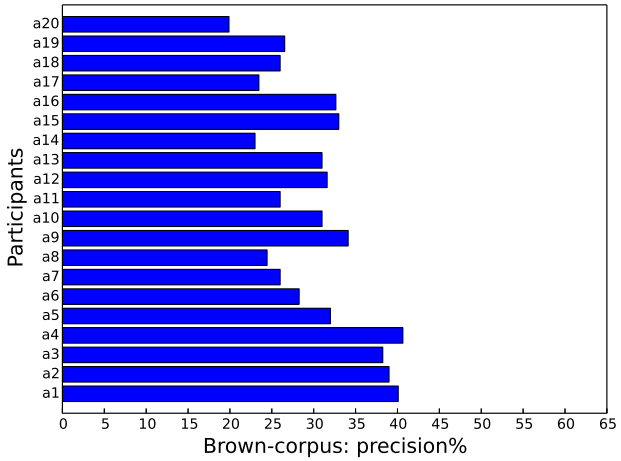

Figure 6.9: Precision rate for the list of competing algorithms and convolutional recurrent neural network on brown-corpus data-set. Evaluation is based on average value of twenty experiment with same hyper-parameters.

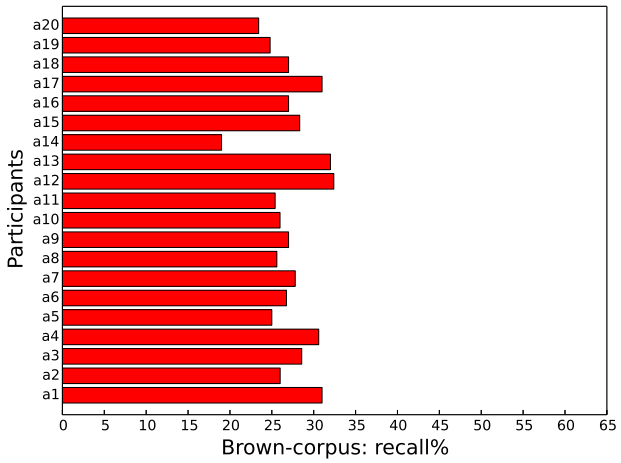

Figure 6.10: Recall rate for the list of competing algorithms and convolutional recurrent neural network on brown-corpus data-set. Evaluation is based on average value of twenty experiment with same hyper-parameters.

formance examination on the brown corpus, the hypothesis 'Convolutional recurrent neural network offers better sentence classification performance than that of plain convolutional neural network.' is tested to be yes.

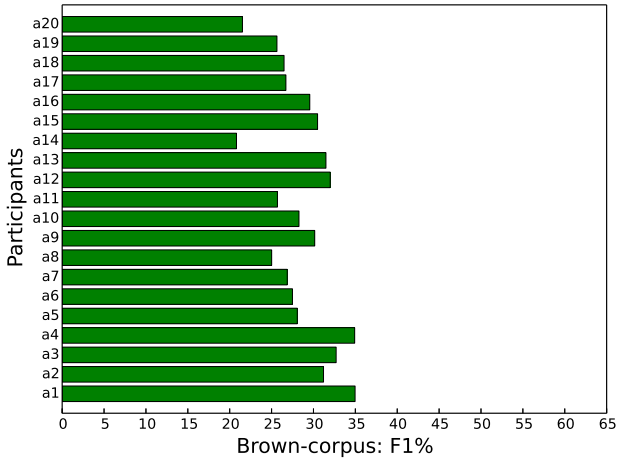

Figure 6.11: F1 score for the list of competing algorithms and convolutional recurrent neural network on brown-corpus data-set. Evaluation is based on average value of twenty experiment with same hyper-parameters.

6.7.4 Sentence Classification upon Question Classification Collection

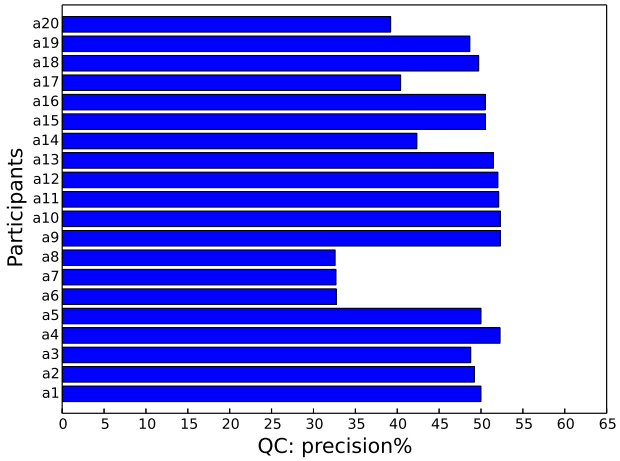

Figure 6.12: Precision rate for the list of competing algorithms and convolutional recurrent neural network on question classification collection. Evaluation is based on average value of twenty experiment with same hyper-parameters.

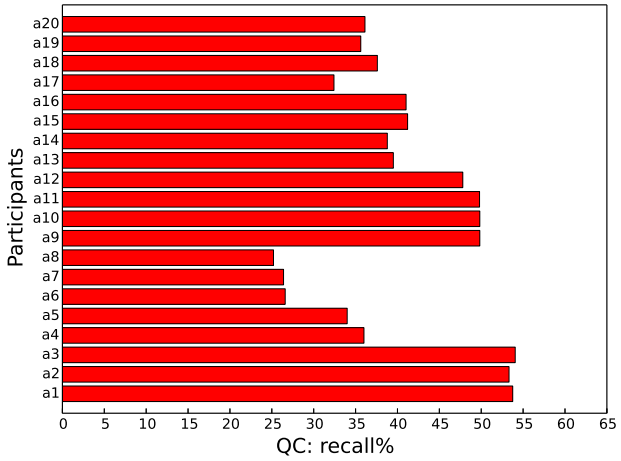

Figure 6.13: Recall rate for the list of competing algorithms and convolutional recurrent neural network on question classification collection. Evaluation is based on average value of twenty experiment with same hyper-parameters.

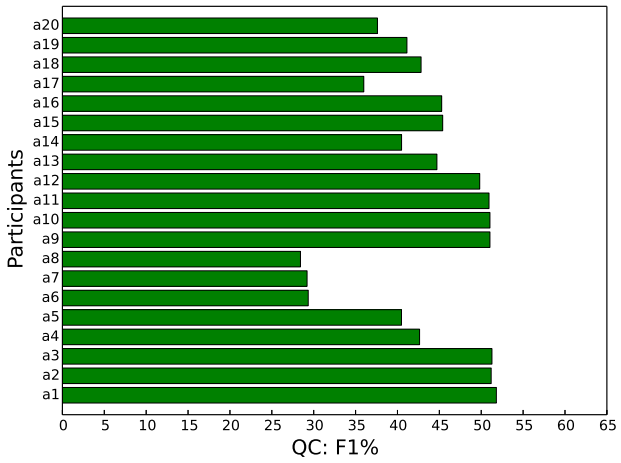

Figure 6.14: F1 score for the list of competing algorithms and convolutional recurrent neural network on question classification collection. Evaluation is based on average value of twenty experiment with same hyper-parameters.

For question classification corpus, precision rate, recall rate, and F1 score were displayed in Figure 6.12, 6.13, and 6.14 respectively. From Figure 6.12, convolutional recurrent neural network model ranks ninth on precision rate, with 2.31% less than

the best performer. For the recall ratio and F1 score, the proposed algorithm ranks first on both, achieving 4.23% and 0.77% more than the next best algorithm. The hypothesis 'Convolutional recurrent neural network offers better sentence classification performance than that of plain convolutional neural network.' can be verified to be yes in the context of question classification collection.

6.7.5 Runtime Evaluation for the Benchmark Data-sets

This specific subsection presents the system runtime efficiency for the proposed convolutional recurrent neural network model and each comparison framework. The elapsed time statistics is averaged from the twenty runs of each participant and aggregated from a complete 1,000 epochs or training steps.

The proposed convolutional recurrent neural network requires more system runtime compared to other participated singleton neural network models simply because it has larger quantity of neurons or parameters to train and update during the training process. Word-level convolutional neural network has longer system elapsed time than character-level convolutional neural network which is due to sophisticated input sequence encoding based on three-hundred word embeddings. In other words, character-level sentence encoding only produces low-hundred matrix for the neural network model. Likewise, word-aware recurrent neural network and character-aware recurrent neural network pair holds that order of runtime difference. One interesting observation from the runtime statistics evaluation on Table 6.6 is that it is much slower to train a neural network than that of traditional classifiers. That is due to the high number of hyper-parameters update mechanism and sophisticated hidden neurons weight optimisation inherited in neural network models.

For the test bed on the neural network-based performance evaluation, the present research deployed the codebase on the popular Google tensorflow environment for Ubuntu version 16.04 operating system. The codebase was executed on the Spyder IDE for the intensive programme evaluation. The hardware platform was configured

on an Intel eight core CPU, 32GB system RAM desktop computer with one Nvidia Tesla K40c high performance graphics card.

6.7.6 Findings from Benchmark Data-sets Evaluation

This chapter of work has previously mentioned about the performance comparison on convolutional recurrent neural network model over the competitors. It is time now analyse the performance difference on word-level convolutional neural networks, SVM classifier [234], k nearest neighbour-glsplwmd, and k nearest neighbour series.

The average precision rate and F1 score for word-level convolutional neural network+word2vec was 0.91% and 0.32% higher than word-level convolutional neural network+global vectors for word representation. Based on such a small scale data-set, this kind of difference was significant. Similarly, for word-level recurrent neural networks, the mean precision and F1 score for word2vec based was 1.06% and 0.47% higher than the Glove sponsored. The precision rate, recall rate, and F1 score difference between word-level convolutional neural network+global vectors for word representation and word-level convolutional neural network+random was 0.31%, 0.12%, and 0.28% respectively. For word-level recurrent neural network+random and word-level recurrent neural network+global vectors for word representation, the performance difference on precision rate, recall ratio, and F1 score was 2.64%, 3.43%, and 3.10% respectively.

For non-neural networks, the thesis emphasized the SVM, k nearest neighbour-word mover's distance and traditional k nearest neighbour. Sent2vec [42] encoding schema tries to interpret the sentential information into a single skip-thought vector rather than word level embeddings. From the corresponding results for precision rate, recall ratio, and F1 score, this present work observed that the performance of SVM+sent2vec was the worst in its series for all four benchmarking collections. This phenomenon also applied to k nearest neighbour-word mover's distance+sent2vec and k nearest neighbour+sent2vec.

This work explains non-neural networks in the context of word2vec and global vectors for word representation. For SVM, both produced resemblant precision, recall, and F1 score. The only exception was question classification collection where word2vec encoding yielded 8.29% more recall and 5.11% more F1 score. For the k nearest neighbour-word mover's distance, two embeddings posed almost identical impact under each measurement excepted the precision and F1 score on twenty-news-groups. Word2vec version yielded 6.15% more precision and 4.04% more F1 score. K nearest neighbour classifier followed the above tendency. The only exception was twenty-news-groups of which word2vec version produced 5.99%, 7.00%, and 6.65% more precision, recall, and F1 score respectively.

The present work now analyses the evaluation results for k nearest neighbour-word mover's distance and k nearest neighbour model. The former one utilised word mover's distance as the spatial distance function, the latter one applied plain cosine distance. From results for precision rate, recall ratio, and F1 score, readers can perceive that word mover's distance dominant the contest over precision, recall, and F1 score. The average precision for the former one was 42.17, the latter one was 39.27. For the average recall rate, word mover's distance sponsored models achieved 43.39% and plain k nearest neighbour obtained 37.43%. For the mean F1 score, the former one yielded 42.30%, the latter one produced 38.02%. It is obvious that word mover's distance was much better at measuring the spatial dissimilarity.

Referring to performance analysis for precision rate, recall ratio, and F1 score, long short-term memory, gated recurrent unit, and minimal gated unit produced almost equivalent result across four data-sets. This inspired the present research to conduct an additional experiment involving runtime. As can be perceived from Figure 6.15, minimal gated unit had the minimal average runtime under each benchmarking test.

From Table 6.7, it can be perceived that for comparison pair (convolutional recurrent neural network, a15) and (convolutional recurrent neural network, a16) against Google-news, there is no significant performance difference on F1 score,

which means convolutional recurrent neural network has similar F1 score on average compared to a15 and a16. For other pairs over Google-news, convolutional recurrent neural network achieves significantly better F1 score with its competitor.

For twenty-news-groups collection, practitioners can refer to second column of Table 6.7 for the Mann-Whitney-U test value. Convolutional recurrent neural network achieves statistically significant F1 score against each competitor algorithm at .01 level with 2-tailed hypothesis.

Third column of Table Table 6.7 displays the Mann-Whitney-U statistical test for Brown Corpus. It can be observed that convolutional recurrent neural network obtains significant result against each competing algorithm at specific level. More specifically, for the comparison pair (convolutional recurrent neural network, a4), statistical p value is yielded at .05 level. For other comparison pairs, p value is measured at .01 significance level.

For question classification collection, users can refer to fourth column at Table 6.7. It can be observed that convolutional recurrent neural network obtains significant result against each competing algorithm at specific level. More specifically, for the comparison pair (convolutional recurrent neural network, a10), statistical p value is yielded at .05 level. For other comparison pairs, p value is measured at .01 significance level.

Based on the Mann-Whitney-U statistical test over the four benchmark data-sets, the answer to the hypothesis 'Given identical hyper-parameters and benchmark data-sets, convolutional recurrent neural network model produces better sentence classification performance than that of the comparison frameworks.' is yes.

For the three variations of convolutional recurrent neural network, they produce almost identical performance based on the precision rate, recall ratio, and F1 score against the four experimental data-sets. Statistical verification on Mann-Whitney-U test shown in Table 6.8 indicates that performance difference between any pairs is not significant. System runtime is an important factor when implementing the algorithm. Figure 6.15 shows convolutional recurrent neural network-{minimal gated

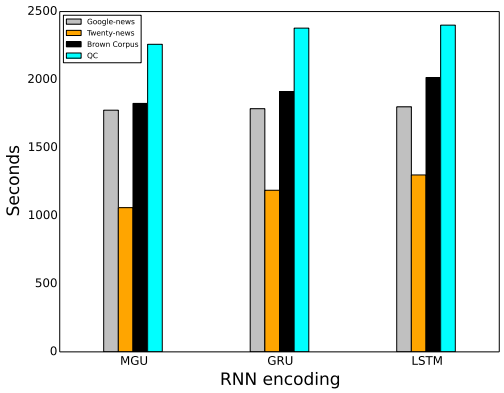

Figure 6.15: Convolutional recurrent neural network average runtime comparison with three different encodings.

unit, gated recurrent unit, long short-term memory} system elapsed time on the four data-sets. As can be perceived from Figure 6.15, minimal gated unit yields the minimal average runtime under each benchmarking test. Thus in practice minimal gated unit-based convolutional recurrent neural network is a strong candidate among the three variants. Therefore the answer to the hypothesis 'Long short-term memory, gated recurrent unit, and minimal gated unit offer statistically insignificant sentence categorisation performance when tested on same data-sets.' is yes. The hypothesis 'Minimal gated unit requires lowest computational overhead, followed by gated recurrent unit and long short-term memory when constructing recurrent unit.' is tested to be yes as well.

The most important conclusion from the designed experiments was that convolutional recurrent neural network model offered competing performance on all four data streams. For Google-news, convolutional recurrent neural network model obtained 2.62% more recall ratio than the next best. Convolutional recurrent neural network framework achieved 2.44% less precision rate and 0.57% less F1 score than the best one. For twenty-news-groups, the proposed algorithm produced the best precision rate, recall rate, and F1 score, with 3.68%, 1.00%, and 2.20% more than the next best. For Brown Corpus, convolutional recurrent neural network model

yielded 0.05% more F1 score than the next optimum and 0.54% less precision rate and 1.40% less recall than the optimal one. For question classification stream, convolutional recurrent neural network algorithm achieved the best recall rate and F1 score, having 4.23% and 0.77% more than the next best. For precision ratio, convolutional recurrent neural network model obtained 2.31% less than the best performer. This work attributed this to the superiority of the joint two neural networks, character-level convolutional neural network and character-aware recurrent neural network specifically. The aggregation layer in convolutional recurrent neural network combined together two neural structures to improve redundancy detection. The present thesis also perceived that character-level neural models often yielded better accuracy than word-level ones. Character-level encoding can capture subword information, misspelling, and grammatical errors, which formed a rich syntactical knowledge base.

Based on the evaluation on word2vec, global vectors for word representation, and randomised word vector initialisation, users can perceive that pre-trained word vectors exerted a positive effect on the classification tasks. Further, as word2vec contains a vocabulary of three million words and the volume of global vectors for word representation is 0.4 million, readers could expect that the possibility of out-of-vocabulary for word2vec is much lower than global vectors for word representation. From the evaluation results, word2vec sponsored models performed better than the others.

This work emphasised the difference between sentence level vectorisation (i.e., sent2vec [42]) and word level encodings on non-neural network frameworks. It is summarised that smaller embeddings unit can lead to better performance. The work attributed this to the rich word-level semantics understanding. Another important finding was that the substitution of word embeddings schema posed less effect on non-neural networks than neural models. This may be due to the way sentence similarity was computed. In non-neural models, spatial distance was often determined by true spatial distance.

The present work also observed that k nearest neighbour-word mover's distance gained better results than k nearest neighbour. This thesis attributed this to the accurate computation of the target word to the pivot word distance and summation of the corresponding distance piles. When combining base algorithm with the pre-trained word embeddings like word2vec and global vectors for word representation, the performance of k nearest neighbour-word mover's distance and k nearest neighbour can be further boosted.

Efficient runtime is important for practitioners because convolutional recurrent neural network involves large amount of training parameters. An effective encoding cell saves computational time and boosts throughput. From Figure 6.15, the current work observed that minimal gated unit required the minimum runtime without compromising performance. This conformed to the lowest number of training parameters within minimal gated unit.

6.8 Contribution to Sentence-level Neural Networks

The following contributions are made from this chapter.

- The work presented in this chapter proposes a novel sentence categorisation algorithm, which effectively exploits semantic information in text snippets to extract prominent features for categorisation;
- This chapter explores how long term morpheme understanding in character-aware recurrent neural network helps detecting sentence semantic similarity in text stream by integration of character-level convolutional neural network and character-aware recurrent neural network with an additional aggregating mechanism;
- The current chapter studies three different recurrent units like gated recurrent

unit [21], minimal gated unit [118], and long short-term memory [22] and compare their performance difference on four benchmark data-sets;

- This chapter performs a comparison on the character-aware neural network vs. the word-aware neural network;

- The work rendered in the chapter offers a performance comparison on the architecture, traditional character-level convolutional neural network structure, and plain character-aware recurrent neural network framework;

- This chapter carries out a group of non neural network systems comparison based on sent2vec [42] vs. word2vec [67, 69]/global vectors for word representation [70];

- The work explores into the efficacy differences between word2vec and global vectors for word representation [70] on the participating frameworks;

- The work conducted in this chapter exploits different state transformation units such as gated recurrent unit [21], minimal gated unit [118], and long short-term memory unit [22] in structuring the character-aware neural network and their usefulness through sentence classification performance.

6.9 Conclusion

This chapter shows the novel design and implementation of the proposed convolutional recurrent neural network model for sentence classification tasks. This chapter presents the relevant algorithms' performance evaluation to answers the following research questions and corresponding hypotheses.

For the question 'Can recurrent unit be leveraged to improve the convolutional neural network's performance in the context of sentence classification?', the following hypothesis can be derived.

(1) Convolutional recurrent neural network offers better sentence classification performance than that of plain convolutional neural network.

Based on the sentence classification results analysis, the above hypothesis is tested to be yes.

For the question 'How does the performance of long short-term memory differ from that of gated recurrent unit and minimal gated unit?', the subsequent two hypotheses can be yielded.

(1) Long short-term memory, gated recurrent unit, and minimal gated unit offer statistically insignificant sentence categorisation performance when tested on same data-sets.

(2) Minimal gated unit requires lowest computational overhead, followed by gated recurrent unit and long short-term memory when constructing recurrent unit.

Since long short-term memory, minimal gated unit, and gated recurrent unit offer similar expressive power with regard to recurrent unit on the four testing streams, the hypothesis 'Long short-term memory, gated recurrent unit, and minimal gated unit offer statistically insignificant sentence categorisation performance when tested on same data-sets.' is verified to be yes. For the hypothesis 'Minimal gated unit requires lowest computational overhead, followed by gated recurrent unit and long short-term memory when constructing recurrent unit.', a report on system elapsed time reveals that minimal gated unit requires on average least computational time, thus the answer to it is yes as well.

For the question 'How does the performance of convolutional recurrent neural network differ from that of the comparison frameworks for sentence classification?', the following hypothesis can be generated.

(1) Given identical hyper-parameters and benchmark data-sets, convolutional recurrent neural network model produces better sentence classification performance than that of the comparison frameworks.

According to the overall system performance examination, convolutional recurrent neural network achieves competing performance across the four benchmark corpus, therefore the answer to the above hypothesis is yes. For the final chapter, a summary of the whole thesis will be given, followed by contribution to the knowledge and research community, limitations of this present work, a conclusion section, and possible future work are offered.

Table 6.2: Char-CNN, Char-RNN, and CRNN stand for character-level convolutional neural network, character-aware recurrent neural network, and convolutional recurrent neural network respectively. GRU, MGU, and LSTM refer to gated recurrent unit, minimal gated unit, and long short-term memory respectively. Word-CNN and Word-RNN represent word-level convolutional neural network and word-level recurrent neural network respectively. GloVe stands for global vectors for word representation dense embeddings. KNN-WMD and KNN denote k nearest neighbour-word mover's distance and k nearest neighbour respectively.

Numbering	Base	Encoding
a1	CRNN	GRU
a2		MGU
a3		LSTM
a4	Char-CNN [17]	–
a5	Char-RNN [22]	–
a6	Word-CNN [75]	word2vec
a7		GloVe
a8		random
a9	Word-RNN [117]	word2vec
a10		GloVe
a11		random
a12	SVM [198]	word2vec
a13		GloVe
a14		sent2vec [42]
a15	KNN-WMD [23]	word2vec
a16		GloVe
a17		sent2vec [42]
a18	KNN	word2vec
a19		GloVe
a20		sent2vec [42]

Table 6.3: SVM parameters. Variable gamma defines the kernel coefficient. Shrinking heuristic is enabled here.

Name	**Value**
Penalty	1.0
gamma	1/n_features
Shrinking	True

Table 6.4: The list of designed experiments including algorithmic performance comparison based on benchmarking data-sets and distance metrics evaluation. The evaluation on performance difference is validated by the statistical family involving the Mann Whitney U test.

Comparison	**Evaluation Criteria**	Hypothesis
Each algorithm	Precision rate Recall ratio F1 score	There is significant performance difference on encoding

Table 6.5: Data-sets statistics. QC stands for Question Classification collection. EN denotes English language.

Data	**Google-news**	**Twenty-news**	**Brown Corpus**	**QC**
Size	2066	18000	57340	5952
Vocabulary	11194	252999	55528	8634
Total Words	49988	1806249	649408	32356
Mean Sent. Length	24.2	159.6	11.1	5.4
Meta-data	No	Yes	Yes	Yes
Classes	55	20	15	50
Language	EN	EN	EN	EN

Table 6.6: Runtime statistics for convolutional recurrent neural network and the comparison algorithms on four benchmark streams. The runtime is approximated in minute. CRNN denotes the proposed convolutional recurrent neural network framework.

Algorithm	**Google-news**	**Twenty-news**	**Brown Corpus**	**QC**
CRNN	185	211	195	204
Char-CNN	142	157	146	152
Char-RNN	138	151	143	149
Word-CNN	167	186	171	177
Word-RNN	166	181	169	172
SVM	<0.1	<0.1	<0.1	<0.1
KNN-WMD	<10	<10	<10	<10
KNN	<0.1	<0.1	<0.1	<0.1

Table 6.7: Mann-Whitney-U test p-value for F1 score against the list of competing algorithm and convolutional recurrent neural network. The experiment assumes 2-tailed hypothesis. This table includes statistical verification p for each benchmarking data stream. The term CRNN stands for convolutional recurrent neural network model.

Algorithm	**CRNN**			
	Google	Twenty	Brown	QC
a4	<.01	<.01	<.05	<.01
a5	<.01	<.01	<.01	<.01
a6	<.01	<.01	<.01	<.01
a7	<.01	<.01	<.01	<.01
a8	<.01	<.01	<.01	<.01
a9	<.01	<.01	<.01	<.01
a10	<.01	<.01	<.01	<.05
a11	<.01	<.01	<.01	<.01
a12	<.01	<.01	<.01	<.01
a13	<.01	<.01	<.01	<.01
a14	<.01	<.01	<.01	<.01
a15	>.05	<.01	<.01	<.01
a16	>.05	<.01	<.01	<.01
a17	<.01	<.01	<.01	<.01
a18	<.05	<.01	<.01	<.01
a19	<.01	<.01	<.01	<.01
a20	<.01	<.01	<.01	<.01

Table 6.8: Mann-Whitney-U test p-value for F1 score against the three variants of convolutional recurrent neural network. The experiment data-sets are the four benchmarking collections. Statistical hypothesis is two-tailed. MGU stands for minimal gated unit, GRU refers to gated recurrent unit, and LSTM represents long short-term memory cell respectively.

Comparison	p**-value**			
	Google	Twenty	Brown	QC
(MGU, GRU)	>.05	>.05	>.05	>.05
(MGU, LSTM)	>.05	>.05	>.05	>.05
(GRU, LSTM)	>.05	>.05	>.05	>.05

Chapter 7

Conclusion and Future Directions

The final chapter first summarises this thesis, then gives the overall contributions of the whole thesis to the knowledge and relevant research community, followed by an outlook of the real world applications and limitations of the work. Finally, the current chapter concludes the work presented in this thesis by revisiting the research questions and associated hypotheses raised in the introduction chapter and pointing possible future directions out.

7.1 Summary

In this thesis, two non-neural oriented sentence categorisation models are developed, namely hierarchical agglomerative clustering-word mover's distance and k nearest neighbour-word mover's distance for unsupervised learning and supervised learning respectively. Both models utilise word mover's distance as an accurate sentence dissimilarity metrics and offer robust performance over the four benchmarking data-sets. Since word mover's distance is a computationally expensive operator, the implementation is based on a multi-threading platform.

For the three available word frequency counting approaches, a processing speed comparison was conducted to systematically show how runtime statistics is affected by the changing input sentences size.

For the unsupervised learning, there are twelve different variants of hierarchical agglomerative clustering-word mover's distance. Each variant is constructed by three effective components, namely vector space model, distributed word embeddings, and a distortion function. The available vector space model options are two-fold, i.e., count vectoriser and binary one-hot vectorisation. Word2vec and global vectors for word representation are the two chosen pre-trained word embeddings for the hierarchical agglomerative clustering-word mover's distance framework. Word mover's distance, relaxed word mover's distance, and earth mover's distance are the three candidate distance functions. The proposed framework is compared to hierarchical agglomerative clustering-{cosine, Euclidean, Manhattan} and kmeans algorithm. The employed performance evaluation metrics include completeness score, homogeneity value, v-measure, and system elapsed time statistics. Since traditional earth mover's distance, relaxed word mover's distance, and word mover's distance are three variations of dense word similarity computation, the current work provides an additional accuracy comparison on them.

For the supervised learning, there are in total eighteen variations of k nearest neighbour-word mover's distance. The proposed novel k nearest neighbour-word mover's distance framework incorporates three distinct term frequency counting methods, dense word embeddings vector, and distance function to form an efficient sentence classification model. Similar to the construction and implementation of hierarchical agglomerative clustering-word mover's distance, k nearest neighbour-word mover's distance series has count vectoriser and binary one-hot vectorisation. The available distance functions are three-fold as well. However, k nearest neighbour-word mover's distance accepts term frequency-inverse document frequency model as an alternative vector space model which is not the case for hierarchical agglomerative clustering-word mover's distance. This work presented in the thesis first provides an overview of plain k nearest neighbour algorithm, followed by the detailed construction and implementation of the proposed k nearest neighbour-word mover's distance. The participated algorithms subsume k nearest neighbour-{cosine, Euclidean, Man-

hattan}, SVM, Naive Bayesian classifiers, and locality sensitive hashing. Precision rate, recall ratio, and F1 score are evaluated for each participant. Because traditional earth mover's distance, relaxed word mover's distance, and word mover's distance are three distortion variations for word embeddings, the work provides an additional accuracy comparison on them. These three distance functions are also evaluated by the t-distributed stochastic neighbour embedding visualisation tool to give an intuitive insight on accuracy of word similarity computation.

For neural network based framework, the work conducted in the present research applied the latest development on the character-aware deep neural classification network, character-level convolutional neural network and character-aware recurrent neural network specifically, for the effective categorisation of sentential snippets. The work extended the classical character-level convolutional neural network structure in a singleton to incorporate with the character-aware recurrent neural network framework to form an efficient sequence classification model. This novel framework benefits from the advantageous salient feature-filtering from the character-level convolutional neural network and the long term memory cells from the character-aware recurrent neural network. This work further explored the usefulness of an aggregation layer as a penultimate gate, gluing the encoding matrix generated from the character-level convolutional neural network and character-aware recurrent neural network jointly for convolutional recurrent neural network. From the experimental result it can be perceived that this enhances label classification accuracy. Evaluation on precision, recall, and F1 score indicated the efficacy of the proposed framework offered in this thesis. This work also assessed the effects of applying different hidden units on the benchmarking data-sets and reported their performance and runtime statistics. This thesis utilised the different word encoding schemes to deliver an exhaustive sequence labelling experiment.

7.2 Contributions

In this section, the contributions this thesis made to the relevant research community are summarised as follows.

Literature review chapter has made the following contributions.

- An examination of Topic Detection and Tracking methods and their implications to sentence categorisation tasks;
- A novel manner of decomposing count-based sentence categorisation into vector space model, distributed word embeddings, feature weighting, distance metrics, and similarity computation;
- Reviewing recent neural network models such as convolutional neural network, recurrent neural network, and their variations for the tasks of sentence-level categorisation;
- Interpreting neural network-based sentence categorisation framework based on ingredients such as cost function, optimiser, hyper-parameter optimisation, and activation function;
- Surveying prevalent neural model activation functions, neural network optimisers, and cost functions respectively for deep neural network construction;

The chapter of methodology contributes to the field of sentence categorisation community by the following directions:

- Composing a list of relevant data dimensionality reduction methods and their potential in the generic sentence categorisation tasks;
- An examination of related machine learning mechanisms and statistical significance test for sentence categorisation framework's performance evaluation;

- The chapter of generic methodology introduces a novel series of unsupervised sentence categorisation evaluation metrics which is applicable for other relevant work;
- A visualised presentation or demonstration of how T-distributed stochastic neighbour embedding can be deployed to verify sentence distortion function's efficacy.

The contributions made to the domain of sentence categorisation from the proposed novel supervised k nearest neighbour-word mover's distance and unsupervised hierarchical agglomerative clustering-word mover's distance are listed as follows.

- A novel sentence level distance metric word mover's distance is proposed for unsupervised sentence clustering;
- Word-level sentence encodings like word2vec and global vectors for word representation and sentence-aware distributed encoding scheme sent2vec are incorporated into the proposed novel frameworks;
- Hierarchical agglomerative clustering-word mover's distance is firstly introduced in the present work which offers optimum clustering performance on the provided two benchmarking data streams;
- The current work presented in the chapter extends the usual supervised word mover's distance-based k nearest neighbour [23] to the field of unsupervised sentence clustering, i.e., word mover's distance-based hierarchical agglomerative clustering framework;
- The presented supervised and unsupervised word mover's distance-oriented model work on sentence level textual snippets, where many existing models work on the document-level categorisation;

The following contributions have been made from the chapter of convolutional recurrent neural network framework.

- Convolutional recurrent neural network is a novel sentence classification model for efficiently exploring and extracting character-level latent semantic features from textual sentences;
- The model presented in the chapter exploits how long term dependency interpretation in character-aware recurrent neural network improves sentential semantic information understanding by incorporating character-level convolutional neural network with character-aware recurrent neural network on an added aggregator;
- The present work considers gated recurrent unit [21], minimal gated unit [118], and long short-term memory [22] in the convolutional recurrent neural network model construction and evaluates their performance difference based on four benchmark data-sets;
- Character-aware neural network and word-aware neural networks are involved in the extensive experiment in current chapter of research;
- Plain character-level convolutional neural network and classical character-aware recurrent neural network framework are participated in the sentence classification performance evaluation;
- Word-level dense embeddings and sentence-level encodings are examined in the experiment of the present chapter and performance or efficacy differences are reported;

7.3 Application

This section discusses the potential application which can be derived from the proposed two sentence categorisation frameworks. As sentence categorisation in textual data mining is a fundamental problem in natural language processing, it has wide coverage in application areas.

For instance, stance classification, opinion mining [3], document recommendation system [235], plagiarism detection, and social stream event detection [176] are well-known applications to sentence categorisation. More specifically, those aforementioned research fields require distilled bag-of-words features, sentence distance metrics, and label classification, which are central to word mover's distance-based sentential snippets categorisation offered in the current work [36, 11].

Since the continuous evolution on the Internet and E-commerce, large scale recommender system architecture is the current tendency [179]. Dimensionality reduction techniques are highly applicable under this circumstance. For instance, collaborative filtering-based recommendation procedure has shown high quality recommendations due to incorporating consumer preferences with new consumer's preferences. However, this procedure has limited implication for large-scale consumer-product database. Document compression deployed in the word mover's distance-supported framework can be readily transferred to traditional recommender.

A product recommender engine comprises of user profiling and document feature extraction to generate recommendation for the target customers [235]. Document feature matrix requires vector space model and recommendation process is achieved by computing pairwise similarity. The usual outcome of a recommender is a list of documents which is considered to be most relevant to the user query. Model developed in Chap. 5 employs an efficient word mover's distance-based distance metrics, suggesting strong application to recommendation systems.

Automated machine translation, neural machine translation in particular utilises autoencoder (an instance of sequence to sequence learning [236]) to learn and obtain good sentence representation. Attention regime [237] is often deployed to optimise word-to-word translation process. It is very useful when an aligned language pair like English-German translation is performed. The proposed convolutional recurrent neural network developed in the present research sheds some light on sequence learning in automated machine translation by character-level sentence encodings.

N-gram language model was initially advocated for natural language processing

and has led the community for several decades. However, classical n-gram models like uni-gram, bi-gram, and tri-gram do not scale well on real world tasks due to the high dimensional gram dictionary [20]. The novel word mover's distance-based models implemented in the current thesis enhances the traditional n-gram models by optimising the overall bag-of-words feature vectors through space saving. The n-gram models can be further extended from counting-based to word mover's distance-underpinned for more accurate sentence categorisation.

7.4 Limitations

For benchmarking data streams selection, this thesis focuses on English corpus only due to space constraint and PhD timeframe. It is known that other human languages such as Chinese has no space when phrasing a sentence, which will pose a significant difference in word splitting strategy. The sentence splitting approaches and vector space modelling employed in this thesis are suitable for English data-sets. The present work has not tested the pre-processing method on Chinese texts yet. In fact, Chinese word segmentation is usually conducted with either word pair dictionary look-up or statistical counting [65].

For the word mover's distance-based model development and implementation, this work considers multi-threading enabled python environment. Since pairwise distance computation requires high computational overhead, conventional software platforms and commodity computers have disadvantages regarding the performance. However, so far there is no off-the-shelf computing resources or clusters support. Therefore there is limitation in testing word mover's distance, earth mover's distance, and relaxed-word mover's distance on large scale cluster.

Deep neural network and its variants have gone through a rapid growth because of the evolution on neural hardware, graphics card, and large scale cluster. Automatic machine translation through neural network has seen many successful applications and frameworks. For example, encoder-decoder structure and attention

mechanism can be applied to learn an optimised feature distribution with reduced spatial complexity.

Recurrent neural network like long short-term memory has a series of model parameters. The present work leaves some hyper-parameters like peep-hole connector [238] and grid long short-term memory cell [239] free of fine-tuning and uses vanilla setting instead.

7.5 Conclusions

Natural language mining and understanding has been a open issue and human language is a reflection of human intelligence and engages with a huge amount of grammatical rules. Such nature of human language implies that machine interpretation of it will be a strong challenge. This thesis attempts to develop efficient and effective sentence categorisation frameworks, namely word mover's distance-based term frequency model and convolutional recurrent neural network algorithm. Text classification or sentence categorisation empowers machines to understand and encode human languages in an efficient manner.

The broad problem domain this work tends to solve is twofold. First, this work studies the usefulness of word mover's distance on categorising unlabelled/unclassified sentences with respect to sentence similarity metric. Second, this thesis investigates deep neural model's effectiveness on improving sentence classification performance. Recall that this work attempts to raise and answer a list of six research questions and correspondent hypotheses.

Research questions that this work raises and answers are summarised as follows.

(1) How does the expressiveness of term frequency-inverse document frequency differ from that of count vectoriser and binary one-hot vectorisation?

(2) Can word mover's distance-based distance function improve the accu-

racy of sentence similarity measurement over existing spatial distance functions?

(3) How does the performance of word mover's distance-based sentence categorisation models differ from that of the other competing algorithms?

(4) Can recurrent unit be leveraged to improve the convolutional neural network's performance in the context of sentence classification?

(5) How does the performance of long short-term memory differ from that of gated recurrent unit and minimal gated unit?

(6) How does the performance of convolutional recurrent neural network differ from that of the comparison frameworks for sentence classification?

In the following the stemmed research hypotheses are given with each research question deriving a list of corresponding hypotheses.

For the research question 'How does the expressiveness of term frequency-inverse document frequency differ from that of count vectoriser and binary one-hot vectorisation?', two hypotheses can be obtained.

(1) Given same quantity of sentences for vector space model matrix construction, binary one-hot vectorisation obtains the best time efficiency, followed by count vectorisation and term frequency-inverse document frequency.

(2) Term frequency-inverse document frequency vectoriser provides better vector space model representation than that of count vectoriser and binary one-hot vectorisation, leading to better sentence categorisation performance.

From the research question 'Can word mover's distance-based distance function improve the accuracy of sentence similarity measurement over existing spatial distance functions?', the following hypotheses can be generated:

(1) Given same benchmark data stream, word2vec dense embeddings offers better word vector representation than global vectors for word representation distributed embeddings scheme.

(2) Word mover's distance provides better sentence classification performance than earth mover's distance and relaxed word mover's distance over standard machine learning metrics on the specified supervised benchmark data-sets.

(3) Word mover's distance provides superior sentence clustering performance against earth mover's distance and relaxed word mover's distance over standard machine learning metrics on the specified unsupervised benchmark data-sets.

(4) Word mover's distance-based distance function improves the accuracy of sentence similarity measurement over cosine, Euclidean, and Manhattan on the specified benchmark textual streams.

The research question 'How does the performance of word mover's distance-based sentence categorisation models differ from that of the other competing algorithms?' implies the following two hypotheses

(1) Given same testing sequences, k nearest neighbour-word mover's distance yields better performance than the other comparison frameworks with respect to precision rate, recall ratio, and F1 score.

(2) Given same testing sequences, Hierarchical agglomerative clustering-word mover's distance achieves better mean squared error value, homogeneity score, completeness value, and v-measure than the competing algorithms in unsupervised learning.

The aforementioned three research questions and associated hypotheses are answered in Chapter 5. The next three presented research questions and derived hypotheses are addressed in Chapter 6.

For the question 'Can recurrent unit be leveraged to improve the convolutional neural network's performance in the context of sentence classification?', the following hypothesis can be derived.

(1) Convolutional recurrent neural network offers better sentence classification performance than that of plain convolutional neural network.

For the question 'How does the performance of long short-term memory differ from that of gated recurrent unit and minimal gated unit?', the subsequent two hypotheses can be yielded.

(1) Long short-term memory, gated recurrent unit, and minimal gated unit offer statistically insignificant sentence categorisation performance when tested on same data-sets.

(2) Minimal gated unit requires lowest computational overhead, followed by gated recurrent unit and long short-term memory when constructing recurrent unit.

For the question 'How does the performance of convolutional recurrent neural network differ from that of the comparison frameworks for sentence classification?', the following hypothesis can be generated.

(1) Given identical hyper-parameters and benchmark data-sets, convolutional recurrent neural network model produces better sentence classification performance than that of the comparison frameworks.

7.6 Future Work

For convolutional recurrent neural network-based experiment, it is hoped to extend the model to a wide variety of multi-class benchmark data for a generic sentence

classification framework and performance evaluation.

Deep neural network and its variants have gone through a rapid growth because of the evolution on neural hardware, graphics card, and large scale cluster. Automatic machine translation through neural network has seen many successful applications and frameworks. For example, encoder-decoder structure and attention mechanism can be applied to reconstruct input sequence for optimised features distribution.

Recurrent neural network like long short-term memory has a series of model parameters like peep-hole connector [238] and grid long short-term memory cell [239]. Those aforementioned variations can be alternatives in the future versions of the convolutional recurrent neural network framework. Besides, character-word combination during the preparation of deep neural network input sequences is an optional approach for long short-term memory model [240]. This can be a comparison method for the convolutional recurrent neural network model which accepts pure character-aware encoding and word-level neural networks.

Bibliography

[1] Z. Zhu and J. Hu, "Context aware document embedding," *arXiv Preprint arXiv:1707.01521*, 2017.

[2] S.-H. Lo, H. Meng, and W. Lam, "Automatic bilingual text document summarization," in *Proceedings of the 6th World Multiconference on Systematic, Cybernetics and Informatics. Orlando, Florida, USA*, 2002.

[3] N. Alsaedi, P. Burnap, and O. Rana, "Temporal tf-idf: A high performance approach for event summarization in twitter," in *Web Intelligence (WI), 2016 IEEE/WIC/ACM International Conference on*, pp. 515–521, IEEE, 2016.

[4] A. J. Roa-Valverde and M.-A. Sicilia, "A survey of approaches for ranking on the web of data," *Information Retrieval*, vol. 17, no. 4, pp. 295–325, 2014.

[5] J. B. Vuurens and A. P. de Vries, "Distance matters! cumulative proximity expansions for ranking documents," *Information Retrieval*, vol. 17, no. 4, pp. 380–406, 2014.

[6] J. Parapar, A. Bellogín, P. Castells, and Á. Barreiro, "Relevance-based language modelling for recommender systems," *Information Processing & Management*, vol. 49, no. 4, pp. 966–980, 2013.

[7] T. Chen, Y. Sun, Y. Shi, and L. Hong, "On sampling strategies for neural network-based collaborative filtering," in *Knowledge Discovery and Data*

Mining, Proceedings of the 23rd ACM SIGKDD International Conference on. ACM, 2017.

[8] M. Deshpande and G. Karypis, "Item-based top-n recommendation algorithms," *Information Systems, ACM Transactions on (TOIS)*, vol. 22, no. 1, pp. 143–177, 2004.

[9] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," in *Association for Computational Linguistics, Proceedings of the 34th Annual Meeting on*, pp. 310–318, Association for Computational Linguistics, 1996.

[10] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush, "Character-aware neural language models," in *Artificial Intelligence, 13th AAAI Conference on* , 2016.

[11] X. Fu, E. Ch'ng, U. Aickelin, and L. Zhang, "An improved system for sentence-level novelty detection in textual streams," in *Smart and Sustainable City and Big Data (ICSSC), 2015 International Conference on*, pp. 1–6, IET, 2015.

[12] Y. Rao and Q. Li, "Term weighting schemes for emerging event detection," in *Web Intelligence and Intelligent Agent Technology, Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on -Volume 01*, pp. 105–112, IEEE Computer Society, 2012.

[13] T. Brants, F. Chen, and A. Farahat, "A system for new event detection," in *Research and Development in Informaion Retrieval, Proceedings of the 26th Annual International ACM SIGIR Conference on* , pp. 330–337, ACM, 2003.

[14] J. Allan, V. Lavrenko, D. Malin, and R. Swan, "Detections, bounds, and timelines: Umass and tdt-3," in *Proceedings of Topic Detection and Tracking Workshop*, pp. 167–174, 2000.

[15] J. G. Fiscus and G. R. Doddington, "Topic detection and tracking evaluation overview," in *Topic Detection and Tracking*, pp. 17–31, Springer, 2002.

[16] S. Vosoughi, P. Vijayaraghavan, and D. Roy, "Tweet2vec: Learning tweet embeddings using character-level cnn-lstm encoder-decoder," in *Research and Development in Information Retrieval, Proceedings of the 39th International ACM SIGIR conference on*, pp. 1041–1044, ACM, 2016.

[17] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in Neural Information Processing Systems*, pp. 649–657, 2015.

[18] J. Li, M.-T. Luong, and D. Jurafsky, "A hierarchical neural autoencoder for paragraphs and documents," *arXiv Preprint arXiv:1506.01057*, 2015.

[19] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, no. Feb, pp. 1137–1155, 2003.

[20] P. F. Brown, P. V. Desouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai, "Class-based n-gram models of natural language," *Computational Linguistics*, vol. 18, no. 4, pp. 467–479, 1992.

[21] K. Cho, B. Van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv Preprint arXiv:1406.1078*, 2014.

[22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[23] M. Kusner, Y. Sun, N. Kolkin, and K. Q. Weinberger, "From word embeddings to document distances," in *Machine Learning, Proceedings of the 32nd International Conference on (ICML-15)*, pp. 957–966, 2015.

[24] G. Huang, C. Guo, M. J. Kusner, Y. Sun, F. Sha, and K. Q. Weinberger, "Supervised word mover's distance," in *Advances in Neural Information Processing Systems*, pp. 4862–4870, 2016.

[25] M. Steinbach, G. Karypis, V. Kumar, *et al.*, "A comparison of document clustering techniques," in *Text Mining, KDD Workshop on*, vol. 400, pp. 525–526, Boston, 2000.

[26] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *International Journal of Computer Vision*, vol. 40, no. 2, pp. 99–121, 2000.

[27] O. Pele and M. Werman, "Fast and robust earth mover's distances," in *Computer vision, 2009 IEEE 12th International Conference on*, pp. 460–467, IEEE, 2009.

[28] Z. Ren, J. Yuan, and Z. Zhang, "Robust hand gesture recognition based on finger-earth mover's distance with a commodity depth camera," in *Multimedia Proceedings of the 19th ACM International Conference on*, pp. 1093–1096, ACM, 2011.

[29] T. Mikolov, "Statistical language models based on neural networks," *Presentation at Google, Mountain View, 2nd April*, 2012.

[30] H. T. Le, C. Cerisara, and A. Denis, "Do convolutional networks need to be deep for text classification?," *arXiv Preprint arXiv:1707.04108*, 2017.

[31] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv Preprint arXiv:1508.04025*, 2015.

[32] F. Hill, K. Cho, S. Jean, and Y. Bengio, "The representational geometry of word meanings acquired by neural machine translation models," *Machine Translation*, pp. 1–16, 2017.

[33] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Nural Information Processing Systems*, pp. 1097–1105, 2012.

[34] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of Machine Learning Research*, vol. 12, no. Aug, pp. 2493–2537, 2011.

[35] Y. Zhang, S. Roller, and B. Wallace, "Mgnc-cnn: A simple approach to exploiting multiple word embeddings for sentence classification," *arXiv Preprint arXiv:1603.00968*, 2016.

[36] X. Fu, E. Ch'ng, U. Aickelin, and S. See, "CRNN: a joint neural network for redundancy detection," in *Smart Computing (SMARTCOMP), 2017 IEEE International Conference on*, pp. 1–8, IEEE, 2017.

[37] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[38] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[39] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, p. 391, 1990.

[40] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *Computational Geometry, Proceedings of the 20th Annual Symposium on*, pp. 253–262, ACM, 2004.

[41] C. C. Aggarwal and C. Zhai, "A survey of text classification algorithms," *Mining Text Data*, pp. 163–222, 2012.

[42] R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, and S. Fidler, "Skip-thought vectors," in *Advances in Neural Information Processing Systems*, pp. 3294–3302, 2015.

[43] S. Tang, H. Jin, C. Fang, Z. Wang, and V. R. de Sa, "Rethinking skip-thought: A neighborhood based approach," *arXiv Preprint arXiv:1706.03146*, 2017.

[44] S. Tang, H. Jin, C. Fang, Z. Wang, and V. R. de Sa, "Trimming and improving skip-thought vectors," *arXiv Preprint arXiv:1706.03148*, 2017.

[45] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[46] C. Fellbaum, *WordNet*. Wiley Online Library, 1998.

[47] A. Tato, R. Nkambou, A. Dufresne, and M. H. Beauchamp, "Convolutional neural network for automatic detection of sociomoral reasoning level,"

[48] W. Ling, T. Luís, L. Marujo, R. F. Astudillo, S. Amir, C. Dyer, A. W. Black, and I. Trancoso, "Finding function in form: Compositional character models for open vocabulary word representation," *arXiv Preprint arXiv:1508.02096*, 2015.

[49] M. Karkali, F. Rousseau, A. Ntoulas, and M. Vazirgiannis, "Using temporal idf for efficient novelty detection in text streams," *arXiv Preprint arXiv:1401.1456*, 2014.

[50] Y. Linde, A. Buzo, and R. Gray, "An algorithm for vector quantizer design," *Communications, IEEE Transactions on*, vol. 28, no. 1, pp. 84–95, 1980.

[51] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in Neural Information Processing Systems*, pp. 556–562, 2001.

[52] D. Seung and L. Lee, "Algorithms for non-negative matrix factorization," *Advances in Neural Information Processing Systems*, vol. 13, pp. 556–562, 2001.

[53] S. Petrović, M. Osborne, and V. Lavrenko, "Streaming first story detection with application to twitter," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 181–189, Association for Computational Linguistics, 2010.

[54] J. Allan, J. G. Carbonell, G. Doddington, J. Yamron, and Y. Yang, "Topic detection and tracking pilot study final report," 1998.

[55] Z. Yang, H. Ma, Z. He, and X. S. Wang, "Finding maximal ranges with unique topics in a text database," *World Wide Web*, pp. 1–22, 2017.

[56] J. Allan, *Topic Detection and Tracking: Event-based Information Organization*, vol. 12. Springer Science & Business Media, 2012.

[57] Y. Gong and W. Xu, *Machine Learning for Multimedia Content Analysis*, vol. 30. Springer Science & Business Media, 2007.

[58] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki, "The det curve in assessment of detection task performance," tech. rep., DTIC Document, 1997.

[59] Darpa, *Broadcast News Workshop'99 Proceedings*. Morgan Kaufmann, 1999.

[60] X. Li and D. Roth, "Learning question classifiers," in *Computational linguistics, Proceedings of the 19th International Conference on -Volume 1*, pp. 1–7, Association for Computational Linguistics, 2002.

[61] J. Allan, C. Wade, and A. Bolivar, "Retrieval and novelty detection at the sentence level," in *Research and Development in Informaion Retrieval, Proceed-*

ings of the 26th Annual International ACM SIGIR Conference on, pp. 314–321, ACM, 2003.

[62] M. McCandless, E. Hatcher, and O. Gospodnetic, *Lucene in Action: Covers Apache Lucene 3.0*. Manning Publications Co., 2010.

[63] E. Hatcher and O. Gospodnetic, "Lucene in action," 2004.

[64] R. Papka, J. Allan, *et al.*, "On-line new event detection using single pass clustering," *University of Massachusetts, Amherst*, pp. 37–45, 1998.

[65] C. Huang and H. Zhao, "Chinese word segmentation: A decade review," *Journal of Chinese Information Processing*, vol. 21, no. 3, pp. 8–20, 2007.

[66] M. Zechner, M. Muhr, R. Kern, and M. Granitzer, "External and intrinsic plagiarism detection using vector space models," in *Proc. SEPLN*, vol. 32, pp. 47–55, 2009.

[67] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv Preprint arXiv:1301.3781*, 2013.

[68] J. Reisinger and R. J. Mooney, "Multi-prototype vector-space models of word meaning," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 109–117, Association for Computational Linguistics, 2010.

[69] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems*, pp. 3111–3119, 2013.

[70] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *EMNLP*, vol. 14, pp. 1532–43, 2014.

[71] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," *arXiv Preprint arXiv:1405.4053*, 2014.

[72] D. A. Huffman, "A method for the construction of minimum-redundancy codes," *Proceedings of the IRE*, vol. 40, no. 9, pp. 1098–1101, 1952.

[73] C. N. Dos Santos and M. Gatti, "Deep convolutional neural networks for sentiment analysis of short texts.," in *COLING*, pp. 69–78, 2014.

[74] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv Preprint arXiv:1408.5882*, 2014.

[75] Y. Zhang and B. Wallace, "A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification," *arXiv Preprint arXiv:1510.03820*, 2015.

[76] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *arXiv Preprint arXiv:1607.04606*, 2016.

[77] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," *arXiv Preprint arXiv:1607.01759*, 2016.

[78] X. Li and W. B. Croft, "Improving novelty detection for general topics using sentence level information patterns," in *Information and Knowledge Management, Proceedings of the 15th ACM International Conference on*, pp. 238–247, ACM, 2006.

[79] A. Doko, M. Stula, and D. Stipanicev, "A recursive tf-isf based sentence retrieval method with local context," *International Journal of Machine Learning and Computing*, vol. 3, no. 2, p. 195, 2013.

[80] C. Blake, "A comparison of document, sentence, and term event spaces," in *Computational Linguistics, Proceedings of the 21st International Conference on and the 44th Annual Meeting of the Association for Computational Linguistics*, pp. 601–608, Association for Computational Linguistics, 2006.

[81] S. Robertson, H. Zaragoza, *et al.*, "The probabilistic relevance framework: Bm25 and beyond," *Foundations and Trends® in Information Retrieval*, vol. 3, no. 4, pp. 333–389, 2009.

[82] K. Soumya George and S. Joseph, "Text classification by augmenting bag of words (bow) representation with co-occurrence feature,"

[83] S. Robertson, H. Zaragoza, and M. Taylor, "Simple bm25 extension to multiple weighted fields," in *Information and Knowledge Management, Proceedings of the 13th ACM International Conference on*, pp. 42–49, ACM, 2004.

[84] J. R. Pérez-Agüera, J. Arroyo, J. Greenberg, J. P. Iglesias, and V. Fresno, "Using bm25f for semantic search," in *Proceedings of the 3rd International Semantic Search Workshop*, p. 2, ACM, 2010.

[85] S. Campinas, R. Delbru, and G. Tummarello, "Effective retrieval model for entity with multi-valued attributes: Bm25mf and beyond," in *Knowledge Engineering and Knowledge Management, International Conference on*, pp. 200–215, Springer, 2012.

[86] Y. Yang, T. Pierce, and J. Carbonell, "A study of retrospective and on-line event detection," in *Research and Development in Information Retrieval, Proceedings of the 21st Annual International ACM SIGIR Conference on*, pp. 28–36, ACM, 1998.

[87] Y. Rao, Q. Li, Q. Wu, H. Xie, F. L. Wang, and T. Wang, "A multi-relational term scheme for first story detection," *Neurocomputing*, 2017.

[88] M.-C. Su and C.-H. Chou, "A modified version of the k-means algorithm with a distance based on cluster symmetry," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, no. 6, pp. 674–680, 2001.

[89] S.-H. Cha, "Comprehensive survey on distance/similarity measures between probability density functions," *City*, vol. 1, no. 2, p. 1, 2007.

[90] P. J. Groenen and K. Jajuga, "Fuzzy clustering with squared minkowski distances," *Fuzzy Sets and Systems*, vol. 120, no. 2, pp. 227–237, 2001.

[91] J. V. De Oliveira and W. Pedrycz, *Advances in Fuzzy Clustering and its Applications*. John Wiley & Sons, 2007.

[92] L. Hamers, Y. Hemeryck, G. Herweyers, M. Janssen, H. Keters, R. Rousseau, and A. Vanhoutte, "Similarity measures in scientometric research: the jaccard index versus salton's cosine formula," *Information Processing & Management*, vol. 25, no. 3, pp. 315–318, 1989.

[93] G. Dinu and M. Lapata, "Measuring distributional similarity in context," in *Empirical Methods in Natural Language Processing, Proceedings of the 2010 Conference on*, pp. 1162–1172, Association for Computational Linguistics, 2010.

[94] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih, "Image indexing using color correlograms," in *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pp. 762–768, IEEE, 1997.

[95] M. A. Stricker and M. Orengo, "Similarity of color images," in *IS&T/SPIE's Symposium on Electronic Imaging: Science & Technology*, pp. 381–392, International Society for Optics and Photonics, 1995.

[96] J. Goldberger, S. Gordon, H. Greenspan, *et al.*, "An efficient image similarity measure based on approximations of kl-divergence between two gaussian mixtures.," in *ICCV*, vol. 3, pp. 487–493, 2003.

[97] S. Lee, "Online sentence novelty scoring for topical document streams.," in *EMNLP*, pp. 567–572, Citeseer, 2015.

[98] Y. Rubner, C. Tomasi, and L. J. Guibas, "A metric for distributions with applications to image databases," in *Computer Vision, 1998. 6th International Conference on*, pp. 59–66, IEEE, 1998.

[99] M. Steyvers, P. Smyth, M. Rosen-Zvi, and T. Griffiths, "Probabilistic author-topic models for information discovery," in *Knowledge Discovery and Data Mining, Proceedings of the 10th ACM SIGKDD International Conference on*, pp. 306–315, ACM, 2004.

[100] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth, "The author-topic model for authors and documents," in *Uncertainty in Artificial Intelligence, Proceedings of the 20th Conference on*, pp. 487–494, AUAI Press, 2004.

[101] E. Hellinger, "Neue begründung der theorie quadratischer formen von unendlichvielen veränderlichen.," *Journal Für die Reine Und Angewandte Mathematik*, vol. 136, pp. 210–271, 1909.

[102] C.-H. Lee and D.-G. Shin, "Using hellinger distance in a nearest neighbour classifier for relational databases," *Knowledge-Based Systems*, vol. 12, no. 7, pp. 363–370, 1999.

[103] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv Preprint arXiv:1706.03762*, 2017.

[104] R. Johnson and T. Zhang, "Semi-supervised convolutional neural networks for text categorization via region embedding," in *Advances in Neural Information Processing Systems*, pp. 919–927, 2015.

[105] P. Liu, X. Qiu, and X. Huang, "Dynamic compositional neural networks over tree structure," *arXiv Preprint arXiv:1705.04153*, 2017.

[106] R. Johnson and T. Zhang, "Deep pyramid convolutional neural networks for text categorization," in *Proceedings of the 55th Annual Meeting of the Associ-*

ation for Computational Linguistics (Volume 1: Long Papers), vol. 1, pp. 562–570, 2017.

[107] R. Kneser and H. Ney, "Improved backing-off for m-gram language modeling," in *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, vol. 1, pp. 181–184, IEEE, 1995.

[108] R. Nallapati and J. Allan, "Capturing term dependencies using a language model based on sentence trees," in *Information and Knowledge Management, Proceedings of the 11th International Conference on*, pp. 383–390, ACM, 2002.

[109] R. Jozefowicz, O. Vinyals, M. Schuster, N. Shazeer, and Y. Wu, "Exploring the limits of language modeling," *arXiv Preprint arXiv:1602.02410*, 2016.

[110] H. B. Demuth, M. H. Beale, O. De Jess, and M. T. Hagan, *Neural Network Design*. Martin Hagan, 2014.

[111] D. CiresAn, U. Meier, J. Masci, and J. Schmidhuber, "Multi-column deep neural network for traffic sign classification," *Neural Networks*, vol. 32, pp. 333–338, 2012.

[112] O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, and G. Penn, "Applying convolutional neural networks concepts to hybrid nn-hmm model for speech recognition," in *Acoustics, Speech and Signal Processing, 2012 IEEE International Conference on (ICASSP)*, pp. 4277–4280, IEEE, 2012.

[113] M. Kempka, M. Wydmuch, G. Runc, J. Toczek, and W. Jaskowski, "Vizdoom: A doom-based ai research platform for visual reinforcement learning," *arXiv Preprint arXiv:1605.02097*, 2016.

[114] J. Chung, K. Cho, and Y. Bengio, "A character-level decoder without explicit segmentation for neural machine translation," *arXiv Preprint arXiv:1603.06147*, 2016.

[115] W. Yin and H. Schutze, "Multichannel variable-size convolution for sentence classification," *arXiv Preprint arXiv:1603.04513*, 2016.

[116] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv Preprint arXiv:1412.3555*, 2014.

[117] J. Chung, C. Gülçehre, K. Cho, and Y. Bengio, "Gated feedback recurrent neural networks," *CoRR, abs/1502.02367*, 2015.

[118] G.-B. Zhou, J. Wu, C.-L. Zhang, and Z.-H. Zhou, "Minimal gated unit for recurrent neural networks," *International Journal of Automation and Computing*, vol. 13, no. 3, pp. 226–234, 2016.

[119] A. Karpathy, J. Johnson, and L. Fei-Fei, "Visualizing and understanding recurrent networks," *arXiv Preprint arXiv:1506.02078*, 2015.

[120] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification," in *AAAI*, pp. 2267–2273, 2015.

[121] Y. Xiao and K. Cho, "Efficient character-level document classification by combining convolution and recurrent layers," *arXiv Preprint arXiv:1602.00367*, 2016.

[122] N. Murata, S. Yoshizawa, and S.-i. Amari, "Network information criterion-determining the number of hidden units for an artificial neural network model," *Neural Networks, IEEE Transactions on*, vol. 5, no. 6, pp. 865–872, 1994.

[123] D. F. Specht, "Probabilistic neural networks for classification, mapping, or associative memory," in *Neural Networks, IEEE International Conference on*, vol. 1, pp. 525–532, 1988.

[124] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Computational Learning Theory, Proceedings of the 5th Annual Workshop on*, pp. 144–152, ACM, 1992.

[125] D. M. Kline and V. L. Berardi, "Revisiting squared-error and cross-entropy functions for training neural network classifiers," *Neural Computing & Applications*, vol. 14, no. 4, pp. 310–318, 2005.

[126] T. Zhang, "Solving large scale linear prediction problems using stochastic gradient descent algorithms," in *Machine Learning, Proceedings of the 21st International Conference on*, p. 116, ACM, 2004.

[127] W. Xu, "Towards optimal one pass large scale learning with averaged stochastic gradient descent," *arXiv Preprint arXiv:1107.2490*, 2011.

[128] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.

[129] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop, coursera: Neural networks for machine learning," *University of Toronto, Tech. Rep*, 2012.

[130] T. Schaul, S. Zhang, and Y. LeCun, "No more pesky learning rates.," *ICML (3)*, vol. 28, pp. 343–351, 2013.

[131] N. L. Roux and A. W. Fitzgibbon, "A fast natural newton method," in *Machine Learning, Proceedings of the 27th International Conference on (ICML-10)*, pp. 623–630, 2010.

[132] M. D. Zeiler, "Adadelta: an adaptive learning rate method," *arXiv Preprint arXiv:1212.5701*, 2012.

[133] P. Burge and J. Shawe-Taylor, "An unsupervised neural network approach to profiling the behavior of mobile phone users for use in fraud detection," *Journal of Parallel and Distributed Computing*, vol. 61, no. 7, p. 915925, 2001.

[134] S. Basu, M. Bilenko, and R. J. Mooney, "A probabilistic framework for semi-supervised clustering," in *Knowledge Discovery and Data Mining, Proceedings of the 10th ACM SIGKDD International Conference on*, pp. 59–68, ACM, 2004.

[135] R. Kompass, "A generalized divergence measure for nonnegative matrix factorization," *Neural Computation*, vol. 19, no. 3, pp. 780–791, 2007.

[136] C. Févotte, N. Bertin, and J.-L. Durrieu, "Nonnegative matrix factorization with the itakura-saito divergence: With application to music analysis," *Neural Computation*, vol. 21, no. 3, pp. 793–830, 2009.

[137] S. Zafeiriou, "Discriminant nonnegative tensor factorization algorithms," *Neural Networks, IEEE Transactions on*, vol. 20, no. 2, pp. 217–235, 2009.

[138] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv Preprint arXiv:1609.04747*, 2016.

[139] R. S. Sexton, B. Alidaee, R. E. Dorsey, and J. D. Johnson, "Global optimization for artificial neural networks: A tabu search application," *European Journal of Operational Research*, vol. 106, no. 2-3, pp. 570–584, 1998.

[140] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Cognitive Modeling*, vol. 5, no. 3, p. 1, 1988.

[141] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv Preprint arXiv:1412.6980*, 2014.

[142] T. Dozat, "Incorporating nesterov momentum into adam," tech. rep., Stanford University, Tech. Rep., 2015.[Online]. Available: http://cs229. stanford. edu/proj2015/054 report. pdf, 2015.

[143] I. Sutskever, J. Martens, G. E. Dahl, and G. E. Hinton, "On the importance of initialization and momentum in deep learning.," *ICML (3)*, vol. 28, pp. 1139–1147, 2013.

[144] T. Tieleman and G. Hinton, "Rmsprop: Divide the gradient by a running average of its recent magnitude. coursera: Neural networks for machine learning," tech. rep., Technical report, 2012. 31.

[145] R. E. Dorsey, J. D. Johnson, and W. J. Mayer, "A genetic algorithm for the training of feedforward neural networks," *Advances in Artificial Intelligence in Economics, Finance and Management*, vol. 1, pp. 93–111, 1994.

[146] A. Corana, M. Marchesi, C. Martini, and S. Ridella, "Minimizing multimodal functions of continuous variables with the simulated annealing algorithm corrigenda for this article is available here," *Mathematical Software, ACM Transactions on (TOMS)*, vol. 13, no. 3, pp. 262–280, 1987.

[147] Y. He, Y. Qiu, G. Liu, and K. Lei, "Optimizing weights of neural network using an adaptive tabu search approach," in *Neural Networks, International Symposium on*, pp. 672–676, Springer, 2005.

[148] G.-y. Liu, Y. He, Y. Fang, and Y. Qiu, "A novel adaptive search strategy of intensification and diversification in tabu search," in *Neural Networks and Signal Processing, 2003. Proceedings of the 2003 International Conference on*, vol. 1, pp. 428–431, IEEE, 2003.

[149] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, no. Feb, pp. 281–305, 2012.

[150] H. N. Mhaskar and C. A. Micchelli, "How to choose an activation function," *Advances in Neural Information Processing Systems*, pp. 319–319, 1994.

[151] C. Dugas, Y. Bengio, F. Bélisle, C. Nadeau, and R. Garcia, "Incorporating second-order functional knowledge for better option pricing," *Advances in Neural Information Processing Systems*, pp. 472–478, 2001.

[152] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Machine Learning, Proceedings of the 27th International Conference on (ICML-10)*, pp. 807–814, 2010.

[153] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. ICML*, vol. 30, 2013.

[154] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Computer Vision, Proceedings of the IEEE International Conference on*, pp. 1026–1034, 2015.

[155] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. C. Courville, and Y. Bengio, "Maxout networks.," *ICML (3)*, vol. 28, pp. 1319–1327, 2013.

[156] A. Berger and J. Lafferty, "Information retrieval as statistical translation," in *Research and Development in Information Retrieval, Proceedings of the 22nd Annual International ACM SIGIR Conference on*, pp. 222–229, ACM, 1999.

[157] V. Lavrenko and W. B. Croft, "Relevance based language models," in *Research and Development in Information Retrieval, Proceedings of the 24th Annual International ACM SIGIR Conference on*, pp. 120–127, ACM, 2001.

[158] X. Li and W. Croft, "Improving the robustness of relevance-based language models," tech. rep., Technical Report IR-401, Center for Intelligent Information Retrieval. University of Massachusetts, 2005.

[159] C. D. Manning, P. Raghavan, H. Schütze, *et al.*, *Introduction to Information Retrieval*, vol. 1. Cambridge university press Cambridge, 2008.

[160] V. Metsis, I. Androutsopoulos, and G. Paliouras, "Spam filtering with naive bayes-which naive bayes?," in *CEAS*, vol. 17, pp. 28–69, 2006.

[161] A. McCallum, K. Nigam, *et al.*, "A comparison of event models for naive bayes text classification," in *Learning for Text Categorization, AAAI-98 workshop on*, vol. 752, pp. 41–48, Citeseer, 1998.

[162] J. D. Rennie, L. Shih, J. Teevan, and D. R. Karger, "Tackling the poor assumptions of naive bayes text classifiers," in *Machine Learning, Proceedings of the 20th International Conference on (ICML-03)*, pp. 616–623, 2003.

[163] W. Dai, G.-R. Xue, Q. Yang, and Y. Yu, "Transferring naive bayes classifiers for text classification," in *AAAI*, vol. 7, pp. 540–545, 2007.

[164] J. Su, J. S. Shirab, and S. Matwin, "Large scale text classification using semi-supervised multinomial naive bayes," in *Machine Learning, Proceedings of the 28th International Conference on (ICML-11)*, pp. 97–104, 2011.

[165] F. G. Cozman, I. Cohen, M. C. Cirelo, *et al.*, "Semi-supervised learning of mixture models," in *ICML*, pp. 99–106, 2003.

[166] K. Nigam, A. McCallum, and T. Mitchell, "Semi-supervised text classification using em," *Semi-Supervised Learning*, pp. 33–56, 2006.

[167] H. Zhang, "The optimality of naive bayes," *AA*, vol. 1, no. 2, p. 3, 2004.

[168] C. Zhai and J. Lafferty, "A study of smoothing methods for language models applied to ad hoc information retrieval," in *Research and Development in Information Retrieval, Proceedings of the 24th annual international ACM SIGIR Conference on*, pp. 334–342, ACM, 2001.

[169] R. Nallapati, "Semantic language models for topic detection and tracking," in *Human Language Technology, Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on: Proceedings of the HLT-NAACL 2003 Student Research Workshop-Volume 3*, pp. 1–6, Association for Computational Linguistics, 2003.

[170] R. Agrawal, R. Bayardo, and R. Srikant, "Athena: Mining-based interactive management of text databases," in *Extending Database Technology, International Conference on* , pp. 365–379, Springer, 2000.

[171] D. A. Field, "Laplacian smoothing and delaunay triangulations," *International Journal for Numerical Methods in Biomedical Engineering*, vol. 4, no. 6, pp. 709–712, 1988.

[172] M. Federico, N. Bertoldi, and M. Cettolo, "Irstlm: an open source toolkit for handling large scale language models.," in *Interspeech*, pp. 1618–1621, 2008.

[173] C. Zhai and J. Lafferty, "Two-stage language models for information retrieval," in *Research and Development in Information Retrieval, Proceedings of the 25th annual international ACM SIGIR Conference on*, pp. 49–56, ACM, 2002.

[174] H. Song, L. Wang, B. Li, and X. Liu, "New trending events detection based on the multi-representation index tree clustering," *International Journal of Intelligent Systems and Applications*, vol. 3, no. 3, p. 26, 2011.

[175] G. Luo, C. Tang, and P. S. Yu, "Resource-adaptive real-time new event detection," in *Management of Data, Proceedings of the 2007 ACM SIGMOD international conference on*, pp. 497–508, ACM, 2007.

[176] W. Huang, T. Wang, W. Chen, and Y. Wang, "Category-level transfer learning from knowledge base to microblog stream for accurate event detection," in *Database Systems for Advanced Applications, International Conference on*, pp. 50–67, Springer, 2017.

[177] C. A. Gomez-Uribe and N. Hunt, "The netflix recommender system: Algorithms, business value, and innovation," *Management Information Systems, ACM Transactions on (TMIS)*, vol. 6, no. 4, p. 13, 2016.

[178] S. Feyer, S. Siebert, B. Gipp, A. Aizawa, and J. Beel, "Integration of the scientific recommender system mr. dlib into the reference manager jabref," in *Information Retrieval, European Conference on*, pp. 770–774, Springer, 2017.

[179] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Application of dimensionality reduction in recommender system-a case study," tech. rep., DTIC Document, 2000.

[180] J. Davidson, B. Liebald, J. Liu, P. Nandy, T. Van Vleet, U. Gargi, S. Gupta, Y. He, M. Lambert, B. Livingston, *et al.*, "The youtube video recommendation system," in *Recommender Systems, Proceedings of the fourth ACM conference on*, pp. 293–296, ACM, 2010.

[181] G. Adomavicius and A. Tuzhilin, "Context-aware recommender systems," in *Recommender Systems Handbook*, pp. 191–226, Springer, 2015.

[182] K. Oku, S. Nakajima, J. Miyazaki, and S. Uemura, "Context-aware svm for context-dependent information recommendation," in *Mobile Data Management, 2006. MDM 2006. 7th International Conference on*, pp. 109–109, IEEE, 2006.

[183] Z. Yu, X. Zhou, D. Zhang, C.-Y. Chin, X. Wang, and J. Men, "Supporting context-aware media recommendations for smart phones," *IEEE Pervasive Computing*, vol. 5, no. 3, pp. 68–75, 2006.

[184] N. Hariri, B. Mobasher, and R. Burke, "Query-driven context aware recommendation," in *Recommender Systems, Proceedings of the 7th ACM Conference on*, pp. 9–16, ACM, 2013.

[185] S. Abbar, M. Bouzeghoub, and S. Lopez, "Context-aware recommender systems: A service-oriented approach," in *VLDB PersDB workshop*, pp. 1–6, 2009.

[186] Y. Park, S. Park, W. Jung, and S.-g. Lee, "Reversed cf: A fast collaborative filtering algorithm using a k-nearest neighbor graph," *Expert Systems with Applications*, vol. 42, no. 8, pp. 4022–4028, 2015.

[187] Y. Qin, D. Wurzer, V. Lavrenko, and C. Tang, "Counteracting novelty decay in first story detection," in *Information Retrieval, European Conference on*, pp. 555–560, Springer, 2017.

[188] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," in *Computer Vision, European Conference on*, pp. 43–58, Springer, 1996.

[189] D. Wurzer, V. Lavrenko, and M. Osborne, "Twitter-scale new event detection via k-term hashing.," in *EMNLP*, pp. 2584–2589, 2015.

[190] C.-J. Lin, "Projected gradient methods for nonnegative matrix factorization," *Neural Computation*, vol. 19, no. 10, pp. 2756–2779, 2007.

[191] P. N. Belhumeur and D. J. Kriegman, "What is the set of images of an object under all possible illumination conditions?," *International Journal of Computer Vision*, vol. 28, no. 3, pp. 245–260, 1998.

[192] P. Indyk and R. Motwani, "Approximate nearest neighbors: towards removing the curse of dimensionality," in *Theory of Computing, Proceedings of the 13th annual ACM Symposium on*, pp. 604–613, ACM, 1998.

[193] I. S. Gradshteyn and I. M. Ryzhik, *Table of Integrals, Series, and Products*. Academic press, 2014.

[194] E. Abbena, S. Salamon, and A. Gray, *Modern differential geometry of curves and surfaces with Mathematica*. CRC press, 2006.

[195] M. M. Deza and E. Deza, "Encyclopedia of distances," in *Encyclopedia of Distances*, pp. 1–583, Springer, 2009.

[196] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge university press, 2012.

[197] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and other Kernel-based Learning Methods*. Cambridge university press, 2000.

[198] C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines," *Intelligent Systems and Technology, ACM Transactions on (TIST)*, vol. 2, no. 3, p. 27, 2011.

[199] A. Rosenberg and J. Hirschberg, "V-measure: A conditional entropy-based external cluster evaluation measure.," in *EMNLP-CoNLL*, vol. 7, pp. 410–420, 2007.

[200] R. Y. Rubinstein and D. P. Kroese, *The Cross-entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation and Machine Learning*. Springer Science & Business Media, 2013.

[201] J. Shore and R. Johnson, "Properties of cross-entropy minimization," *Information Theory, IEEE Transactions on*, vol. 27, no. 4, pp. 472–482, 1981.

[202] J. Shore and R. Johnson, "Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy," *Information Theory, IEEE Transactions on*, vol. 26, no. 1, pp. 26–37, 1980.

[203] R. Manmatha, A. Feng, and J. Allan, "A critical examination of tdt's cost function," in *Research and Development in Information Retrieval, Proceedings*

of the 25th Annual International ACM SIGIR Conference on, pp. 403–404, ACM, 2002.

[204] P. A. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach*, vol. 761. Prentice-Hall London, 1982.

[205] L. Van Der Maaten, "Accelerating t-sne using tree-based algorithms.," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 3221–3245, 2014.

[206] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[207] L. Van Der Maaten, "Barnes-hut-sne," *arXiv Preprint arXiv:1301.3342*, 2013.

[208] J. D. Gibbons and S. Chakraborti, *Nonparametric Statistical Inference*. Springer, 2011.

[209] P. E. McKnight and J. Najab, "Mann-whitney u test," *Corsini Encyclopedia of Psychology*, 2010.

[210] G. D. Ruxton, "The unequal variance t-test is an underused alternative to student's t-test and the mann–whitney u test," *Behavioral Ecology*, vol. 17, no. 4, pp. 688–690, 2006.

[211] W. J. Dixon and A. M. Mood, "The statistical sign test," *Journal of the American Statistical Association*, vol. 41, no. 236, pp. 557–566, 1946.

[212] W. J. Dixon, F. J. Massey, *et al.*, *Introduction to Statistical Analysis*, vol. 344. McGraw-Hill New York, 1969.

[213] T. A. Almeida, J. M. G. Hidalgo, and A. Yamakami, "Contributions to the study of sms spam filtering: new collection and results," in *Document Engineering, Proceedings of the 11th ACM symposium on*, pp. 259–262, ACM, 2011.

[214] C. Tagg, *A corpus linguistics study of SMS text messaging.* PhD thesis, The University of Birmingham, 2009.

[215] B. Tang, H. He, P. M. Baggenstoss, and S. Kay, "A bayesian classification approach using class-specific features for text categorization," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 28, no. 6, pp. 1602–1606, 2016.

[216] A. M. Dai and Q. V. Le, "Semi-supervised sequence learning," in *Advances in Neural Information Processing Systems*, pp. 3079–3087, 2015.

[217] P. Willett, "The porter stemming algorithm: then and now," *Program*, vol. 40, no. 3, pp. 219–223, 2006.

[218] O. Pele and M. Werman, "A linear time histogram metric for improved sift matching," *Computer Vision–ECCV 2008*, pp. 495–508, 2008.

[219] S. Godbole and S. Sarawagi, "Discriminative methods for multi-labeled classification," in *Knowledge Discovery and Data Mining, Pacific-Asia Conference on*, pp. 22–30, Springer, 2004.

[220] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[221] K. Lang, "Newsweeder: Learning to filter netnews," in *Machine Learning, Proceedings of the 12th International Conference on*, pp. 331–339, 1995.

[222] W. N. Francis and H. Kucera, "The brown corpus: A standard corpus of present-day edited american english," *Providence, RI: Department of Linguistics, Brown University [producer and distributor]*, 1979.

[223] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting.," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[224] T.-F. Wu, C.-J. Lin, and R. C. Weng, "Probability estimates for multi-class classification by pairwise coupling," *Journal of Machine Learning Research*, vol. 5, no. Aug, pp. 975–1005, 2004.

[225] S. Lloyd, "Least squares quantization in pcm," *Information Theory, IEEE Transactions on*, vol. 28, no. 2, pp. 129–137, 1982.

[226] B. Frénay and M. Verleysen, "Classification in the presence of label noise: a survey," *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 25, no. 5, pp. 845–869, 2014.

[227] T. Hastie and R. Tibshirani, "Discriminant adaptive nearest neighbor classification," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 18, no. 6, pp. 607–616, 1996.

[228] J. Gao and L. Xu, "A novel spatial analysis method for remote sensing image classification," *Neural Processing Letters*, vol. 43, no. 3, pp. 805–821, 2016.

[229] M. Cliche, "Bb_twtr at semeval-2017 task 4: Twitter sentiment analysis with cnns and lstms," *arXiv Preprint arXiv:1704.06125*, 2017.

[230] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 39, no. 4, pp. 640–651, 2017.

[231] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Computer Vision and Pattern Recognition, Proceedings of the IEEE Conference on* , pp. 3431–3440, 2015.

[232] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Artificial Intelligence and Statistics,Proceedings of the 13th International Conference on*, pp. 249–256, 2010.

[233] T. Robinson, M. Hochberg, and S. Renals, "The use of recurrent neural networks in continuous speech recognition," in *Automatic Speech and Speaker Recognition*, pp. 233–258, Springer, 1996.

[234] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, no. Aug, pp. 1871–1874, 2008.

[235] A. Gautam and P. Bedi, "Developing content-based recommender system using hadoop map reduce," *Journal of Intelligent & Fuzzy Systems*, vol. 32, no. 4, pp. 2997–3008, 2017.

[236] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems*, pp. 3104–3112, 2014.

[237] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of NAACL-HLT*, pp. 1480–1489, 2016.

[238] R. Johnson and T. Zhang, "Supervised and semi-supervised text categorization using lstm for region embeddings," *arXiv Preprint arXiv:1602.02373*, 2016.

[239] N. Kalchbrenner, I. Danihelka, and A. Graves, "Grid long short-term memory," *arXiv Preprint arXiv:1507.01526*, 2015.

[240] L. Verwimp, J. Pelemans, P. Wambacq, *et al.*, "Character-word lstm language models," *arXiv Preprint arXiv:1704.02813*, 2017.