New Weighting Schemes for Document Ranking and Ranked Query Suggestion

Suthira Plansangket

A thesis submitted for the degree of Doctor of Philosophy

School of Computer Science and Electronic Engineering

University of Essex

April 2017

Abstract

Term weighting is a process of scoring and ranking a term's relevance to a user's information need or the importance of a term to a document. This thesis aims to investigate novel term weighting methods with applications in document representation for text classification, web document ranking, and ranked query suggestion. Firstly, this research proposes a new feature for document representation under the vector space model (VSM) framework, i.e., class specific document frequency (CSDF), which leads to a new term weighting scheme based on term frequency (TF) and the newly proposed feature. The experimental results show that the proposed methods, CSDF and TF-CSDF, improve the performance of document classification in comparison with other widely used VSM document representations. Secondly, a new ranking method called GCrank is proposed for re-ranking web documents returned from search engines using document classification scores. The experimental results show that the GCrank method can improve the performance of web returned document ranking in terms of several commonly used evaluation criteria. Finally, this research investigates several state-of-the-art ranked retrieval methods, adapts and combines them as well, leading to a new method called Tfjac for ranked query suggestion, which is based on the combination between TF-IDF and Jaccard coefficient methods. The experimental results show that Tfjac is the best method for query suggestion among the methods evaluated. It outperforms the most popularly used TF-IDF method in terms of increasing the number of highly relevant query suggestions.

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Professor John Q. Gan, who never tires of helping and guiding me researching. I wish to sincerely thank him for his patience and understanding, without his guidance and persistent help, this thesis would not have been successfully completed.

I would like to sincerely thank my supervisory board members: Dr. Francisco Sepulveda, Professor Qingfu Zhang, Professor Klaus McDonald-Maier, and Professor Edward Tsang, for their helpful suggestions.

Thanks also go to my office mates and friends who never hesitated to help and support my PhD study.

I would like to thank my original affiliation, Prince of Songkla University, for funding me the doctoral scholarship.

Finally, I would like to thank my beloved family for their love and support throughout my life.

Contents

Abstract II

Acknowledgements III

Contents IV

List of figures VIII

List of tables X

List of equations XIII

Chapter 1 Introduction 1

1.1 Motivation 1

1.2 Research objectives 3

1.3 Thesis contributions 3

1.4 Thesis organisation 4

Chapter 2 Literature review 7

2.1 Document representation 7

2.1.1 Vector space model (VSM) 9

2.1.2 Graph-based model 11

2.2 Term weighting 14

2.2.1 Statistical term weighting 15

2.2.2 Semantic term weighting 17

2.2.3 Term weighting through machine learning 19

2.3 Document or text classification 20

2.4 Classifier/decision fusion 22

2.5 Document ranking criteria.....25

2.5.1 Content-based ranking 26

2.5.2 Hyperlink-based ranking or connectivity-based ranking.....29

2.5.3 Hyperlink-content-based ranking.....31

2.6 Query suggestion.....31

2.6.1 Features for query suggestion 34

2.6.2 Methods for query suggestion.....39

2.7 Evaluation methods 43

2.8 Summary 48

Chapter 3 CSDF and semantic information for VSM-based document representation and classification 51

3.1 Introduction 51

3.2 Baseline document representation methods 52

3.3 Term frequency relevance frequency (TFRF).....53

3.4 Class specific document frequency (CSDF) 54

3.5 Semantic information for VSM-based document representation and classification.....56

3.5.1 Semantic representation 57

3.5.2 Class prediction using semantic information.....59

3.6 Classifier fusion.....61

3.7 Experiments and results 62

3.7.1 Experimental procedure 62

3.7.2 Experimental results.....71

3.8 Summary 97

Chapter 4 GCrank: A new ranking method using document classification scores 99

4.1 Introduction 99

4.2 Document ranking criteria.....100

4.2.1 Google ranking.....100

4.2.2 The proposed ranking method: GCrank.....101

4.3 Experiments and results 103

4.3.1 Experimental procedure 103

4.3.2 Selection of a weighting factor (α).....104

4.3.3 Experimental results and evaluation 106

4.4 Summary 112

Chapter 5 A hybrid method for term ranking and its applications in automatic query suggestion 114

5.1 Introduction 114

5.2 Query suggestion methods 115

5.2.1 TF-IDF 116

5.2.2 Jaccard coefficient.....116

5.2.3 Cosine similarity 117

5.2.4 A new method based on the combination of TF-IDF and Jaccard coefficient.....118

5.2.5 Using LDA classification scores for ranking query suggestions ...121

5.3 Evaluation methods ...122

5.4 Experiments and results ...123

5.4.1 Experimental design ...123

5.4.2 User's selection of suggested queries and assessment of relevance of search results ...124

5.4.3 Experimental results and evaluation ...126

5.5 Summary ...135

Chapter 6 Conclusion ...137

6.1 Summary of contributions ...137

6.2 Limitations and future work ...140

References ...142

Appendix A ...169

List of figures

Figure 1.1 The overall thesis contributions and related application domains 6

Figure 2.1 Hierarchy of grammatical units [19] 8

Figure 2.2 An example of standard graph document representation [21].....12

Figure 2.3 Dasarathy's classification [78] 23

Figure 2.4 An example of query suggestion and reformulation on Google.....33

Figure 2.5 The overall process of generating query suggestions [10] 34

Figure 3.1 An example of the relationship between hyponyms and hypernym [161].....58

Figure 3.2 The prediction process on test set.....61

Figure 3.3 F-measure scores of TF-based document representation 67

Figure 3.4 F-measure scores of TP-based document representation 67

Figure 3.5 Classification accuracy with different number of features on 20newsgroups 68

Figure 3.6 Classification accuracy with different types of features on Reuters dataset 72

Figure 3.7 Experimental results of TF-CSDF 77

Figure 3.8 Classification accuracy with different types of features on 20-class 20newsgroup dataset.....80

Figure 3.9 Classification accuracy with different types of features on 7-class 20newsgroup dataset.....81

Figure 3.10 The classification accuracy of two-fold CV on 20newsgroups (20 classes).....86

Figure 3.11 The classification accuracy of two-fold CV on 20newsgroups (7 classes).....86

Figure 3.12 The training and testing accuracy with 20 classes using IG score 0.01 88

Figure 3.13 The training and testing accuracy with 7 classes using IG score 0.01 89

Figure 3.14 The classification accuracy of TF-CSDF 90

Figure 3.15 The classification accuracy on web returned documents 93

Figure 4.1 Average results with different weighting factors 105

Figure 4.2 Classification accuracy of the LDA classifier for each category 107

Figure 5.1 Diagram of the Tfjac method.....120

Figure 5.2 Experimental results of the Tfjac experiment 127

Figure 5.3 Integrated evaluation in MRR scores 128

Figure 5.4 User evaluation in MRR scores.....129

Figure 5.5 Experimental results of GCrank experiment 131

Figure 5.6 Integrated evaluation in average MRR scores.....132

Figure 5.7 Additional results.....133

Figure 5.8 Experimental results of QS experiment.....134

List of tables

Table 2.1 Comparison of the sources of features.....38

Table 2.2 Comparisons of the reviewed models.....42

Table 2.3 The contingency table.....43

Table 3.1 Five representative words of each class.....60

Table 3.2 Training and testing datasets of Reuters-21578.....62

Table 3.3 Training and testing datasets of 20newsgroups.....63

Table 3.4 Categories of queries.....64

Table 3.5 The number of web returned documents in training and testing datasets.....64

Table 3.6 The classification accuracy of using different parts of speech.....65

Table 3.7 F-measure scores with different number of features on Reuters dataset.....66

Table 3.8 Classification accuracy with different number of features on 20newsgroups.....68

Table 3.9 Representative words of each class of Reuters dataset.....70

Table 3.10 Representative words of each class of 20newsgroups dataset.....70

Table 3.11 Representative words of each class of web document dataset.....70

Table 3.12 Experimental results on Reuters dataset (I).....71

Table 3.13 Experimental results on Reuters dataset (II).....72

Table 3.14 F_1 scores of kNN classifier on Reuters dataset.....73

Table 3.15 F_1 scores of LDA classifier on Reuters dataset.....74

Table 3.16 F_1 scores of SVM classifier on Reuters dataset.....74

Table 3.17 F_1 scores of Naïve Bayes classifier on Reuters dataset.....75

Table 3.18 F_1 scores of logistic regression classifier on Reuters dataset.....75

Table 3.19 Experimental results of semantic representation 77

Table 3.20 An example of classifier fusion 79

Table 3.21 The classification accuracy on 20newsgroups 80

Table 3.22 F_1 scores of SVM classifier on 20newsgroups (7 classes).....81

Table 3.23 F_1 scores of SVM classifier on 20newsgroups (20 classes).....82

Table 3.24 F_1 scores of LDA classifier on 20newsgroups (7 classes) 83

Table 3.25 F_1 scores of LDA classifier on 20newsgroups (20 classes) 83

Table 3.26 F_1 scores of Naïve Bayes classifier on 20newsgroups (7 classes).....84

Table 3.27 F_1 scores of Naïve Bayes classifier on 20newsgroups (20 classes)....84

Table 3.28 The results of two-fold CV on 20newsgroups (20 classes) 85

Table 3.29 The results of two-fold CV on 20newsgroups (7 classes) 86

Table 3.30 Training accuracy with 20 classes 87

Table 3.31 Testing accuracy with 20 classes 87

Table 3.32 Training accuracy with 7 classes 88

Table 3.33 Testing accuracy with 7 classes 88

Table 3.34 Experimental results of TF-CSDF 90

Table 3.35 Experimental results on web returned documents (I) 92

Table 3.36 Experimental results on web returned documents (II).....92

Table 3.37 Experimental results on web returned documents (III).....92

Table 3.38 The classification accuracy on web returned documents.....92

Table 3.39 F_1 scores of LDA classifier on web returned documents.....94

Table 3.40 F_1 scores of LDA classifier on web returned documents (wrapper)...95

Table 3.41 F_1 scores of Naïve Bayes classifier on web returned documents.....95

Table 3.42 F_1 scores of Naïve Bayes classifier on web returned documents (wrapper).....96

Table 3.43 F_1 scores of logistic regression classifier on web returned documents 96

Table 3.44 F_1 scores of logistic regression classifier on web returned documents (wrapper).....97

Table 4.1 Experimental results of using different weighting factor values 105

Table 4.2 Classification accuracy of LDA classifier 106

Table 4.3 Classification accuracy of LDA classifier for each category.....106

Table 4.4 Evaluation results of the original Google ranking 108

Table 4.5 Evaluation results of the GCrank method.....108

Table 4.6 Statistical significance test results: GCrank vs. Google ranking 109

Table 4.7 Examples of re-ranking using GCrank (I) 111

Table 4.8 Examples of re-ranking using GCrank (II) 112

Table 5.1 Query suggestion methods to be investigated.....118

Table 5.2 Experimental results of the Tfjac experiment.....127

Table 5.3 Statistical test results of the Tfjac experiment 127

Table 5.4 Summary of evaluation results.....128

Table 5.5 Integrated evaluation in MRR scores.....128

Table 5.6 Statistical test results of integrated evaluation in MRR scores.....129

Table 5.7 User evaluation in MRR scores 129

Table 5.8 Experimental results of GCrank experiment 130

Table 5.9 Summary of evaluation results.....131

Table 5.10 Integrated evaluation in MRR scores.....132

Table 5.11 Additional results.....133

Table 5.12 Experimental results of QS experiment 134

Table 5.13 Statistical test results of QS experiment 134

List of equations

Equation	(Chapter.No)	Page
$TP_{ji} = \begin{cases} 1, \text{if term i is in document j} \\ 0, \text{otherwise} \end{cases}$	(2.1)	16
$TFIDF_{ji} = \left(0.5 + 0.5\,\frac{TF_{ji}}{\max TF_j}\right) \times \log_2 \frac{N}{DF_i}$	(2.2)	16
$TFIDF_{ji} = \begin{cases} (1 + \log_2 TF_{ji}) \times \log_2 \frac{N}{DF_i}, & \text{if } TF_{ji} > 0 \\ 0, & \text{otherwise} \end{cases}$	(2.3)	16
$TFIDF_i = \sum_{j=1}^{N} w_{i,j}$	(5.1)	116
$w_{i,j} = \begin{cases} (1 + \log_2 TF_{i,j}) \times \log_2 \frac{N}{DF_i}, & \text{if } TF_{i,j} > 0 \\ 0, & \text{otherwise} \end{cases}$	(5.2)	116
$\cos(\vec{q}, \vec{d}) = \vec{q}.\vec{d} = \sum_{i=1}^{\lvert v \rvert} q_i d_i$	(2.4)	27
$\cos(\vec{q}, \vec{s}) = \vec{q}.\vec{s} = \sum_{i=1}^{B} q_i s_i$	(5.5)	117
$Jaccard(A, B) = \frac{\lvert A \cap B \rvert}{\lvert A \cup B \rvert}$	(2.5)	27
$Jaccard(D_1, D_2) = \frac{\lvert D_1 \cap D_2 \rvert}{\lvert D_1 \cup D_2 \rvert}$	(5.3)	116
$Jaccard(D_1, D_2, \dots D_M) = \frac{\lvert D_1 \cap D_2 \cap \dots D_M \rvert}{\lvert D_1 \cup D_2 \cup \dots D_M \rvert}$	(5.4)	117
$PR(a) = \frac{q}{T} + (1 - q) \sum_{i=1}^{n} \frac{PR(p_i)}{L(p_i)}$	(2.6)	30
$R(P, Q) = \alpha BM25(P, Q) + (1 - \alpha) PR(P)$	(2.7)	31
$Precision = \frac{tp}{tp + fp}$	(2.8)	43
$Recall = \frac{tp}{tp + fn}$	(2.9)	43
$P@10 = \frac{number\ of\ relevant\ suggestions\ among\ top\ 10}{10}$	(2.10)	44
$P@20 = \frac{number\ of\ relevant\ suggestions\ among\ top\ 20}{20}$	(2.11)	44
$F(j) = \frac{2}{\frac{1}{r(j)} + \frac{1}{P(j)}}$	(2.12)	44

Equation	(Chapter.No)	Page
$Accuracy = \frac{tp + tn}{tp + fp + fn + tn}$	(2.13)	44
$RR_{ji} = \frac{1}{r_{ji}}$	(2.14)	45
$MRR = \frac{1}{q}\sum_{j=1}^{q}\frac{1}{Q_j}\sum_{i=1}^{Q_j} RR_{ji}$	(2.15)	45
$P_{ji} = \frac{number\ of\ relevant\ suggestions}{number\ of\ suggestions\ examined} = \frac{i}{r_{ji}}$	(2.16)	46
$MAP = \frac{1}{q}\sum_{j=1}^{q}\frac{1}{Q_j}\sum_{i=1}^{Q_j} P_{ji}$	(2.17)	46
$CG_j = w_1 + w_2 + \cdots w_{Q_j}$	(2.18)	46
$DCG_j = w_1 + \frac{w_2}{\log_2 2} + \frac{w_3}{\log_2 3} + \ldots \frac{w_{Q_j}}{\log_2 Q_j}$	(2.19)	47
$nDCG_j = \frac{DCG_j}{IDCG}$	(2.20)	47
$DCG = \frac{1}{q}\sum_{j=1}^{q} DCG_j$	(2.21)	47
$nDCG = \frac{1}{q}\sum_{j=1}^{q} nDCG_j$	(2.22)	47
$norTF_{ji} = \frac{TF_{ji}}{\max TF_j}$, $0 < norTF_{ji} < 1$	(3.1)	52
$norTFIDF_{ji} = \frac{TFIDF_{ji}}{\max TFIDF_j}$, $0 < norTFIDF_{ji} < 1$	(3.2)	52
$RF = \log(2 + \frac{a}{\max(1,c)})$	(3.3)	53
$TF.RF = TF * RF$	(3.4)	53
$TF.RF_{ik} = TF_i * \log_2(2 + \frac{DF_{ik}}{\max(1, DF_i - DF_{ik})})$	(3.5)	53
$TF.RF_i = \mathrm{var}(TF.RF_{ik})$	(3.6)	53
$CSDF_{ik} = \begin{cases} \frac{DF_{ik}/N_k}{(DF_i - DF_{ik})/(N - N_k) + 1}, & \text{if } TF_{ik} > 0 \\ 0, & \text{otherwise} \end{cases}$	(3.7)	54
$CSDF_i = \mathrm{var}(CSDF_{ik})$	(3.8)	56
$TF - CSDF_{ji} = \propto CSDF_i + (1-\propto) norTF_{ji}$, $0 < \propto < 1$	(3.9)	56
$Semantic_{ik} = \max(path_{sim_{k1}}, path_{sim_{k2}}, path_{sim_{k3}})$	(3.10)	59
$Semantic_i = \mathrm{var}(Semantic_{ik})$	(3.11)	59

Equation	(Chapter.No)	Page
$\Sigma = (n_1\Sigma_1 + n_2\Sigma_2)/n$	(4.1)	101
$\mathbf{w} = \Sigma^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$	(4.2)	101
$w_0 = \mathbf{w}^T(n_1\boldsymbol{\mu}_1 + n_2\boldsymbol{\mu}_2)/n$	(4.3)	101
$Cscore = \mathbf{w}^T\mathbf{x} - w_0$	(4.4)	101
$norGscore_j = \frac{1}{GoogleRank_j}$, $0 \le norGscore_j \le 1$	(4.5) (5.6)	102 121
$norCscore_j = \begin{cases} \frac{Cscore_j}{\text{Max}Cscore}, & 0 \le norCscore_j \le 1 \\ 0, & \text{if document j is not in the same topic category as the query} \end{cases}$	(4.6) (5.7)	102 121
$GCrank_j = \alpha \times norGscore_j + (1-\alpha) \times norCscore_j$	(4.7)	102
$GCrank_j = \begin{cases} \alpha \times norGscore_j + (1-\alpha) \times norCscor_j \\ 0, \quad \text{if document j is not in the same topic category as the query} \end{cases}$	(5.8)	121
$sGCrank_i = \sum_{j=1}^{n} GCrank_{ij}$	(5.9)	122
$mGCrank_i = \sum_{j=1}^{n} (GCrank_{ij} \times TF_{ij})$	(5.10)	122

Chapter 1 Introduction

1.1 Motivation

It is a challenge to provide users with relevant information quickly from the available large amount of document data. A search engine is an information retrieval (IR) system designed to help find useful information from databases or the Internet. Traditional documents and web documents are different. Things that work well on the benchmark documents often do not produce good results on the web. Web documents have extreme variation from normal documents in terms of the language, vocabulary, type or format, and whether or not it is machine generated. Furthermore, web documents are unprecedented in scale [1]; therefore, web search is different and generally far harder than searching traditional documents [2].

In information retrieval, one of the great challenges faced by search engines is to precisely understand users' need, since users usually submit a very short (only one to three words) and imprecise query [3]. Most existing search engines retrieve information by finding exact keywords. Sometimes, users do not know the precise vocabulary of the topic to be searched and they do not know how search algorithms work so as to produce proper queries [4]. To deal with these problems, a huge amount of documents need to be categorised into different categories. After that, these documents should be ranked where the most relevant document that a user needs should appear first. Term weighting techniques help us distinguish between important and unimportant terms in a document. It is helpful in classifying, matching the documents to the correct categories, and ranking these documents given to users.

Text or document classification has been involved in IR [5]. To automatically organise documents into topic groups, document classification has been widely applied for this purpose. Apart from their applications in search engines, document

classification techniques have been applied to other areas such as spam filtering [6], email routing [7], and genre classification [8]. The content of a document can be represented as a collection of terms: words, stems, phrases, or other units derived from the text of the document. These terms are usually weighted to indicate their importance within the document [9]. This is called document representation. The representation of a set of documents as vectors in a common vector space is known as the vector space model (VSM) and is fundamental in scoring documents on a query and document classification [2]. A main problem in text categorisation is how to improve the classification accuracy. Most of the research on text classification has focused on introducing various machine learning methods rather than discussing particular features of text documents relevant to classification [2].

With regard to the ranking problem, previous research [10] has shown that almost 80% of the users who use search engines are interested in only the top 3 returned results. He and Ounis [11] proposed an entropy measure which estimates how the occurrences of a query term spread over returned documents. The higher the entropy is, the more a returned document is related to the query. Their results show that the entropy in the top 5 returned documents is very high, and it decreases rapidly in the remaining documents. It means that if only the top 5 web returned documents are relevant to the user's query and they are not properly ranked, the user may miss the relevant information. Therefore, it is very important to develop effective document ranking algorithms.

Another solution to the IR problem is to devise a query suggestion section in search engines. Diane Kelly et al. [12] investigated the effects of popularity and quality on the usage of query suggestion. The results show that query suggestions are helpful when users run out of ideas or faced a cold-start issue. Therefore, query

suggestion is a useful tool that helps users in their searching activities; for instance, it can help users to specify their information needs more accurately.

1.2 Research objectives

This Ph.D. research aims to investigate novel approaches, including new methods for weighting or scoring terms and documents, to improve the performance of document ranking and ranked query suggestion. In particular, it focuses on three major objectives:

- To improve the performance of document classification by using features sensitive to class memberships, a new term weighting technique is to be proposed.
- To improve the performance of web returned document ranking and user's satisfaction, a new ranking method is to be proposed.
- To improve the performance of ranked query suggestion, the state-of-the-art ranked retrieval methods are to be investigated, adapted, and combined for effective query suggestion.

1.3 Thesis contributions

Three major contributions of this thesis work can be summarised as follows:

- This research proposes a new feature for document representation under the VSM framework, i.e., class specific document frequency (CSDF), which leads to a novel term weighting scheme based on term frequency (TF) and the newly proposed feature. The experimental results show that the proposed methods, CSDF and TF-CSDF, improve the performance of document classification in comparison with other widely used VSM document representations.
- A new ranking method called GCrank for re-ranking search engine returned web documents by making use of document classification scores is proposed. The

experimental results show that GCrank can improve the original Google document ranking in terms of comparing with human participants ranking performance using the following criteria: mean average precision (MAP), and discounted cumulated gain (DCG), and Precision @10.

- A new method called Tfjac for query suggestion is proposed, which combines term frequency and inverse document frequency (TF-IDF) and Jaccard coefficient in an effective manner. The experimental results show that Tfjac is the best method for query suggestion among the methods evaluated. It outperforms the most popularly used TF-IDF method in terms of increasing the relevance of the query suggestions or the number of highly relevant query suggestions.

Figure 1.1 illustrates the overall thesis contributions and related application domains. In addition, some of the original content used in this thesis has been published in the peer-reviewed paper. These papers are detailed in Appendix A.

1.4 Thesis organisation

This thesis is divided into six chapters. Chapter 2 presents an overview of the main topics and important research works related to this research, such as the different types of document representation, term weighting schemes, state-of-the-art query suggestion methods, and evaluation methods.

Chapter 3 proposes and evaluates a new weighting method based on a new feature called class specific document frequency (CSDF) for document representation. Semantic information has also been explored for document representation and classification. Furthermore, classifier fusion has been investigated in this chapter as well.

Chapter 4 proposes a new ranking method called GCrank for re-ranking web documents returned from Google search engine, and its superior performance has been demonstrated by experimental results.

Chapter 5 proposes and evaluates new query suggestion methods: a hybrid query suggestion method called Tfjac and two query suggestion ranking methods called sGCrank and mGCrank based on the GCrank scores developed in Chapter 4. In addition, this chapter also shows the effect of query suggestion on the user's satisfaction to search returned results.

Finally, the overall contributions and findings, the limitations and future work are detailed in Chapter 6.

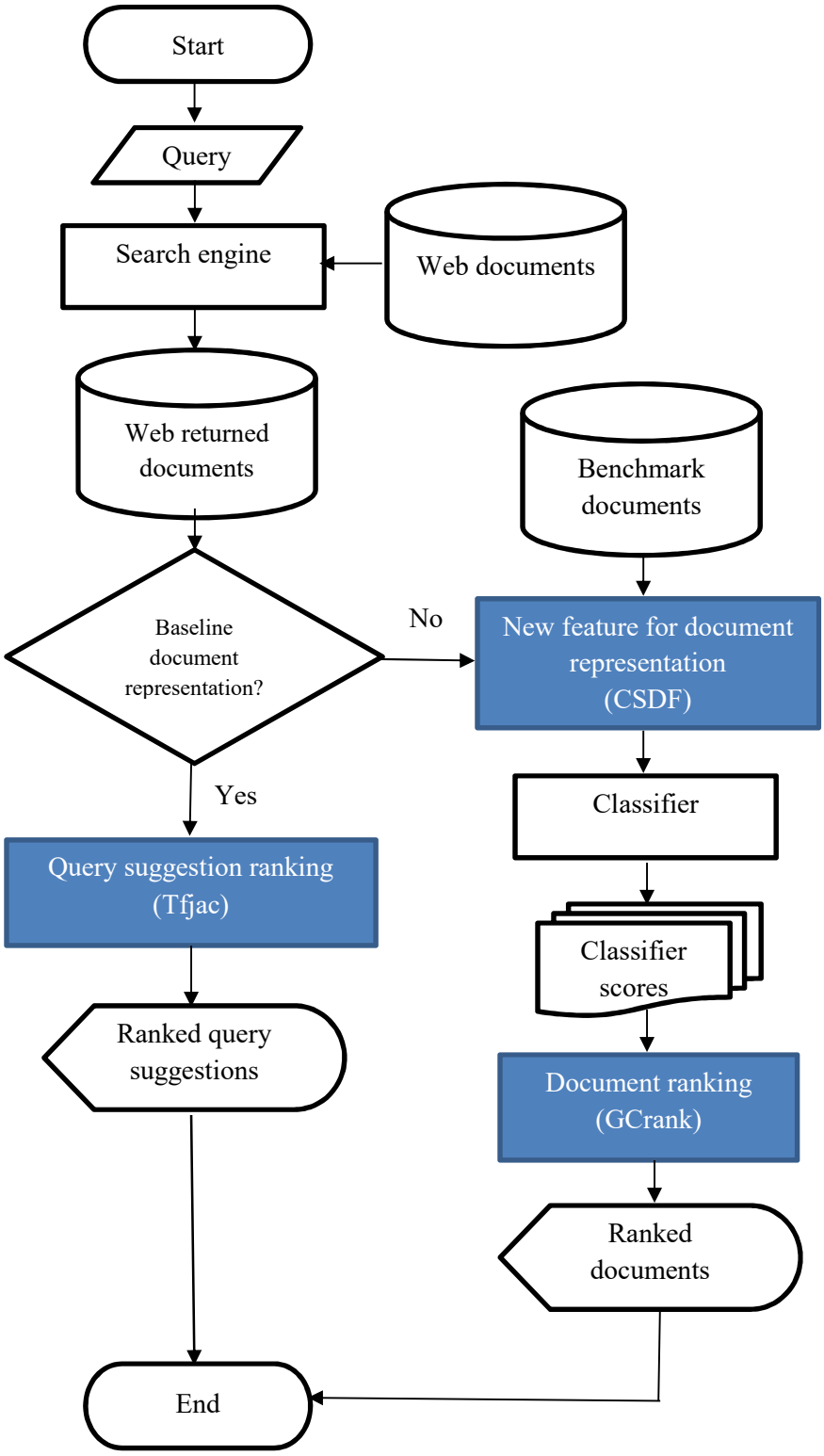


Figure 1.1 The overall thesis contributions and related application domains

Chapter 2 Literature review

Information retrieval (IR) mainly involves similarity between documents or information in documents. It assumes that the data are unstructured and the queries are formed mainly by keywords [13]. Ad hoc retrieval is the standard retrieval task, in which the user specifies his or her information need through a query that initiates a search for documents likely to be relevant to the user [2]. IR deals with the representation, storage, organization, and access to information items such as web pages or documents. It focuses primarily on providing the users with easy access to information in which they are interested, and retrieves all the documents that are relevant to a user query while retrieving as few non-relevant documents as possible. Today's research in IR includes modelling, web search, and text classification [10]. This PhD thesis relates to both web search, especially on query suggestion, and text classification, aiming to propose new weighting techniques for document ranking and ranked query suggestion.

2.1 Document representation

A domain model provides a concise and accessible overview of data and information of interest. In IR, domain modelling is the process of capturing and structuring knowledge or information within a collection of documents, a community, or an area of interest [14] [15] [16]. Domain models have been developed in a lot of research, and transformed into many mediums such as graphs, semantic networks, ontology, concept maps, and term association. A collection of documents is an important source for creating a domain model which can be used to present document relationships and responsibilities of conceptual

classes. Before building a domain model for query suggestion or document representation, the researchers have to extract features from text collections.

In order to reduce the complexity of the documents and make them easier to handle, a document should be transformed from the full text version to standard numeric forms, or a document vector which describes the contents of the document. A document representation may be made of a joint membership of terms which have various patterns of occurrence [17] [18]. A document can be represented in many grammatical units. The smallest unit is a morpheme or character unit, while the biggest unit is a sentence [19], as shown in Figure 2.1.

Using the smallest units N-gram [20] is simply a consecutive sequence of characters with or without regard to word length. It is a very simple and fast method. Trigram is used to detect spelling errors; however, the performance of this approach should be lower than methods that make effective use of the language-specific clues.

A single word is the most popular grammatical unit in document representation techniques. There are two major types of document representation using the single word unit: vector space model and graph-based model [21] [22] [23].

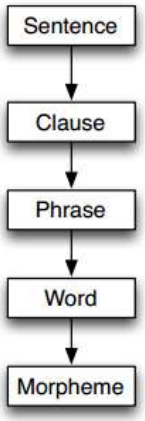


Figure 2.1 Hierarchy of grammatical units [19]

2.1.1 Vector space model (VSM)

Vector space model (VSM) is one of the most popular and widely used models for document representation [24] [25] [26]. Most document representation approaches use a bag-of-words (BOW) as original sources for deriving the representation [24] [27] [28] [29] [30]. The BOW model focuses on the number of occurrences of each term in a document; however, the ordering of the terms is ignored [2]. A vector can represent a document using occurrence counts or other feature values of the terms in the BOW of the document. A weight is assigned to each word using its score. However, spelling errors cause incorrect weights to be assigned to words. Pre-processing and error detection can be helpful. In addition, stemmed single word representation is another solution. Vector space representation is a very quick and simple method, yet it does not consider correlation between and context of keywords, which is very important in understanding the document. Therefore, many researchers have used ontology to solve this problem [24] [25].

There are many different types of VSM. For example, TF-IDF based VSM [10] [20] [24] [31] [32] selects terms that are frequent inside a document but do not appear in many documents. Stemmed single word representation [20] [32] [33] is a method to improve the quality of single word indexing by grouping words that have the same stem. Stemming is the process for reducing inflected or derived words to their stem, base, or root form.

Recent studies have proposed new VSM methods, for example, Tolerance rough set model (TRSM) and Similarity rough set model (SRSM) [34] extended the VSM using Rough Sets Theory and co-occurrence of terms. SRSM is a mathematical model using similarity relation instead of equivalence relation. They used co-occurrence of terms to calculate the semantic relation between terms.

SRSM had better performance than TRSM. There seem to be cases when terms have high co-occurrence but low semantic similarity.

The main problem of a single word representation is the loss of semantic representation. Therefore, a lot of the latest research aims to solve this problem by exploiting semantic features using knowledge-based approaches. For example, WordNet based similarity rough set model (WSSM) [34] is the combination of SRSM and WordNet. The semantic relation between terms is calculated using co-occurrence of terms. WordNet does not include information about pronunciation and the forms of irregular verbs, but contains only limited information about usage. Latent semantic indexing (LSI) [25] [27] [35] [36] is a technique that projects queries and documents into a space with latent semantic dimensions. LSI discovers global structure of the document space, which is based on an algebraic linear transformation of term-document matrix. LSI might not be optimal in discriminating documents with different semantics as it requires additional investment of storage and computation time. Locality preserving indexing (LPI) [25] [36] discovers the local structure and obtains a compact document representation subspace that best detects the essential semantic structure. It has been shown that LPI provides better representation than LSI in the sense of semantic structure. However, the computational complexity of LPI is very expensive and it is unclear how LPI works in real world applications. Wen-tau Yih et al. [37] have proposed a method for measuring word relatedness from various information sources, namely general text corpora (corpus-based), web search results (web-based) and thesaurus-based information. By doing this, they built individual VSMs from each information source separately. Given two words, each VSM measures the semantic relatedness by the cosine similarity of the

corresponding vector in its space. After that, they found the averaged cosine scores derived from these VSMs. It has been shown that the average cosine similarity derived from these models yields a very robust measure. For example, Wikipedia context based VSM provides consistently strong results. This model is close to the average human performance.

2.1.2 Graph-based model

Graph-based representation is another type of single word representation. The strength of the graph approach lies in its ability to capture important structural information hidden in the document and its HTML tags. A graph-based methodology is designed especially for web document representation. The main benefit of graph-based document representation is the retention of the inherent structural information of the original document. A graph can represent any document with minimum loss of information [21] [22]. As shown in Figure 2.2, each unique term (keyword) appearing in a document becomes a node in the graph, and the edges between nodes can represent various relationships between terms. For instance, if word A immediately precedes word B somewhere in section S of a document, then there is a directed edge from the node corresponding to term A to the node corresponding to term B. Although graphs can be directly used for document classification based on graph distance measures, it is common to convert graphs into vectors of various graph measures, especially for machine learning based classifiers [23]. However, the computational complexity of this model is usually very high [21] [22].

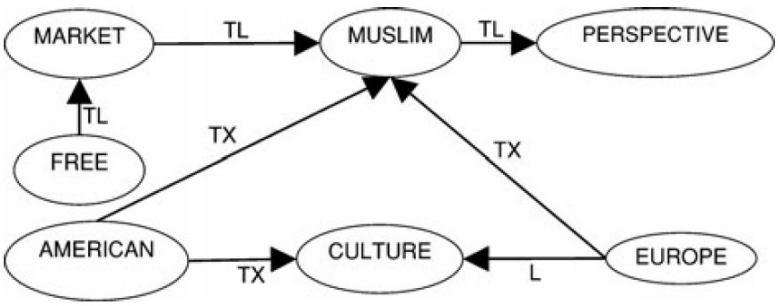


Figure 2.2 An example of standard graph document representation [21]

Graph-theoretic web document representation [21] [22] uses graphs instead of vectors. Each word that appears in a web document, except for stop words, is a vertex in the graph representing that document. Tag sensitive graph model (TSGM) [22] is a directed graph. It can represent the sequence of word occurrence within a document. It can capture some important structural information such as the location of the word within a document. However, this model cannot reflect the proximity of words directly. The context sensitive graph model (CSGM) [22] is a directed distance graph, it can retain information about word pairs which are at a distance of at most *n* in the underlying document, where *n* is the order of the graph. The terms on each web page and their adjacency are examined. Instead of considering only terms immediately within a web document, it looks up to *n* terms ahead and connect the succeeding terms with an edge that is labelled with the distance between them. It can hold almost all of the information that we require to analyse documents. Composite graph [22] uses TSGM model and CSGM model to represent head, link, address, and the text section respectively. CSGM is effective to represent a large text section. Composite graph can hold almost all of the necessary information. Regularized locality preserving indexing (RLPI) [25] decomposes the LPI problem as a graph embedding problem plus a regularised least squares problem. RLPI is significantly faster and obtains similar or better

results when compared to LPI. It remains unclear how to automatically estimate the best parameter to control the amount of shrinkage in the regularisation.

To sum up, a single word representation is the traditional and most popular grammatical unit in document representation techniques. It can reduce the dimensionality of the model space. It is a very simple and fast method.

Other larger grammatical units for document representation, such as phrase representation, clause representation, and sentence representation, have been investigated in recent research. For phrase representation [19] [20], there are two ways to form phrases: statistical and syntactical. Statistical representation uses co-occurrence information in some way to group together words that co-occur more than usual. A syntactical approach uses linguistic information to form the phrases. The performance of phrase representation should be lower than methods that make effective use of the language-specific clues. Example of phrase representation include rich document representation (RDR) [20] [38] and Word N-gram [20] [21]. These methods provide more semantic representation for a document. However, using statistic phrase representation degrades some text processing tasks such as text classification.

Clause representation uses the grammatical unit that can express a complete proposition. A typical clause consists of a subject and a predicate, which is a verb phrase or a verb together with any objects and other modifiers. RDR can represent documents using both phrases and clauses, and bag-of-frame (triplet) [19]. Triplet is a basic unit for document representation (subject-verb-object). RDR and frame-based method perform better than the simpler document representations. These methods provide more semantic representation for a document.

Sentence representation uses the largest grammatical unit. Polarity [39] is a more finely-grained representation of documents, as sequences of emotionally-annotated sentences can increase document classification accuracy. This approach deals with the problem of detecting the overall polarity (positive, negative, or neutral) of a document. However, this method is more suited for datasets with only limited training data. In addition, Hybrid Representation of Documents (HYBRED) [40] is a HYBRED approach which combines different features in a single relevant representation, namely stemming, N-gram, and TF-IDF.

Even though the experimental results from phase, clause, and sentence representations were better than single word representation, these higher level document representations usually result in a higher complexity feature space. Because a single-word representation is very simple and easy to compare with the other methods, in this thesis, only single word VSM models are considered.

2.2 Term weighting

Term weighting is a way to assign numerical values to terms which represent their importance, since not all the terms in a document are the same importance. This numerical statistic is intended to reflect how important a word is to a document in a collection. It helps a word on document stand out from others [41]. Weighting the terms enables IR systems to improve system effectiveness [42]. Term weighting is mainly concerned with the representation of the document space [43]. Therefore, document representation is highly related to term weighting.

For text classification, it is concerned with the automatic classification of documents according to relatively static topic categories. In addition, term weighting is used to reduce the feature space to those terms that are more specific to the topics [44]. Most existing term weighting methods have been proposed and

evaluated in IR and text mining tasks. There are two major types of term weighting: semantic term weighting and statistical term weighting [45]. A semantic term weighting is related to a term's meaning which exploits the semantics of categories and terms using knowledge bases such as WordNet [46]. Statistical term weighting is related to how a term appears in a document or group of documents in a statistical sense.

2.2.1 Statistical term weighting

Statistical term weighting methods assume that a term's statistical behavior within individual documents or sets of documents reflects the term's ability to represent a document's content and distinguish it from other documents. They can be divided into two categories: supervised term weighting methods and unsupervised term weighting methods [47] [48]. Supervised term weighting methods use the class membership information of training documents. More details are presented in Section 2.2.3.

Unsupervised or traditional term weighting methods do not make use of the information on the category membership of training documents. The traditional or baseline term weighting methods which are wildly used in VSM based document representations are term frequency (TF) [10], term presence (TP), and term frequency and inverse document frequency (TF-IDF) [10] [20] [24] [32] [49]. These methods are based on monotonicity assumptions [50]. Firstly, rare terms are no less important than frequent terms. This is reflected by IDF [51] [52]. Secondly, multiple appearances of a term in a document are no less important than single appearances, as reflected by TF. Finally, for the same quantity of term matching, long documents are no more important than short documents, which can be implemented by normalisation.

For term frequency, the simplest choice is to use the raw frequency of term *i* in document *j* [2], i.e., TF_{ji} is the number of times that term *i* occurs in document *j*. Term presence TP_{ji} is the presence of term *i* in document *j*., i.e.,

$$TP_{ji} = \begin{cases} 1, \text{if term i is in document j} \\ 0, \qquad \text{otherwise} \end{cases} \tag{2.1}$$

TF-IDF is the most popular term weighting scheme used in IR. Terms that are more frequent inside a document but less frequent in other documents have higher weights. Some TF-IDF variants have been proposed. Two recommended forms of TF-IDF weights [10] [53] are defined by equations (2.2) and (2.3):

$$TFIDF_{ji} = \left(0.5 + 0.5 \frac{TF_{ji}}{\max TF_j}\right) \times \log_2 \frac{N}{DF_i} \tag{2.2}$$

$$TFIDF_{ji} = \begin{cases} (1 + \log_2 TF_{ji}) \times \log_2 \frac{N}{DF_i}, & \text{if } TF_{ji} > 0 \\ 0, & \text{otherwise} \end{cases} \tag{2.3}$$

where $TFIDF_{ji}$ is the TF-IDF score of term *i* in document *j*, TF_{ji} is the term frequency of term *i* in document *j*, *N* is the number of documents in the training document set, DF_i is the number of documents in which term *i* appears in the training document set.

Statistical term weighting was investigated in many studies. For example, Salton and Buckley [54] have summarised the insights gained in automatic term weighting. The main function of a term-weighting system is the enhancement of retrieval effectiveness. Tsai and Kwee [55] have investigated the impact of term weighting on the evaluation measures. Their research recommends the best term weighting function for both document and sentence-level novelty mining. Novelty mining or novelty detection is a process to filter out repeated or redundant information and to present documents/sentences that have novel information

based on a given threshold. That research compared and evaluated several term weighting functions: TF-IDF, TF, and TP and their performance on document-level and sentence-level novelty mining. Overall, TP was the best term weighting function for document-level novelty mining and TF-IDF was the best term weighting function for sentence-level novelty mining. With a low percentage of novel documents, TF outperformed TP. For a high percentage of novel documents, TF-IDF outperformed TF on the high-precision cases.

2.2.2 Semantic term weighting

Semantic-based text classification was developed after topic models became popular for semantic analysis [56]. Semantic technologies allow the usage of features on a higher semantic level than single words for text classification purposes. The classic document representation enhances the concepts through the extraction of the background knowledge or ontology [57] [58] [59]. Ontology is an explicit knowledge source, such as WordNet, Wikipedia, ODP and YAGOs [60] [61]. These ontologies are used for the extraction of conceptual or semantic features for text documents.

The WordNet database organizes simple words and multi-word expressions of different syntactic categories into the so-called synonym sets (synsets), each of which represents an underlying concept linked through semantic relations [57]. Word structures are provided by WordNet, which not only arranges words into groups of synonyms, but also arranges the synsets into hierarchies representing the relationships between concepts [62]. WordNet also provides different types of word similarity and word relatedness.

In natural language processing (NLP), word similarity is often distinguished from word relatedness. Similar words are near-synonyms, such as car and bicycle,

while related words can be related in any way, such as car and gasoline. They are related but not similar. There are many specific vocabularies in NLP [53]. Homonym refers to two words that share a form, but has unrelated or distinct meanings, such as bank and bat. Polysemy refers to a word used in two different ways. Lemma is the canonical form, dictionary form, or citation form of a set of words, such as banks is equivalent to bank, or sung is equivalent to sing. Synonyms are words that have the same meaning in the same contexts, such as automobile and car, or big and large. Antonyms are words that possess opposite meanings, such as short and long, or up and down. Hyponymy is the class denoted by the super ordinate extensionally, including the class denoted by the hyponym IS-A hierarchy, such as car is vehicle, or mango is fruit.

Many papers about semantic features for text classification have been published. For example, Ferretti et al. [63] have found that the inclusion of semantic information in syntactically and semantically richer corpora could improve the text categorization task, if vocabularies with a sufficient number of features were considered. Document classification based on word semantic hierarchies increases the classification accuracy by 14% in Peng and Choi [62]. Nagaraj et al. [64] have proposed a new approach to represent the semantic level with the use of ontologies. The semantic weight of terms related to the concepts from Wikipedia and WordNet is used to represent semantic information. The semantic vector space model of terms combining the WordNet and Wikipedia can help to further improve the performance of classification. In Yang et al. [56], a novel approach to classifying short texts by combining both lexical and semantic features has been proposed. The combination of lexical and semantic features is achieved by mapping words to topics with different weights. They use Wikipedia

as background knowledge. The results show that their approach has better effectiveness compared with existing methods for classifying short texts.

In particular, Qiming Luo et al. [46] have proposed a novel term weighting scheme by exploiting the semantics of categories and term indexing. TF-IDF exploits only the statistical information of terms in documents. The semantics of categories are represented by sense of terms appearing in the category labels as well as the interpretation of them by WordNet. The process starts from determining the semantics of categories based on terms appearing in category labels, then estimating the semantic similarity of each term with the categories. Finally, they combine the semantic similarity of each term with the category and its term frequency in a document to obtain the feature vector of each document. The results show that the proposed approach outperforms TF-IDF in the cases that the amount of training data is small or the content of documents is focused on well-defined categories.

This thesis considers unsupervised term weighting methods in both statistical and semantic term weighting techniques. A new statistical term weighting method for query suggestion has been proposed. Both statistical and semantic term weighting methods have been investigated for classification tasks.

2.2.3 Term weighting through machine learning

Supervised term weighting methods use the class membership information of training documents. It learns to weight terms using training examples or machine-learned relevance which makes use of prior information on the membership of training documents in predefined categories. Supervised term weighting methods make use of this known information in several ways. One approach is to weight terms by using feature selection methods, such as chi-square, information gain,

and gain ratio. These methods help to assign appropriate weights to terms. Another approach is based on statistical confidence intervals which rely on the prior knowledge of the statistical information in the labelled training data. The third approach combines the term weighting method with a text classifier. The scores used by the text classifier aim to distinguish the positive documents from negative documents are believed to be effective in assigning more appropriate weights to the terms [47]. A lot of research deals with this type of term weighting, such as Lan et al. [47] [65] [66], and Gautam and Kumar [48].

Debole and Sebastiani [50] have pointed out that supervised term weighting is the optimal choice of term weighting function. The best discriminators are the terms which are distributed most differently in the sets of positive and negative training examples. Their proposed method has taken the form of replacing IDF by the category-based term evaluation function that has previously been used in the term selection phase. Their results show that supervised term weighting is efficient and is reused for weighting purposes. In addition, the experimental results of Lan et al. [47] show that the supervised methods outperform unsupervised method in general; however, they are sensitive to noise, and not all supervised term weighting methods are superior to unsupervised methods.

2.3 Document or text classification

A generic problem of finding documents on a specific topic is to group documents by common topics and name each group with one or more meaningful labels. Each labelled group is called a class, that is, a set of documents whose contents can be described by its label [10].

Machine learning learns patterns present in training data, which is used to make predictions relative to unseen and new data. Given a sample of past

experience and correct answer for each example, the objective is to find the correct answers for new examples. There are three types of machine learning: supervised learning, unsupervised learning, and semi-supervised learning. This thesis focuses on supervised learning only. Supervised learning is the machine learning task of finding a model or function from labelled training data that describes and distinguishes data classes or concepts [67]. The performance of the resulting function should be measured on a test set which is separated from the training set.

Document/text classification is an application of machine learning in the form of NLP. It is also called text categorization, topic classification, or topic spotting [2]. By classifying text, it aims to assign one or more classes or categories to a document, and deals with the catergorisation of a new data entry into one or more of the categories based on the values of different attributes [10]. This makes it easier to manage and sort.

One of the main issues in text classification is the transformation of text into numerical data and the selection of important attributes. In machine learning and statistics, feature selection, also known as attribute selection or variable subset selection [13] [68] [69], is an important process of selecting a subset of relevant features for use in model construction. Feature subset selection reduces the data set size by removing irrelevant or redundant attributes (or dimensions) helping to make the patterns easier to understand, and it is an optimisation problem. There are two categories of feature selection: filter approach and wrapper approach. The filter approach evaluates features by their information content. The basic method is the selection of the top-k features from the sorted features in order of their scores; for example, interclass distance, statistical dependence, or information

measure scores. The advantages of the filter method are its speed and generality; however, it tends to select large subsets and is less accurate. On the other hand, the wrapper approach evaluates features by their predictive accuracy using statistical resampling or cross validation. The benefits of the wrapper method are advantageous for giving better performances since they use the target classifier in the feature selection algorithm, but they suffer from being computationally expensive. There are three types of search strategies that are used for feature selection: exponential algorithm, sequential algorithm, and random algorithm [70]. Sequential forward floating search (SFFS) is one of the best feature selection methods [71].

This thesis investigates supervised leaning for text classification. There are three classifiers which are used in this research: kNN [27] [49] [72], LDA [73] [74] [75], SVM [27] [76], Naïve Bayes and logistic regression classifiers.

2.4 Classifier/decision fusion

Fusion technique is to integrate information from multiple sources to produce specific and comprehensive unified units [77]. In Castanedo study [78] with regard to the Joint Directors of Laboratories (JDL) workshop, information or data fusion is a multi-level process dealing with the association, correlation, and combination of information from single and multiple sources to achieve improved accuracy, less expense, and higher quality than could be achieved by the use of a single source alone. Information fusion can be divided according to the relations between the information sources and the input/output data types. There are three types of information for fusion based on the relation between sources, which are proposed by Durrant-Whyte [79]. Complementary is the information provided by the input sources, representing different parts of the scene or composed of non-

redundant pieces to obtain more complete global information. Redundant is two or more independent input sources providing information the same pieces of information or target to increment the associated confidence. Redundant fusion might be used to increase the reliability and accuracy of the information. Finally, cooperative provides information combined into new information from two or more independent sources that is more complex than the original information [80]. In addition, there are five categories of information fusion based on the input/output data type proposed by Dasarathy [81]: data in-data out, data in-feature out, feature in-feature out, feature in-decision out, and decision in-decision out. Feature in – decision out obtains a set of features as input and provides a set of decisions as output. Most of the classification systems perform a decision based on the inputs into the category of classification. Decision in – decision out is also known as classifier fusion or decision fusion. It fuses input decisions to obtain a better or new decision [78].

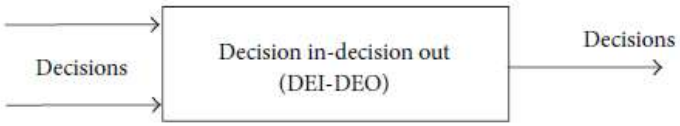


Figure 2.3 Dasarathy's classification [78]

Alternately, Mangai et al. [82] and Ruta et al. [83] have categorised the fusion techniques into three levels of fusion strategies: information or data fusion, feature fusion, and decision fusion, which are in a low-level fusion, an intermediate-level fusion, and high-level fusion, respectively. Firstly, information or data fusion is the process of integration of multiple data and knowledge into a final decision [84]. Data fusion combines several sources of raw data to produce new raw data that is expected to be more informative than the inputs. Secondly, in feature

fusion, multiple feature sets are used to produce new fused feature sets. Feature fusion can derive the most discriminatory information from original multiple feature sets. It is able to select and combine the features, and to eliminate the redundant information and irrelevant features that benefits the final decision. The final set of features is fused together to obtain a better feature set, which is given to a classifier to obtain the final result. Finally, decision fusion or classifier fusion is the combination of classifiers to achieve better classification accuracy [85]. A single classifier is generally unable to handle the wide variability and scalability of the data in any problem domain. The individual decisions are first made based on different feature sets, and then they are combined into a global decision. Most modern techniques of pattern classification use a combination of classifiers, and then fuse the decisions provided by the same selected set of appropriate features for the task. There are several reasons for preferring a multi-classifier system over a single classifier. For example, the dataset is too large to be handled by a single classifier. A single classifier cannot perform well when the nature of features is different, nor improve the generalisation performance.

A multiple classifier system can be achieved in one of the following ways. First of all, a set of classifiers can be created by varying the initial parameters, using the same training data. Secondly, multi-classifier systems can be built by training each classifier with different training datasets. Finally, the variations in the number of individual classifiers such as SVM or kNN are used with the same training dataset. Furthermore, there are two types of classifier combination strategies: classifier fusion and classifier selection. For classifier fusion, every classifier is provided with complete information on the feature space, and the outputs from different classifiers are combined. On the other hand, with classifier

selection, only one classifier's output is chosen in terms of certain criteria. A simple technique used to combine class labels (crisp outputs) from more than one classifiers is the majority voting.

Many papers about information fusion have been published. For example, Dasigi et al. [86] have reported the experimental results on the effectiveness of different feature sets and information fusion from some combinations of them. Information fusion almost always gives better results than the individual feature sets. Danesh et al. [87] have proposed a voting method and decision template method in text classification for combining classifiers. Their results show that these methods decrease the classification error to 15% on 2,000 training data from 20newgroups dataset. Furthermore, Xiao-Dan Zhang [88] has proposed a new decision classification fusion model and algorithm called D-S Theory. The experimental results show that the text classification fusion model can improve the classification precision effectively.

2.5 Document ranking criteria

Since almost 80% of the users who use search engine interest in only top 3 returned results, the ranking at the very top of the results list is exceedingly important [10]. With regard to web documents, the identification of quality context in the web includes domain name, text content, counts, link, web access patterns, click, layout of web page, title, metadata, and font size. There are four types of ranking signals: context signal, structured signal, web usage, and other signal. Firstly, context signals are based on the contents in a page, such as text or word. Secondly, structured signals are the most popular signals, such as the linked structure and anchor text. Thirdly, the web usage or implicit feedback is inferred from user behavior, such as click data. Finally, the other signals include IP

address, language, query history, and cookies or personalisation. Recently, machine learning has found applications in IR, especially in the ranking process. There are some reasons to expect the use and importance of machine learned ranking approaches to increase over time [2]. Web search ranking often serves as a supervised machine learning problem. The success of Google, Yahoo! and Bing search engines led to an increased challenge in algorithms for automated web search ranking [89].

Document ranking are separated into three major categories: content-based ranking, hyperlink-based ranking or connectivity-based ranking, and hyperlink-content-based ranking.

2.5.1 Content-based ranking

Content-based ranking technologies were developed for retrieving web pages for specific queries and similarity page queries. It uses context signal as features for ranking. Their algorithms usually implement by matching queries with keywords or features in web documents and users' web logs. VSM is a traditional document content representation method. It is based on term presence or term frequency and inverse document frequency (TF-IDF), or ranked retrieval model. Ranked retrieval model is the traditional ranking model based on VSM framework. The system returns an ordering of the top documents in the collection with respect to a query. When a system produces a ranked result set, the size of the result sets is not an issue. Only the top-k results are concerned. The documents are ranked in order of the query and document matching scores. These scores measure how well the document and query match. Intuitively, the more frequent the query term is in the document, the higher the score should be. A way of assigning a score to a query

and document pair is needed. Cosine similarity and Jaccard coefficient are usually used to give the matching scores.

The VSM using cosine similarity [53] is one of the most commonly used methods to rank returned documents according to the proximity or similarity between two vectors representing the query and the document. For length-normalised vectors, cosine similarity is simply a dot product as shown in equation (2.4):

$$\cos(\vec{q}, \vec{d}) = \vec{q}.\vec{d} = \sum_{i=1}^{|v|} q_i d_i \quad (2.4)$$

where q_i is the TF-IDF weight of term i in the query, d_i the TF-IDF weight of term i in the document, $\cos(\vec{q}, \vec{d})$ is the cosine of the angle between $\vec{q}$ and $\vec{d}$, and $|v|$ the number of terms in the query.

On the other hand, Jaccard coefficient [2] [53] does not consider how many times a term occurs in the document (TF). Sometimes, rare terms in a collection are more informative than frequent terms. Jaccard coefficient is a measure of overlap of two sets: query A and document B, which may not have the same size. The Jaccard coefficient is calculated as follows:

$$Jaccard(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (2.5)$$

where $\cap$ and $\cup$ represent intersection and union, respectively. After computing these scores, the documents are ranked with respect to the query by these scores, and then the top-k documents are returned to the user.

In recent years, there have been new document representation methods and similarity measures proposed such as learning to rank and personalisation-based ranking. For example, Du and Hai [90] have proposed a method based on formal concept analysis (FCA) to measure webpage similarity.

In the past decade, learning to rank has emerged. It is the application of machine learning in the construction of ranking models for IR systems. This method aims to minimise the number of mistakes using supervised learning classifiers such as SVM and training data from users, such as click-through data. Xiang et al. [91] have developed different ranking criteria for different types of contexts. They were integrated into a state-of-the-art ranking model by encoding features of the model using a learning to rank approach, which are the context information including previous queries and the search results that users click on or skip. Derhami et al. [92] have represented two novel ranking methods using reinforcement learning concepts and a new hybrid approach which combines BM25 (best matching 25) [74] and their machine learning methods.

Personalisation based ranking has been recently investigated. Lu et al. [93] have proposed a user model based ranking method. This model is mainly used to capture and record the user's interests. Wang et al. [94] have proposed a general ranking model adaptation framework for personalised search using a user-independent ranking model and the number of adaptation queries from individual users.

Semantic web search is also based on content-based ranking. The objective of research on semantic search ranking [95] [96] [97] is to improve traditional information search and retrieval methods by using ontologies. However, the heterogeneity and overlapping domains are problems.

Zhuang and Cucerzan [98] have proposed a novel Q-rank method using two features from log files: adjacent queries which are the previous and next queries and the most frequently seen query suggestion. A re-ranking score for each document based on its lexical overlap with a set of most popular query

suggestions and adjacent queries to the original query. Their results show that the largest improvements were measured for the top 10 ranked web documents. The proposed method achieved the best ranking performance for the number of re-ranking candidates equal to 30. Using the adjacent query features alone produced the best ranking results. In addition, Xiang et al. [91] have developed different ranking principles for different types of contexts and adopted a learning to rank approach (RankSVM) which integrated the ranking principles into a start-of-the-art ranking model by encoding the context information as features of the model. Context information is the previous queries and the answers clicked on or skipped by users to the previous queries. Their results show that their approaches improved the ranking of a commercial search engine which ignores context information. Their method outperforms a baseline method which considers context information.

2.5.2 Hyperlink-based ranking or connectivity-based ranking

Structured signal is the oldest and most popular signal for web document ranking. Hyperlink-based ranking uses this signal. The early ranking methods focus on the number of hyperlinks that point to a webpage or the incoming links [10]. Links save information that can be used to evaluate the importance and relevance of webpages to the user's query to some extent. HITS and PageRank are the examples of the well-known hyperlink-based ranking methods.

HITS [2] [10] [90] stands for hypertext induced topic search. Hyperlink structures of webpages in the web graph induced are represented by authorities and hubs. A webpage that points to many other webpages is a good hub. A webpage that is linked by many different hubs represents a good authority. It is a query-dependent method; however, the drawbacks of this method are the repeated

web results and topic diffusion. In addition, the HITS algorithm also produces some problems in real applications such as the time and space costs of constructing the subgraph of the search topic being high. It is also unsuitable for specific queries.

PageRank [10] is the best-known method because it is an algorithm used by Google to rank websites in their search engine results. Suppose that a webpage *a* is pointed by webpages p_1 to p_n on the web graph, and a user jumps to webpage *a* with probability *q* or follows one of the hyperlinks of webpage *a* with probability *1-q*. The PageRank of webpage *a* is given by the probability *PR(a)* of finding the user in webpage *a*, which is defined by equation (2.6):

$$PR(a) = \frac{q}{T} + (1-q)\sum_{i=1}^{n}\frac{PR(p_i)}{L(p_i)} \tag{2.6}$$

where *T* is the total number of webpages on the web graph, $PR(p_i)$ is the PageRank of webpage p_i, $L(p_i)$ is the number of outgoing links of webpage p_i, and $n=L(a)$. This method may have a problem, if the real web graph contains dead ends where webpages have self-link or no link. The solution to this problem is the jumping criteria of the Markov chain. Furthermore, Alkhalifa [99] has found that the adjacency matrix used as a basis for PageRank may have biased spaces. This problem needs to be taken into consideration.

Some researchers have proposed novel ranking methods. Baezy-Yates and Davis [100] have presented a variant of PageRank called WLRank (Weighted Links Rank). WLRank gives weights to links based on three attributes: relative position, tag, and length of the anchor text. Their results show that the most effective attribute was anchor text length. WLRank improves PageRank precision for the top 10 results and the relative position was not so effective.

2.5.3 Hyperlink-content-based ranking

Hyperlink-content-based ranking [90] aims to find an appropriate balance between the relevance and popularity of webpages. Search engines use a combination of hyperlink-based and content-based algorithms in general. The ranking score or priority value of a webpage is computed by a combination of a score related to its hyperlinks and another score related to its content. For example, the combination of BM25 and PageRank can be the baseline to evaluate new ranking methods. A simple ranking function [10] is to combine text-based (Bayesian network) and link-based ranking; for example, the combination of BM25 for selection and PageRank for ranking, as in equation (2.7):

$$R(P,Q) = \alpha BM25(P,Q) + (1-\alpha)PR(P) \tag{2.7}$$

where α is between 0 and 1. If α = 1, BM25 method alone is used to rank, while α = 0, PageRank alone is used to rank.

Although Google ranking is well recognised as the best webpage ranking method, there is still room for improvement. This thesis investigates whether re-ranking Google search returned web documents by using document classification scores is able to improve ranking performance in terms of generally used performance evaluation criteria.

2.6 Query suggestion

In general, searching or retrieval of relevant information from the web is a very difficult task because of two main problems. Firstly, there are many problems with the available data, such as very large volume of data available, unstructured and redundant, ubiquitous databases, quality of data and data spam, the fast pace of change and heterogeneous data. Secondly, there are problems from the users.

The system should interpret the queries to find the answers and specify the queries [101]. With these problems, a search engine is an important tool which can help the users to specify their information needs.

A main feature of the search engine is query suggestion, or query expansion. It is a methodology studied in the field of computer science, particularly within NLP and IR where the system gives additional input on query with related words or phrases, possibly suggesting additional query terms. It aims to improve the overall recall of the relevant documents [10] [102]. The ways to do this type of research still present a grand challenge. To develop automatic query suggestions, researchers have to review and learn how to extract some features from data repositories, such as log files or documents or both, how to create models, how to generate query suggestions, and finally, how to adapt this model to be more related to the user's intention. This section will review the current methods for development of query suggestion.

Query suggestion may be automatic or semi-automatic. For automatic query suggestion, the system finds and includes new terms without reference to the user. For semi-automatic query suggestion, the system finds new terms and offers them to the user for possible inclusion. It is necessary to present the terms to the user in some reasonable order, preferably one in which the terms most likely to be useful are near the top [43]. Dynamic query suggestion or query reformulation is more complex than query expansion, which forms new queries using certain models [102] [103] [104]. In modern search engines, query suggestions are triggered automatically as the user types, rather than upon request, and are called "auto-complete or auto-suggest" in the dropdown lists. These are found with various morphological forms of words by stemming each word in the search query and

also fixing spelling errors and automatically searching for the corrected form [10]. An example is given in Figure 2.4.

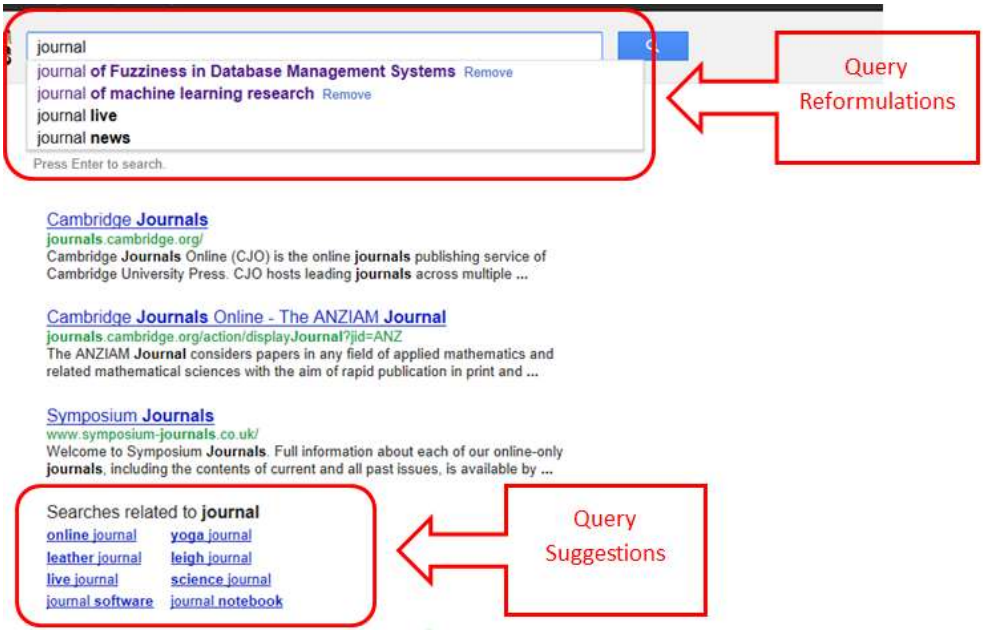


Figure 2.4 An example of query suggestion and reformulation on Google

Figure 2.5 illustrates the overall process of generating query suggestion in search engines. The process starts from a user submitting his/her query, and then a search engine returns relevant documents from a large amount of online databases. At this stage, all actions will be saved in history files or log files which can be a relevance feedback source. After that, the system extracts features from these returned documents (another relevance feedback source) using a query suggestion model to generate and rank query suggestions. Finally, the documents returned from the search engine and the generated query suggestions are given to the user. If the user chooses any suggestion, it will be a new query and this process will be repeated.

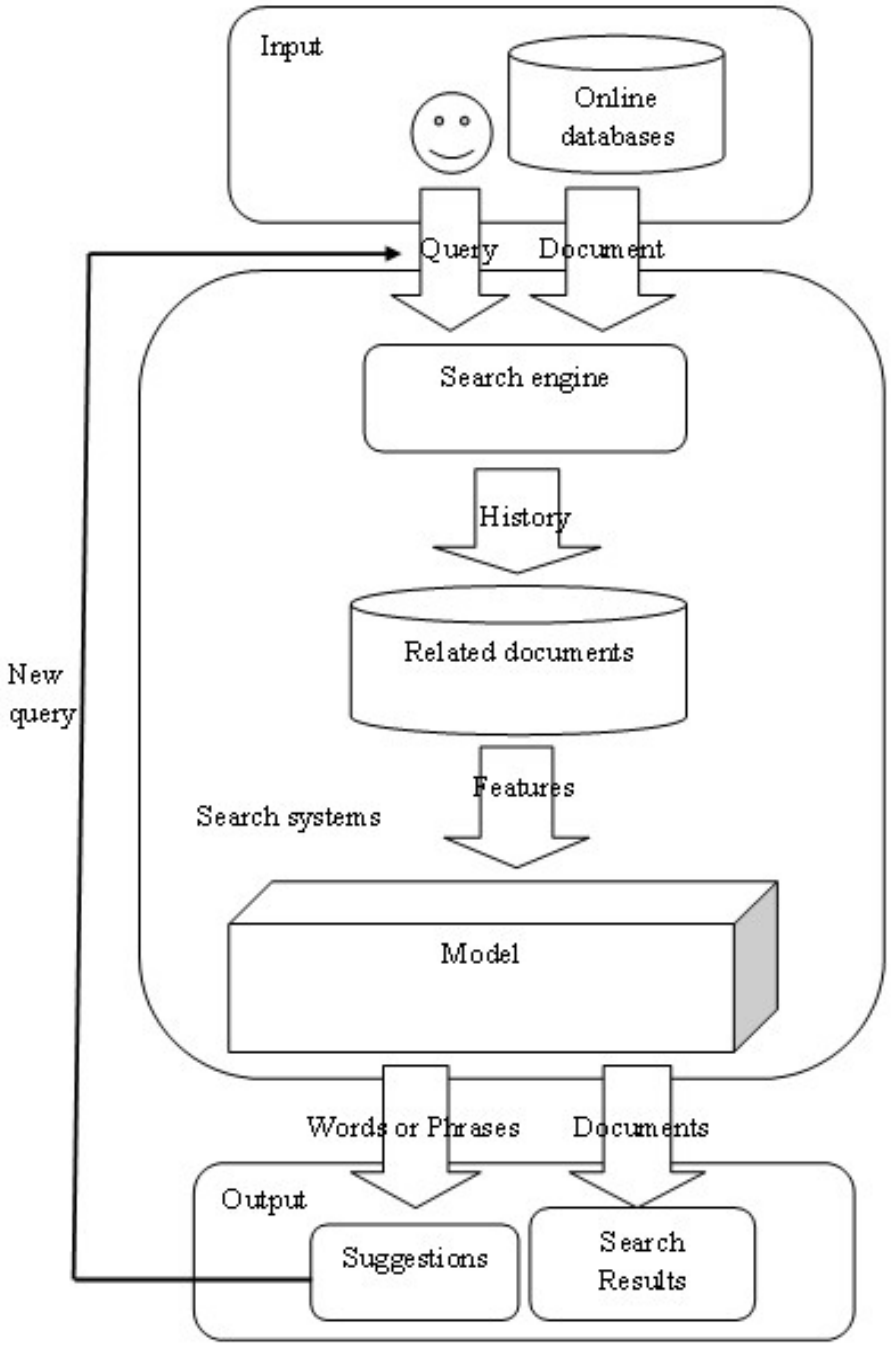


Figure 2.5 The overall process of generating query suggestions [10]

2.6.1 Features for query suggestion

Relevance feedback plays an important role in query suggestion. The system derives the feedback information from various sources of features, such as log files, web documents, and ontologies. There are two main categories of relevance feedback: explicit feedback and implicit feedback.

2.6.1.1 Explicit feedback

Explicit feedback is provided directly by users and is called original formulation. This feedback is used for query suggestion by adding some new terms to the original query. In addition, user click results are also the source of feedback information. However, collecting feedback information is expensive and time consuming [10].

2.6.1.2 Implicit feedback

Implicit feedback is derived by the system and has not been participated in by the user. There are two categories of implicit feedback: global analysis and local analysis [105]. Global analysis uses information from the whole set of documents in the collection. It examines word occurrences and relationships in the corpus and uses this information to expand any particular query. The global thesaurus is composed of classes that group correlated terms in the context of the whole collection. These correlated terms, which have high term discrimination values, can be used to expand the original user query.

Local analysis involves only the top ranked documents retrieved by the original query. These techniques work on local feedback. The suggestions are generated from the correlated terms or similar neighborhoods with the same synonymy relationship. The system derives from the feedback information of several sources of features.

Document-based features are extracted from text transaction from the documents, web documents, XHTML tags, or the URLs the user clicks after having submitted a query to finding relevant terms. In addition, pseudo relevance feedback or blind relevance feedback is a feature which assumes that the top-k ranked documents are relevant and can generate query suggestions [2].

The log-based features are very valuable resources to generate query suggestions and to automatically acquire feedback terms for guided search. They record a query identifier, a session identifier, the submission time, various forms of submitted query, and additional information such as IP address [106]. There are three sub-features from the log files. Click-through data, the clusters of similar queries and similar URLs in the log files that the users have clicked create a graph. A session can easily consist of a number of search goals and search missions. Kruschwitz et al. [106] have decomposed individual sessions in a log file into more fine-grained interactions called dialogues. If a user selects a suggestive term for refinement or replacement, then it is part of a dialogue. The dialogue continues for as long as the user selects terms or until they start a new search or the session expires. The terms from the log file will be ranked higher during refinement recommendation since they come from real users' experiences [107].

Both global analysis and local analysis are capable of expanding the query; however, global analysis is more expensive than local analysis. There is much research on query suggestion using log files, from which users' search behaviors and information needs can be derived, such as Nallapati and Shah [102], Fonseca et al. [3], Kato et al. [108], Beaza-Yates et al. [109], Boldi et al. [110] and [111], Cao et al. [112], Huang et al. [113], Kruschwitz et al. [106], Liao et al. [114], and Mei et al. [115]. Knowledge-driven models for generating query suggestions are created by applying various ontologies, such as WordNet [116] [117], Wikipedia [118], ODP and YAGO [60] [61] [119] [120]. Query suggestions can be developed from query related features extracted from web returned documents by

search engines as well [4]. There are some studies on query suggestion that combined query log and web search results [121] or query log and ontology [122].

2.6.1.3 A comparison of the commonly used features

Global analysis is inherently more expensive than local analysis. On the other hand, global analysis provides a thesaurus-like resource that can be used for browsing without searching. According to Beaza-Yates et al. [10], local analysis techniques are interesting because they take advantage of the local context provided by the query. However, the combination of local analysis, global analysis, and user click is a current important research problem.

Makoto et al. [108] have examined that query session data and click-through data can provide more effective query suggestions. Query session data is the users' query sequence and query sequence history, and click-through data is the user clicked URLs and selected query suggestions. These were evaluated by two types of query ranking methods. The first major method is session-based ranking methods which aim to find queries that often follow, or are followed by, a given query within the same session. The second major method is click-based ranking methods based on the similarity of URLs clicked in response to a query or the clicked-URL similarity. It has been shown that query session data outperforms click-through data in terms of click-through rate. Furthermore, the experimental results from [106] illustrate that dialogues based methods tend to perform better than sessions based methods when assessing the actually extracted suggestions.

There are three important sources for generating query suggestions: search result documents, log files, and ontologies. They have their own advantages and disadvantages which are compared in Table 2.1.

Table 2.1 Comparison of the sources of features

Sources	Benefit	Drawback
Log files	• Low cost • Good result • From real users' needs	• No new word from past • Require a large log • Low frequency query term • From only one search engine
Search returned documents	• Can find more relevant terms • Already relevant to query term	• High cost • Too many non-relevant documents and misleading expansion terms • Documents in the feedback set are only partially related to the topic (topic drift)
Ontologies	• Already relevant to query term • IS-A relationship • Semantic relatedness • Bridge the gap between query and documents • Alternative labels of concept (synonyms) may improve the search	• Corpus bias • Relying on such curated lexical resources • Requires significant expertise and effort • Language specific
Hybrid feature I (From log and document)	• Good result • More relevance terms • Useful for new search engine with little or no query log	• High cost
Hybrid feature II (From document and ontology)	• Consistently strong results • Close to the averaged human performance	• High cost

Log files are very valuable resources from real users' information to generate query suggestions. However, log files are privacy protected and very hard to use for experiments. Therefore, this thesis investigates query suggestion methods based on pseudo relevance feedback which is the top-k document result returned from search engines only.

2.6.2 Methods for query suggestion

There is a lot of research on query suggestion with search engines [123] [124] [125]. The first type of query suggestion methods are graph-based methods based on log files, such as query flow graphs, bipartite graphs [115] [126], or query document graphs [110] [127]. A query flow graph (QFG) is a graph representation of the interesting knowledge about querying behavior or an outcome of query-log mining which is proposed by Paolo Boldi et al. [111]. The nodes of this graph are all the queries contained in the log. Cao et al. [112] built a graph on the same set of nodes of the query graph defining new non-oriented edges which represent the similarity relations among queries. Beeferman and Berger [128] have applied a hierarchical agglomerative clustering technique to click-through data to find clusters of similar queries and similar URLs. A bipartite graph is created from those which are iteratively clustered by choosing the two pairs of most similar queries and URLs. Makoto P. Kato et al. [104] have proposed a new method to present query suggestion which is designed to help two popular query reformulation actions; specialization and parallel movement. Ibrahim et al. [107] [129] proposed a novel method to adapt the concept hierarchy model [130] [131]. The general idea of building this model is to use term co-occurrence to create a subsumption hierarchical tree. This model is not extended from an entire intranet collection but by using terms from search logs. The result of their experiment illustrates that the adaptive model improves its query recommendation performance over a period of time. Yang Song et al. [122] have proposed a novel query suggestion framework which combines the strength of graph-based models capable of addressing topic-level suggestions from log files and the probabilistic models which can generate term-level suggestions.

Query suggestions are also generated from mathematic and statistical method based on web returned documents. The Maximum Likelihood Estimation (MLE) is a basic approach in the statistical natural language processing. For each query, the researchers [106] applied MLE to the pairs of queries which were extracted from the query modification sequences in the log file. Formal concept analysis (FCA) [132] is a branch of mathematical lattice theory that provides the means to identify meaningful groupings of objects that share common attributes as well as providing a theoretical model to analyse hierarchies of these groupings.

A hybrid model for query suggestion combines features from both documents and log files. Jiang-Ming Yang et al. [121] have proposed a unified strategy to combine query logs and search results as the context information for query suggestion. They leveraged both the users' search intentions for popular queries and the power of search engines for unpopular queries. Ibrahim et al. [107] have presented a hybrid model which integrates from two models: concept hierarchy model (SHReC) and QFG. The first model is built from an Intranet's document, and the second model is built from search logs. This is able to mine suggestions from both the document collection as well as the search logs.

In recent years, some researchers proposed machine learning methods for query suggestion. For example, to generate query reformulation that modifies queries from the previous query words [133], Huang et al. [134] have analysed and evaluated various types of query reformulation such as removing words, adding words, spelling correction, or stemming from query logs. By doing this, they constructed their own taxonomy by combining the types of query reformulation and then developed a rule-based classifier. According to Tuan and Kim [135], they developed automatic suggestions for PubMed by query

reformulation from query logs. They used three machine learning methods: Naïve Bays [136], maximum entropy classifier [137], and support vector machine [138] to reformulate classification. Youngho Kim et al. [139] have proposed a novel boolean query suggestion technique for professional searches. Decision tree learning of pseudo-labelled documents was exploited by boolean queries, which then ranked query suggestions using query quality predictors. Umut Ozertem et al. [140] have proposed a machine learning model which learns the probability that a follow-up query relevant to the initial query. It generates query suggestions that are beyond the past related queries. These can improve the suggestion relevance and add more sources of suggestions. Furthermore, association rules [3] [4] [109] [141] [142] [143] [144] can be altered periodically to generate new related query groups. This is a significant feature for searching on dynamic web. The good points of this method are simple: a low computational cost and good results. However, log files are only one component which they found interesting in this research. They do not read any detail of document content or information from the search engine. Baeza-yates et al. [109] have proposed a method to suggest a list of related queries. These related queries are based on previously issued queries. The method proposed is based on a query clustering process in which groups of semantically similar queries are identified. Ant algorithm is proposed by Dorigo et al. [145] as a multi-agent approach to optimisation problems like the travelling salesman problem. Ant colony optimisation (ACO) [146] is based on a colony of artificial ants. The first objective of ACO is to locate the shortest path, and it is then applied as an engineering approach to the design and implementation of software systems for the solution of difficult optimisation problems. In Kruschwitz et al. [106], they used the ACO analogy to first populate and then

adapt a directional graph similar to QFG. There is a lot of research which uses ACO for query suggestion, such as [147] [148] [149].

Table 2.2 shows a summary of the various types of models to generate query suggestions. Each model has different features and different categories. They are different in terms of the types of implicit feedback, such as log files, documents, and the methods for query suggestion. However, they share a similar problem which is high computational complexity.

Table 2.2 Comparisons of the reviewed models

Query suggestion Models	Sources		Models				
	Implicit feedback		Data-driven		Knowledge-driven		
	Local analysis		Unsup*	Sup*	WN*	Wiki*	ODP*
	Doc*	Log					
Association Rule [3]		✓	✓				
Association Rule and Fuzzy [4]	✓		✓				
Query Flow Graph [110] [111] [112] [126] [127]		✓		✓			
Bipartite Graph [115]	✓		✓				
Bipartite Graph [104]	✓			✓			
SHReC [107]	✓		✓				
Mathematic [106] [109] [133] [134] [140]	✓			✓			
Concept Sequence Suffix Tree [112]		✓		✓			
Hybrid [107] [121]	✓	✓	✓				
Term-transition Graph [122]		✓	✓	✓		✓	✓

Doc* = Document or the URLs corresponding to documents from logs
Unsup* = Unsupervised
Sup* = Supervised
WN* = WordNet
Wiki* = Wikipedia
ODP* = Open Directory Project

2.7 Evaluation methods

The standard approach to IR system evaluation relates to the notion of relevant and irrelevant documents. Relevance is evaluated relative to an information need, not a query [2]. Therefore, the evaluation in IR system is to measure how well the system meets the information needs to the users. It is possible to define approximate methods which have a correlation with the preferences of a population of users [10]. There are various methods for evaluating the retrieval quality of the IR system.

Precision and recall are the basic measures used in evaluating search strategies. Precision is the ratio of the number of relevant records retrieved to the total number of irrelevant and relevant records retrieved or the number of true positives divided by the sum of true positives and false positives. Recall is the ratio of the number of relevant records retrieved to the total number of relevant records in the database or the number of true positives divided by the sum of true positives and false negatives.

Table 2.3 The contingency table

	Relevant	Irrelevant
Retrieved	True Positive (*tp*)	False Positive (*fp*)
Not retrieved	False Negative (*fn*)	True Negative (*tn*)

Precision and recall are calculated by equations (2.8) and (2.9), respectively:

$$Precision = \frac{tp}{tp+fp} \tag{2.8}$$

$$Recall = \frac{tp}{tp+fn} \tag{2.9}$$

Recall is difficult to calculate in a large collection. Precision and recall are not always useful. They assume that all the documents in the search results have been seen. However, the user is not usually presented with all the documents in the

search results. Only top ranked documents are concerned. Precision@k (P@k) [150] [151] is the precision for the top-k ranked results. For example, precision@10 (P@10) and precision@20 (P@20) are the precision for the top 10 and the top 20 query suggestions or documents, respectively. They are calculated by equations (2.10) and (2.11):

$$P@10 = \frac{number\ of\ relevant\ suggestions\ among\ top\ 10}{10} \tag{2.10}$$

$$P@20 = \frac{number\ of\ relevant\ suggestions\ among\ top\ 20}{20} \tag{2.11}$$

It has the advantage of not requiring any estimate of the size of the set of relevant documents. However, the disadvantage is that it is the least stable of the commonly used evaluation measures and does not average well, since the total number of relevant documents has a strong influence on precision at k.

F-measure or F_1 score is the harmonic mean which combines precision and recall into a single number. It can be interpreted as a weighted average of the precision and recall. The F_1 score reaches its best score at 1 and worst at 0. It is a popular metric for evaluating text classification algorithm. F-measure is defined by equation (2.12):

$$F(j) = \frac{2}{\frac{1}{r(j)}+\frac{1}{P(j)}} \tag{2.12}$$

where *r(j)* is the recall at the *j*-th position in the ranking. *P(j)* is the precision at the *j*-th position in the ranking [10].

In addition, accuracy is often used for evaluating classification problems. It is the fraction of its classifications that are correct. The accuracy is defined by equation (2.13):

$$Accuracy = \frac{tp+tn}{tp+fp+fn+t} \tag{2.13}$$

Precision, recall, F-measure, and accuracy are set-based measures. They are the evaluations for unranked documents. However, ranked retrieval results are very important in IR applications. Mean reciprocal rank, mean average precision, and discounted cumulated gain are used for evaluating ranked documents [2].

Mean reciprocal rank (MRR) [151] [152] is a statistic measure for evaluating any process that produces a list of possible responses to a sample of queries, ordered by the probability of correctness. For a sample of queries, the reciprocal rank of a query response is the multiplicative inverse of the rank of the first correct answer. MRR is suitable for web document/query suggestion's ranking evaluation. For query j, the reciprocal rank of a relevant document or good query suggestion i, RR_{ji}, is the multiplicative inverse of the rank of this document/query suggestion in the list of potential documents/query suggestions made by a document/query suggestion method, r_{ji}. It equals 0 if no such document/query suggestion is in the list. RR_{ji} is defined by equation (2.14):

$$RR_{ji} = \frac{1}{r_{ji}} \tag{2.14}$$

MRR is the average of the reciprocal ranks of all the relevant documents or good suggestions for all queries which is defined by equation (2.15):

$$MRR = \frac{1}{q}\sum_{j=1}^{q}\frac{1}{Q_j}\sum_{i=1}^{Q_j} RR_{ji} \tag{2.15}$$

where Q_j is the number of the relevant documents or good suggestions for query j, q is the number of queries. In this thesis, its relevant document or good query suggestions for a query are determined partly by users' decisions and partly by the Google query suggestions.

Mean average precision (MAP) [2] [151] supposes that users are concerned about finding many relevant documents/suggestions, and highly relevant documents/suggestions should appear first in a suggested list.

Let the rank of the *i*th relevant document/suggestion in the potential documents/suggestions made by a document/suggestion ranking method for query *j* be r_{ji}. The precision of the *i*th suggestion is defined by equation (2.16):

$$P_{ji} = \frac{number\ of\ relevant\ suggestions}{number\ of\ suggestions\ examined} = \frac{i}{r_{ji}} \tag{2.16}$$

For an irrelevant suggestion, the precision is set to 0. MAP is defined as the average precision of all the documents/query suggestions for the queries, as shown in equation (2.17):

$$MAP = \frac{1}{q}\sum_{j=1}^{q}\frac{1}{Q_j}\sum_{i=1}^{Q_j}P_{ji} \tag{2.17}$$

where Q_j is the number of relevant documents/suggestions for query *j* and q is the number of queries.

MAP allows only binary relevance assessment: relevant or irrelevant. It does not distinguish highly relevant documents/suggestions from mildly relevant documents/suggestions. On the other hand, discounted cumulated gain (DCG) [2] [53] is a metric that combines graded relevance assessments effectively. This grade is the rating or weighting factor of the rank of the *i*th document/suggestion.

Cumulative gain (CG) is designed for situations of non-binary notions of relevance. Cumulative gain of the Q_j documents/suggestions for query *j* is defined by equation (2.18):

$$CG_j = w_1 + w_2 + \cdots w_{Q_j} \tag{2.18}$$

where w_i is the rating or weighting factor of the ith document/suggestion. Discounted Cumulative Gain (DCG) is defined by using a discount factor $1/(log_2 i)$, which is shown in equation (2.19):

$$DCG_j = w_1 + \frac{w_2}{\log_2 2} + \frac{w_3}{\log_2 3} + \ldots \frac{w_{Q_j}}{\log_2 Q_j} \tag{2.19}$$

Normalised discounted cumulative gain (nDCG) of query j is defined by equation (2.20):

$$nDCG_j = \frac{DCG_j}{IDCG} \tag{2.20}$$

where IDCG is the maximum possible DCG. Average DCG (DCG) and $nDCG$ over q queries are defined by equations (2.21) and (2.22), respectively:

$$DCG = \frac{1}{q}\sum_{j=1}^{q} DCG_j \tag{2.21}$$

$$nDCG = \frac{1}{q}\sum_{j=1}^{q} nDCG_j \tag{2.22}$$

Regarding the most standard IR task, the system aims to provide information or documents which the user desires to know more correctly and quickly. Therefore, a user's information needs are the most important issue. To decide whether a document is relevant or not relevant, users play the most important role in this evaluation task. The system and user utility are comprised of how satisfied each user is with the results the system gives for each information need. These might include quantitative measures in both objectives, such as time to complete a task, and subjective, such as a score for satisfaction. The system utility is a satisfaction score of the system which users are given. The user utility is a way of quantifying aggregate user happiness, based on the relevance, speed, and user interface of a system. For example, they are happy if customers click through to

their site. User happiness is an elusive measure, and this is partly why the standard methodology uses the representative of relevance for search results. The participants are observed, and ethnographic interview techniques are used to get qualitative information on satisfaction. Questionnaires provide data about users' opinions and the results are reported to researchers. For the evaluation methods of ranked retrieval results, the users or participants are involved to choose the relevant results and to rank them in order with respect to the query. User studies are very useful, but they are time consuming and expensive to do [2].

2.8 Summary

A single word is the most popular grammatical unit in document representation techniques. There are two major types of document representation using single word unit: vector space model (VSM) and graph-based model. VSM is one of the most popular and widely used models for document representation. However, spelling error and loss of correlation are the major problems. Spelling errors cause incorrect weights to be assigned to words. Ontology or knowledge base can solve this problem. Graph-based model has higher computational complexity than VSM. Furthermore, there are high grammatical levels of document representation, such as phase, clause, and sentence representation. Even though same experimental results from phase, clause, and sentence representation were better than single word representation, these higher level document representations usually result in a higher complexity feature space. Because single-word representation is simple and easy to compare with other methods, in this thesis, only "bag-of-words" VSM model is considered.

Document representation is related to weighting terms in a document. Lan et al. [47] have found that supervised methods outperform unsupervised methods in

general; however, not all supervised term weighting methods are superior to unsupervised methods. In addition, semantic term weighting allows the usage of features on a high semantic level for text classification purposes. However, the performance of classification depends on how good ontology or knowledge bases are.

Term weighting is critical in single word based document representation and classification as well. One of the research focuses of this thesis is to develop effective term weighting methods

Document ranking can be divided into three major categories: content-based ranking, hyperlink-based ranking, and hyperlink-content-based ranking. Although Google ranking is well recognised as the best webpage ranking method, there is still room for improvement. This thesis will focus on content-based ranking and exploit the use of document classification scores in document ranking.

Query suggestion is a main feature of the modern search engines. It can generate from explicit and implicit feedback. Explicit feedback is provided directly by users whilst implicit feedback is derived by the system and has not been participated in by the user. For explicit feedback, collecting feedback information is expensive and time consuming. There are two categories of implicit feedback: global analysis and local analysis. Global analysis uses information from the whole set of documents in the collection whilst local analysis involves only the top ranked documents retrieved by the original query. Global analysis is more expensive than local analysis. The system derives from the implicit feedback information from several sources of features. Log files are very valuable resources from real users' information to generate query suggestions. However, log files are privacy protected and very hard to use. Pseudo relevance feedback is document-

based features which assume that the top-k ranked documents are relevant and useful to generate query suggestions. This thesis develops a query suggestion method based on pseudo-relevance feedback using existing and the proposed term weighting methods.

Chapter 3 CSDF and semantic information for VSM-based document representation and classification

3.1 Introduction

Finding relevant information from the enormous web, which contains millions of web documents on the internet, or from the huge amount of document databases is a grand challenge. One of the solutions to this problem is to automatically organise documents into topic groups. With regard to machine learning tasks, automatic document classification has been widely applied for this purpose. Document classification is usually more challenging than numerical data classification, because it is much more difficult to effectively represent documents than numerical data for classification purposes. Document representation is usually based on weighting terms in a document to indicate their importance within the document [9]. The vector space model (VSM) is fundamental for representing a document for classification tasks. It represents a set of documents as a set of vector in a common vector space [2]. A single word is the traditional and most popular grammatical unit in document representation techniques. Most document representation approaches use a bag-of-words (BOW) as the original sources for deriving the representation [24] [27] [28] [29] [30]. The BOW model focuses on the number of occurrences of each term in a document; however, the exact ordering of the terms is ignored [2]. The main problem of VSM is due to the loss of semantic representation. Therefore, many recent studies aim to solve this problem by exploiting semantic features from knowledge bases.

Apart from its applications in search engines, document classification techniques have been applied to other areas such as spam filtering [6], email routing [7], and genre classification [8]. There are widely used classifiers such as

k-nearest neighbours (kNN) [27] [49], support vector machine (SVM) [27], and linear discriminant analysis (LDA) [73] [74]. These classifiers may suffer from overfitting or underfitting problems, especially in document classification where documents usually have to be represented in very high dimensional feature spaces. As the dimensionality of the data increases, many data analysis and classification problems become significantly harder [153]. This chapter presents a new weighting method for document representation to improve classification performance under the VSM framework.

3.2 Baseline document representation methods

Baseline term weighting methods used in the experiment include term frequency (*TF*), normalised term frequency (*norTF*), term presence (*TP*), term frequency and inverse document frequency (*TFIDF*) and normalised TF-IDF (*norTFIDF*). The last four methods are defined by equations (3.1), (2.1), (2.3), and (3.2), respectively.

$$norTF_{ji} = \frac{TF_{ji}}{\max TF_j}, \quad 0 < norTF_{ji} < 1 \tag{3.1}$$

$$norTFIDF_{ji} = \frac{TFIDF_{ji}}{\max TFIDF_j}, \quad 0 < norTFIDF_{ji} < 1 \tag{3.2}$$

where $norTF_{ji}$ and $norTFIDF_{ji}$ are the normalised version of TF and TF-IDF of term i in document j, and $maxTF_j$ and $maxTFIDF_j$ are the maximum TF value and the maximum TF-IDF value in document j.

Miloš Radovanović and Mirjana Ivanović [32] have described the impact of the BOW document representation for short web-page descriptions. Their experimental results show that stemming generally improved classification performance and logarithm led to performance improvement too. From their

findings, all documents in this thesis are stemmed in the pre-processing step, and logarithm TF-IDF is used as the baseline method. In addition, all the weighting scores are transformed into the normalised form.

3.3 Term frequency relevance frequency (TFRF)

TFRF is a supervised term weighting method proposed by Lan et al. [47]. The basic idea is to focus more on the high-frequency terms in the positive category than in the negative category. RF and TF.RF are defined by equations (3.3) and (3.4), respectively.

$$RF = \log(2 + \frac{a}{\max(1,c)}) \tag{3.3}$$

$$TF.RF = TF * RF \tag{3.4}$$

where *TF* is the term frequency of a term, *a* is the number of documents in the positive category that contain this term, and *c* is the number of documents in the negative category. In this thesis, *TF.RF* was implemented as equation (3.5).

$$TF.RF_{ik} = TF_i * \log_2(2 + \frac{DF_{ik}}{\max(1,DF_i - DF_{ik})}) \tag{3.5}$$

where $TF.RF_{ik}$ is the TFRF value of term *i* in class *k*, TF_i is the term frequency of term *i*, DF_{ik} is the document frequency of term *i* in class *k*, and DF_i is the document frequency of term *i* in the whole dataset.

Since the values of DF_i and DF_{ik} are not supposed to be known in testing data, the TFRF value of term *i* in both training and testing data is defined as the variance of the original TFRF values of term *i* in class *k*, i.e.,

$$TF.RF_i = \text{var}(TF.RF_{ik}) \tag{3.6}$$

3.4 Class specific document frequency (CSDF)

The related research about term weighting and text classification has been published in recent years. For example, Ren and Sohrab [45] have introduced a class-indexing-based term weighting method for automatic text classification. The method is incorporated with term index, document index, and class index. It has been tested in both high-dimensional and low-dimensional vector spaces in comparison with TF-IDF and five other different term weighting approaches. Their experimental results show that the proposed method outperformed the six term weighting approaches. However, this method required more space to store data, and the computation cost is high.

This thesis proposes a new supervised term weighting technique for document representation which is which is called the class specific document frequency (CSDF). In our exploration for new features for document representation, various ideas for term weighting have been investigated, which led to our belief that class specific document frequency values of terms in a document contain critical information for document classification. The basic idea underlying CSDF is that a term in a document is meaningful for classifying documents if it is more frequent inside the document and other documents belonging to the same class, but less frequent in documents belonging to different classes. The CSDF value of term *i* in class k is calculated as follows:

$$CSDF_{ik} = \begin{cases} \frac{DF_{ik}/N_k}{(DF_i - DF_{ik})/(N - N_k) + 1}, & \text{if } TF_{ik} > 0 \\ 0, & \text{otherwise} \end{cases} \quad (3.7)$$

where DF_{ik} is the document frequency of term *i* based on the documents in the training document set and in class k, DF_i is the document frequency of term *i* based on all the documents in the training document set, N_k is the number of

documents in the training document set and in class *k*, and *N* is the number of documents in the training document set. This definition is an estimator for $\frac{P(term|class)}{P(term|not\ class)}$ which seems to be similar to the likelihood ratios in diagnostic testing [154] [155] and the lift in association rule mining [156] [157]. However, they are different in many ways. Firstly, they are used for different purposes and in different applications. Secondly, the definitions in these methods are different. Regarding the likelihood ratios, there are at least two likelihood ratios: positive likelihood ratio (LR+) and negative likelihood ratio (LR-). The positive LR is defined by $LR+ = \frac{P(T+|D+)}{P(T+|D-)}$, which is the probability that an individual with disease has a positive test divided by the probability that an individual without disease has a positive test. On the other hand, the negative LR is defined by $LR- = \frac{P(T-|D+)}{P(T-|D-)}$, which is the probability that an individual with disease has a negative test divided by the probability that an individual without disease has a negative test. Both LR+ and LR- are focused on its own class, neither a positive class nor a negative class. Therefore, LR+ or LR- is compatible only with $\frac{DF_{ik}}{N_k}$ in our method which is only focused on class k. Furthermore, the lift method in association rule mining is denoted by $L(A => B) = \frac{P(B|A)}{P(B)}$ or the proportion of the transactions that contain A which also contains B divided by the proportion of transactions which contain B. There are some duplicate counts of value B in the dividend and the divisor in the lift's equation. However, in our method, term *i* in the divisor and dividend is independent and the total number of documents is not calculated in the dividend. Instead, only the rest after the number of documents in class k has been removed is used.

Since the values of DF_{ik} and N_k are not supposed to be known in testing data, the CSDF value of term *i* in both training and testing data is defined as the variance of the original CSDF values of term *i* in class *k*, i.e.,

$$CSDF_i = \text{var}(CSDF_{ik}) \tag{3.8}$$

In order to derive a more effective term weighting scheme, this research also proposes to combine term frequency and CSDF. The combined feature TF-CSDF of term *i* in document *j* is defined by equation (3.9):

$$TF - CSDF_{ji} = \propto CSDF_i + (1-\propto) norTF_{ji}\ ,\ 0 <\propto< 1 \tag{3.9}$$

where $\propto$ is a weighting factor which aims to find the best tradeoff between two features [158].

3.5 Semantic information for VSM-based document representation and classification

TF, TP, TF-IDF, and CSDF are statistical term weighting methods. These methods are related to a quantity of terms that appear in a document or group of documents. Each term in the document is regarded as independent of each other. Another type of term weighting methods is semantic term weighting. Each term in the document is related to other terms such as class name or query terms. Semantic technologies allow the usage of features on a higher semantic level rather than single words for text classification purposes. The classic document representations are enhanced through concepts extracted from background knowledge or ontology [57] [58] [59]. Ontology is an explicit knowledge source such as WordNet, Wikipedia, ODP and YAGOs [60] [61] [159]. These ontologies are used for the extraction of conceptual or semantic features for text documents.

Recently, semantic information has been used as an important feature for text classification. A lot of research about semantic term weighting has been published, such as Yang et al. [56], Bloehdorn and Hotho [57], Peng and Choi [62], Ferretti et al. [63], and Nagaraj et al. [64]. These publications reported in almost the same way that the inclusion of semantic information improves text classification performance. Therefore, this chapter also investigates semantic information for text classification, in comparison with statistical term weighting methods.

3.5.1 Semantic representation

Semantic information can be extracted from a knowledge base or ontology such as WordNet. In classification tasks, it is assumed that some words or terms are more closely related to the target category. The experiments in this research use WordNet as a knowledge base, with path similarity measuring relationships between words. It is a lexical database for the English language, which was created by Princeton, and is part of the NLTK corpus. Path similarity returns a score denoting how similar two word senses are, based on the distance between the two synsets in the WordNet hierarchy [160]. This distance is the shortest path that connects the senses in the is-a (hypernym/hyponym) taxonomy.

In linguistics, a hyponym is a word or phrase whose semantic field is included within that of another word, its hyperonym or hypernym. In simpler terms, a hyponym shares a type of relationship with its hypernym. For example, cat and dog are all hyponyms of animal (their hypernym) which is a hyponym of creature. Figure 3.1 gives an example of the relationship between hyponyms and hypernym [161].

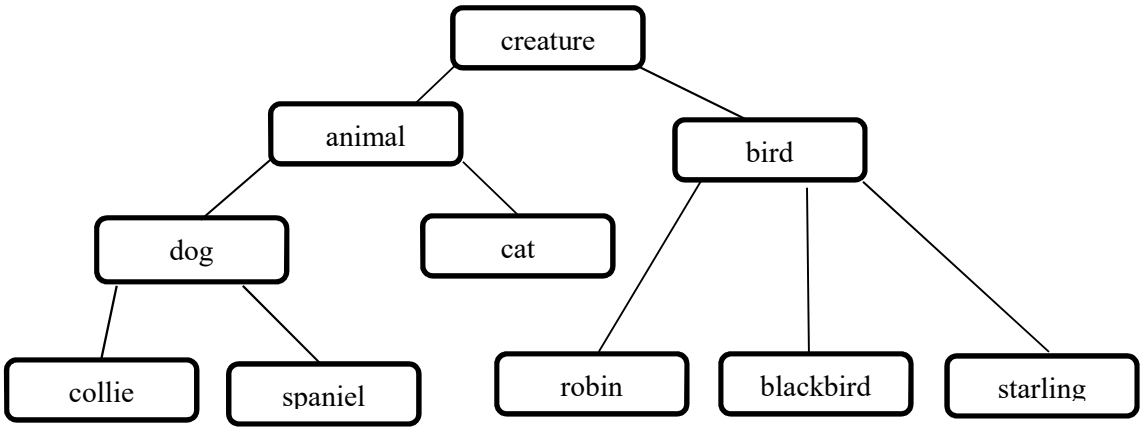


Figure 3.1 An example of the relationship between hyponyms and hypernym [161]

The command "X = synset1.path_similarity(synset2)" is used to return a similarity score between synset1 and synset2. An example of using a path_similarity function in WordNet is shown as follows:

Dog = wn.synset('dog.n.01')

Cat = wn.synset('cat.n.01')

X = Dog.path_similarity(Cat)

where 'dog.n.01' is a synonym set of word 'dog', 'cat.n.01' is a synonym set of word 'cat' and the similarity score is in the range of 0 to 1. X will be 0 if a path could not be found and none was returned. In contrast, X will be 1 if two synsets represent identity, i.e., comparing a sense with itself [162].

This chapter investigates the performance of classification using semantic information as well. These semantic features are extracted from the calculation of path similarity scores between a word in a document and the representative words of classes. In training data, each word has a path similarity score which is calculated from the comparison between itself and the representative word which

is considered to be the best representation of each class. In the case of three representative words, each word is compared to all the selected words, and then a maximum score is chosen. For instance, the semantic score of term *i* in class *k* ($Semantic_{ik}$), chosen from the maximum path similarity score comparing three representative words (*k1, k2, k3*) of class *k*, is defined by equation (3.10):

$$Semantic_{ik} = \max(path\_sim_{k1}, path\_sim_{k2}, path\_sim_{k3}) \quad (3.10)$$

Because we do not know the test document's class, the semantic value of the term *i* in both training and testing data is defined as the variance of the original semantic values of term *i* in class *k* , i.e.,

$$Semantic_i = \text{var}(Semantic_{ik}) \quad (3.11)$$

It is noteworthy that if the representative words for each class are given, there is no need of class labels of training documents in using semantic information for document classifications.

3.5.2 Class prediction using semantic information

Chang et al. [163] have introduced dataless classification, a learning protocol that uses world knowledge (semantic) for class prediction without training any labelled training data. They believe that people can categorize documents into their class without any training because we know the meaning of class names. Therefore, semantic information can predict a document's class if we know class names and their properties. Because of this idea, the classes of test documents are predicted using the representative words of each class, which can be determined by knowledge analysis of training documents if they are available. For example, the five representative words of each class in the Reuters dataset were identified and used in our experiment to predict classes of testing documents. These words

are determined by the name of class and the most frequent terms in each class, which are shown in Table 3.1.

Table 3.1 Five representative words of each class

Class	The representative words
Coffee	coffee, meeting, export, brazil, bag
Corn	corn, department, agriculture, export, grain
Dlr	dollar, currency, exchange, yen, bank
Gnp	gross, economy, product, rate, growth
Gold	gold, mine, silver, company, price
Money-supply	money, supply, deposit, week, reserve
Oilseed	agriculture, soybean, export, trade, grain
Ship	ship, spokesman, cargo, union, port
Sugar	sugar, sweetening, community, tender, trade
Wheat	wheat, export, agriculture, grain, trade

An example of the class prediction process using path similarity scores is illustrated in Figure 3.2. Start from the path similarity scores (e.g., V11,…,V15) of each word (e.g., W1) in a testing document (e.g., D1), which are calculated with the five representative words (e.g., K11,…,K15) of each class (e.g., C1). The maximum of all words in the document path similarity scores (e.g., M11,…,Mn1) of each class (e.g., C1) are chosen. After that, maximum scores of all words in the document of each class are added up (e.g., S1). Finally, the class which has the maximum sum score is the predicted class of the document.

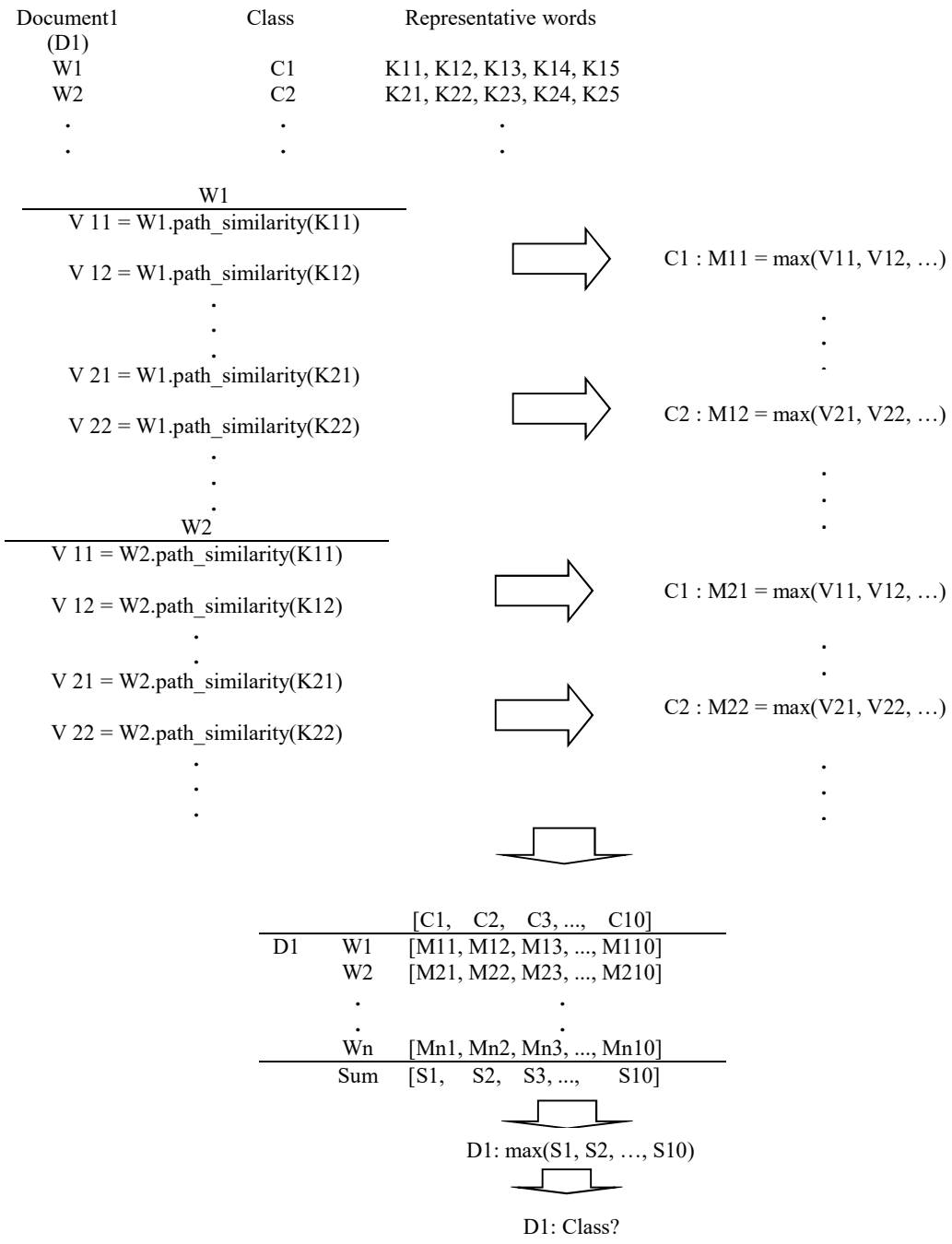


Figure 3.2 The prediction process on test set

3.6 Classifier fusion

Decision fusion or classifier fusion is the combination of classifiers to achieve better classification accuracy in pattern recognition problems [85]. In this chapter, classifier fusion is by majority vote of the predicted labels of classifiers using

different document representation features, such as the baseline representation (TF, TP, and TF-IDF), CSDF, and semantic information.

3.7 Experiments and results

3.7.1 Experimental procedure

To evaluate the proposed method properly, a variety of datasets and various feature selection methods were adopted in the evaluation. Two benchmark document datasets: Reuters-21578 [164] and 20newgroups [165], and a set of web documents returned by Google, were used in the experiments. For the first dataset, the documents from Reuters-21578 were separated into training data and testing data using the standard "modApté" train and test split. The training dataset was used to train classifiers and select features, and the testing dataset to evaluate the performance of the trained classifiers. In order to have sufficient documents for each class, only 10 almost balance classes of documents were adopted in these experiments, with 1,415 documents for training data and 470 documents for testing data, the details are shown in Table 3.2.

Table 3.2 Training and testing datasets of Reuters-21578

No.	Class name	Training	Testing
1	Coffee	111	28
2	Corn	181	56
3	Dlr	131	44
4	Gnp	101	35
5	Gold	94	30
6	Money-supply	138	34
7	Oilseed	124	47
8	Ship	197	89
9	Sugar	126	36
10	Wheat	212	71
Total		1,415	470

The 20newgroups dataset [165] is a benchmark dataset in document classification research. It contains almost 20,000 documents from 20 news topics. The 20 topics can be categorized into seven top-level categories with related news: alternative (alt), computers (comp), miscellaneous (misc), recreation (rec), science (sci), sociology (soc), and talk [166]. From Jason [165], these documents are already separated into training and testing folders. There are 11,314 documents for the training set and 7,532 documents for the testing set, as shown in Table 3.3.

Table 3.3 Training and testing datasets of 20newsgroups

No.	Class name (20)	Class name (7)	Training	Testing
1	alt.atheism	Alt	480	319
2	comp.graphics	Comp	584	389
3	comp.os.ms-windows.misc		591	394
4	comp.sys.ibm.pc.hardware		590	392
5	comp.sys.mac.hardware		578	385
6	comp.windows.x		593	395
7	misc.forsale	Misc	585	390
8	rec.autos	Rec	594	396
9	rec.motorcycles		598	398
10	rec.sport.baseball		597	397
11	rec.sport.hockey		600	399
12	sci.crypt	Sci	595	396
13	sci.electronics		591	393
14	sci.med		594	396
15	sci.space		593	394
16	soc.religion.christian	Soc	599	398
17	talk.politics.guns	Talk	546	364
18	talk.politics.mideast		564	376
19	talk.politics.misc		465	310
20	talk.religion.misc		377	251
		Total	11,314	7,532

The documents returned by Google were collected at University of Essex using Google search API which allows retrieving and displaying search results from Google. Due to the API limits the number of search returned results, only the titles and snippets of the top 56 Google returned documents were considered. For evaluation purposes, 80 queries were selected from eight popular search topics

(categories), as shown in Table 3.4. Each category contains 10 queries consisting of one to three words that are commonly known and convenient for user evaluation. These documents are separated into two sets; approximately 60% and 40% of whole documents are in training dataset and testing dataset, respectively. The details of this dataset are shown in Table 3.5.

Table 3.4 Categories of queries

No.	Class name	Number of queries
1	Animal	10
2	Art	10
3	Flower	10
4	Food	10
5	Movie	10
6	Shopping	10
7	Sport	10
8	Travel	10
	Total	80

Table 3.5 The number of web returned documents in training and testing datasets

No.	Class name	All	Training	Testing
1	Animal	558	334	224
2	Art	560	335	225
3	Flower	558	335	223
4	Food	555	335	220
5	Movie	560	335	225
6	Shopping	559	335	224
7	Sport	558	335	223
8	Travel	556	335	221
	Total	4,464	2,679	1,785

With the Reuters dataset, the experiments were conducted in three steps: pre-processing, feature selection, and document representation and classification. The pre-processing is to remove stop words and unnecessary contents. Table 3.6 illustrates an initial experiment to choose the parts of speech for classification. Weka [167] has been used as a classification tool. For feature selection, the top 100 features have been selected using information gain scores.Four classifiers

were used in this experiment: J48 (C4.5, a decision tree algorithm), IBk (kNN), NB (Naïve Bayes algorithm), and SMO (Support Vector Machine). The experimental results show that using the original content achieved the best performance using IBk classifier, whilst using only noun or noun+verb achieved the best performance using J48 classifier. For NB and SMO classifiers, using only noun or noun+verb achieved the best classification accuracy, respectively. To sum up, using noun+verb achieved the best performance with J48 and SMO classifiers whilst using only noun achieved the best performance with J48 and NB classifiers. Since the most selective terms should be nouns [10] [168], only nouns were considered in this experiment. After that, all words were stemmed or derived to their stems or root forms. The words kept after pre-processing were the sources of feature selection.

Table 3.6 The classification accuracy of using different parts of speech

	Classification accuracy (%)			
	J48	IBk	NB	SMO
Original	77.0213	**52.766**	70.2128	78.9362
Noun+Verb	**78.7234**	52.3404	73.8298	**83.1915**
Noun+Adj	65.7975	44.3252	65.1840	73.7730
Noun	**78.7234**	49.1489	**74.6809**	81.9149

With regard to the Reuters dataset, more than 5,000 words as initial features were extracted from 1,415 documents. Only 1,415 documents as samples cannot be representative in a space with over 5,000 features. Therefore, feature selection is necessary in this case [10] [153]. Stefan Bordag [168] has presented a comparison of co-occurrence and similarity measures for term selection. Only the most significant co-occurrences were used to find new feature candidates. This assumed that higher frequency means higher significance. In our experiments, a suitable number of features was determined by comparing the performances of

using different number of features, different number of document representation and different classifiers on a small part of the dataset. The top 25 terms (producing 120 features) and top 50 terms (producing 242 features) with the highest document frequency values in each class were compared in terms of the average F-measure scores on kNN and SVM classifiers. The validation data was from three categories: corn, oilseed, and wheat. The experimental results are shown in Table 3.7, Figures 3.3 and 3.4, indicating that using the top 25 terms for each class achieved similar or better F-measure scores than using the top 50 terms in both TF-based and TP-based document representations. Therefore, the selected features in this experiment were the terms that have the top 25 document frequency values in each class of documents. With duplicate terms removed, only 120 features were selected from 250 terms with high document frequency values for the 10 classes.

Table 3.7 F-measure scores with different number of features on Reuters dataset

Document Representation	Class	F-measure			
		kNN		SVM	
		120 features	242 features	120 features	242 features
TF	Corn	0.460	0.427	0.537	0.566
	Oilseed	0.309	0.328	0.364	0.654
	Wheat	0.511	0.500	0.680	0.471
TP	Corn	0.533	0.492	0.641	0.647
	Oilseed	0.394	0.356	0.519	0.564
	Wheat	0.628	0.585	0.716	0.731

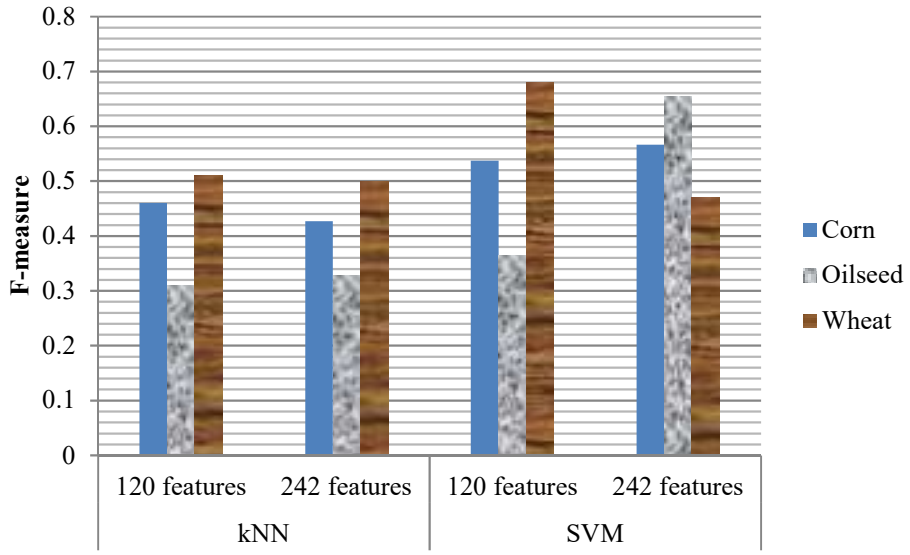


Figure 3.3 F-measure scores of TF-based document representation

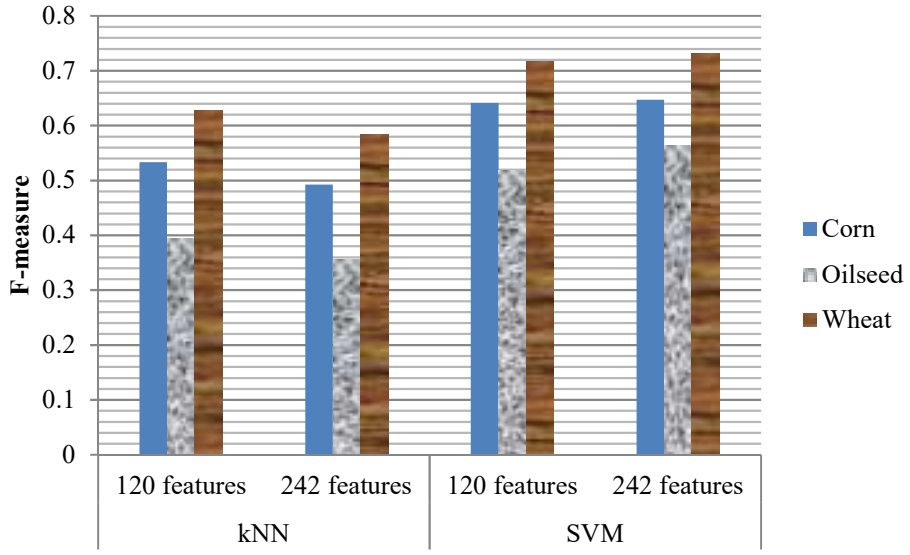


Figure 3.4 F-measure scores of TP-based document representation

For the 20newsgroup dataset, the documents were pre-processed in the same way as for the Reuters dataset. Feature selection using filter approach was applied, with two criteria: document frequency scores of each class and information gain scores. The three different numbers of features, which were selected from the top 50 terms (144 features), top 100 terms (585 features), and top 200 terms (1097 features) that have the highest document frequency scores in each class of

documents, were compared. Their classification accuracies on SVM classifiers using TF-based and TP-based document representations were compared as well. The experimental results are shown in Table 3.8 and Figure 3.5, indicating that the features selected from the top 100 terms for each class (585 features) achieved almost the same performance as those selected from the top 200 terms (1097 features). Therefore, 585 features were adopted in this experiment.

Table 3.8 Classification accuracy with different number of features on 20newsgroups

class	No. of features	Accuracy (%)	
		TF	TP
7 classes	144	58.38	72.72
	585	68.56	77.02
	1097	71.46	77.55
20 classes	144	35.05	54.58
	585	50.62	60.79
	1097	55.14	62.28

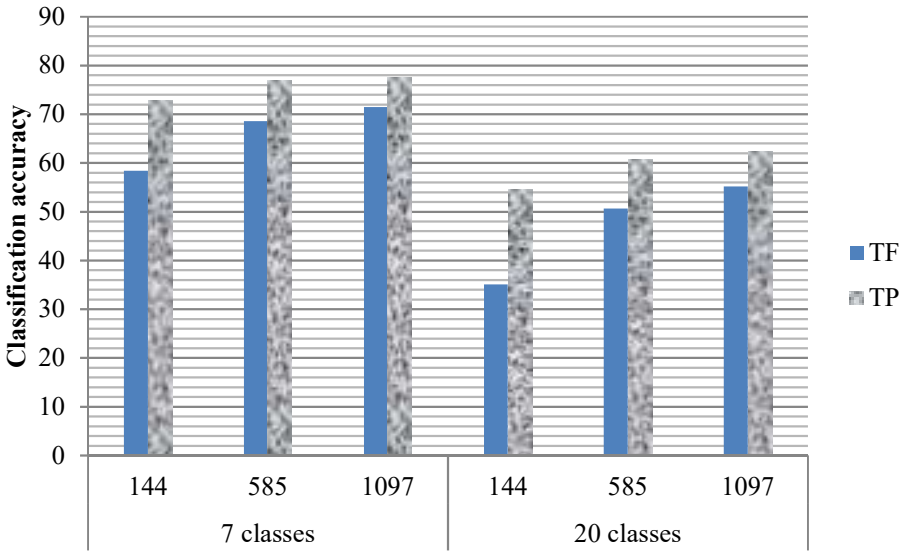


Figure 3.5 Classification accuracy with different number of features on 20newsgroups

Regarding the web document dataset, each document was pre-processed as follows. Firstly, only the title and snippet content (short description) in each document were considered. After that, all HTML tags were removed and all contents were split into tokens, with only nouns selected. The features were selected with both filter approach and wrapper approach. The features were initially selected from the top 40 terms with the highest document frequency scores in each class of documents, resulting in a total of 258 features without duplication, which were further selected using sequential forward floating search (SFFS) method with LDA classifier [71].

For semantic information, the path similarity function of WordNet was implemented by the NLTK library in Python programming [162]. The classifiers tested in our experiments were kNN, SVM, LDA, Naïve Bayes, and logistic regression. Tables 3.9, 3.10, and 3.11 show one word or three words representing semantic information of each class of the Reuters dataset, 20newgroups dataset, and web returned documents, respectively. These words were selected from class names and the related words in each class. Normally, the representative words can be selected from the words which have the highest document frequency values in documents of each class, if labelled training documents are available. Otherwise, they can be chosen based on knowledge only.

Table 3.9 Representative words of each class of Reuters dataset

Class	The representative words	
	1 word	3 words
Coffee	Coffee	coffee, export, brazil
Corn	Corn	corn, maize, grain
Dlr	Dollar	dollar, currency, yen
Gnp	Gnp	gross, economy, product
Gold	Gold	gold, mine, silver
Money-supply	Money	money, supply, growth
Oilseed	Oilseed	soybean, production, trade
Ship	Ship	ship, vessel, port
Sugar	Sugar	sugar, production, trade
Wheat	Wheat	wheat, export, agriculture

Table 3.10 Representative word s of each class of 20newsgroups dataset

Class (20)	The representative words	
alt.atheism	atheist, god, people	atheist, god, people
comp.graphics	graphic, image, version	computer, problem, system
comp.os.ms-windows.misc	window, system, driver	
comp.sys.ibm.pc.hardware	card, controller, drive	
comp.sys.mac.hardware	mac, apple, machine	
comp.windows.x	window, application, server	
misc.forsale	sale, offer, price	sale, offer, price
rec.autos	car, engine, dealer	game, car, bike
rec.motorcycles	dod, bike, motorcycle	
rec.sport.baseball	game, baseball, team	
rec.sport.hockey	term,game,hockey	
sci.crypt	clipper, chip, encryption	science, power, clipper
sci.electronics	power, use, circuit	
sci.med	doctor, case, disease	
sci.space	space, orbit, moon	
soc.religion.christian	god, church, life	god, church, people
talk.politics.guns	gun, weapon, law	state, government, people
talk.politics.mideast	government, right, policy	
talk.politics.misc	state, government, law	
talk.religion.misc	people, god, religion	

Table 3.11 Representative words of each class of web document dataset

Class	The representative words
Animal	sea, animal, wild
Art	art, painting, history
Flower	flower, plant, tulip
Food	recipe, food, rice
Movie	movie, trailer, book
Shopping	shop, body, fashion
Sport	sport, golf, football
Travel	museum, travel, bridge

3.7.2 Experimental results

TF, norTF, TP, TFIDF, norTFIDF, were tested in the experiments as baseline methods, while TFRF was tested as a state-of-the-art supervised term weighting method to evaluate the performance of the proposed methods: CSDF, TF-CSDF, and semantic representation using WordNet. Class prediction using semantic information was also investigated in the experiments. The classification performances of different features for document representation were compared, using the two sample t-test statistical test with $p \leq 0.05$ as significance level.

3.7.2.1 Reuters dataset

Tables 3.12 and 3.13 show the classification performances of kNN, LDA, SVM, Naïve Bayes (NB), and logistic regression (LG) with different types of features for document representation. The experimental results show that the proposed method, CSDF, achieved the highest classification accuracy using LDA, SVM, NB, and LG. On the other hand, TFRF achieved the best performance using kNN classifier, followed by semantic feature. Therefore, CSDF was demonstrated as the best method on the Reuters dataset in general.

Table 3.12 Experimental results on Reuters dataset (I)

Classifier's performance / Document representation	kNN		LDA		SVM	
	Classification accuracy (%)	F_1	Classification accuracy (%)	F_1	Classification accuracy (%)	F_1
TF	68.51	0.6903*	75.96	0.7646*	73.62	0.7404*
norTF	70.00	0.7142*	77.87	0.7879*	78.98	0.8075*
TP	68.72	0.7033*	81.06	0.8232	81.70	0.8328*
TF-IDF	74.47	0.7514	79.15	0.8022	80.21	0.8135*
CSDF	73.62	0.7478	**81.70**	**0.8296**	**83.19**	**0.8426**
Semantic	76.81	0.7777	80.85	0.8214	80.64	0.7810*
TFRF	**78.30**	**0.7939**	81.06	0.8231	81.7	0.8328*

Table 3.13 Experimental results on Reuters dataset (II)

Classifier's performance / Document representation	NB		LG	
	Classification accuracy (%)	F_1	Classification accuracy (%)	F_1
TF	64.68	0.6360*	70.00	0.6970
norTF	69.57	0.6970	72.98	0.7250
TP	75.11	0.7520	73.62	0.7310
TF-IDF	68.94	0.6820*	72.77	0.7230
CSDF	**76.17**	0.7610	**74.25**	**0.7370**
Semantic	**76.17**	**0.7620**	72.34	0.7220
TFRF	75.11	0.7520	73.83	0.7330

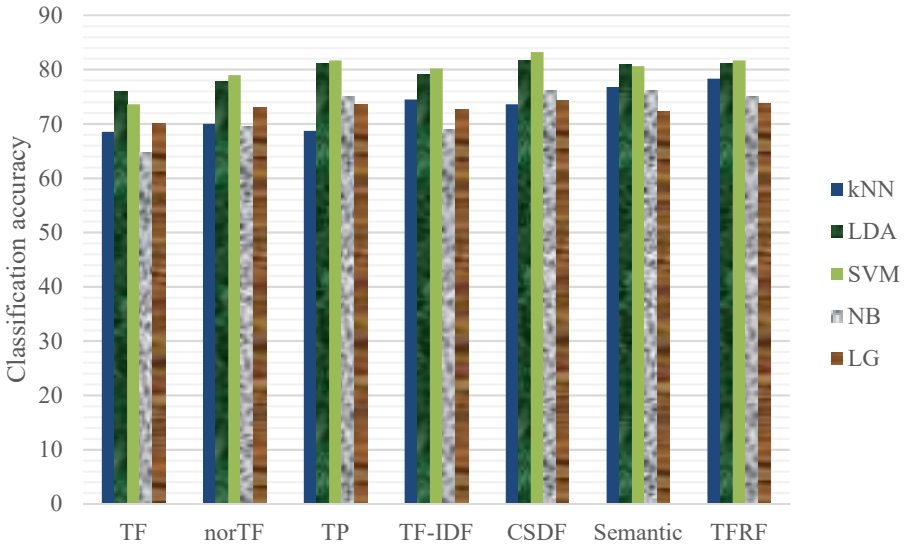


Figure 3.6 Classification accuracy with different types of features on Reuters dataset

Table 3.14 illustrates F_1 score of each class using different types of features for document representation with kNN classifier. The experimental results show that TFRF achieved the best average score which was significantly better than that of TF, norTF, and TP, but not significantly better than that of CSDF. Furthermore, Tables 3.15 and 3.16 illustrate F_1 score of each class using different types of features with LDA and SVM classifier, respectively. The experimental results

show that CSDF achieved the best average score with both classifiers, which was significantly better than that of TF and norTF with LDA classifier, and significantly better than that of all other features including TFR with SVM classifier.

Tables 3.17 and 3.18 illustrate F_1 score of each class using different types of features with Naïve Bayes and logistic regression classifiers, respectively. The experimental results show that semantic features achieved the best average score which was significantly better than that of TF and TFIDF, whilst CSDF score achieved significantly better score than TF, TP, TFIDF, and TFRF with Naïve Bayes classifier. There was no significant difference in performance achieved by CSDF and semantic representation. In addition, the average score achieved by CSDF was the best with logistic regression classifier. To sum up, CSDF was the best document representation on Reuters dataset.

Table 3.14 F_1 scores of kNN classifier on Reuters dataset

Feature / Class	TF	norTF	TP	TFIDF	CSDF	Semantic	TFRF
Coffee	0.8670	0.8810	0.8470	0.8360	0.8060	0.8462	0.9818
Corn	0.5330	0.5430	0.5330	0.5430	0.5380	0.6621	0.6187
Dlr	0.8880	0.8640	0.8000	0.8670	0.8510	0.8211	0.8478
Gnp	0.8290	0.7800	0.8770	0.8500	0.7820	0.8611	0.8108
Gold	0.6380	0.8240	0.6960	0.8790	0.8880	0.9206	0.9831
Money-supply	0.6670	0.8440	0.8050	0.8280	0.8020	0.8611	0.7647
Oilseed	0.3970	0.3610	0.3940	0.3990	0.5440	0.4578	0.4638
Ship	0.8020	0.8400	0.8300	0.8400	0.8070	0.8772	0.8772
Sugar	0.6830	0.6380	0.6230	0.7940	0.7830	0.8267	0.8642
Wheat	0.5990	0.5670	0.6280	0.6780	0.6770	0.6429	0.7273
Average	0.6903	0.7142	0.7033	0.7514	0.7478	0.7777	0.7939
T-test (TFRF)	**0.0135**	**0.0213**	**0.0286**	0.0701	0.0765	0.4741	-
T-test (CSDF)	0.1068	0.2529	0.1732	0.8446	-	0.1652	0.0765

Table 3.15 F_1 scores of LDA classifier on Reuters dataset

Feature / Class	TF	norTF	TP	TFIDF	CSDF	Semantic	TFRF
Coffee	0.9470	0.9640	0.9820	0.9820	0.9820	0.9818	0.9818
Corn	0.5950	0.5770	0.5980	0.5890	0.5980	0.6226	0.5979
Dlr	0.8570	0.8570	0.8670	0.9110	0.8700	0.8864	0.8667
Gnp	0.8800	0.8310	0.9040	0.9170	0.9010	0.8889	0.9041
Gold	0.8440	0.8920	0.9670	0.9060	0.9670	0.9667	0.9667
Money-supply	0.6740	0.8410	0.8770	0.7410	0.9140	0.8267	0.8767
Oilseed	0.5620	0.5180	0.5320	0.5240	0.5470	0.5361	0.5319
Ship	0.9120	0.8900	0.9110	0.8900	0.9230	0.9162	0.9111
Sugar	0.7530	0.8210	0.8640	0.8680	0.8640	0.8642	0.8642
Wheat	0.6220	0.6880	0.7300	0.6940	0.7300	0.7244	0.7299
Average	0.7646	0.7879	0.8232	0.8022	0.8296	0.8214	0.8231
T-test	**0.0292**	**0.0003**	0.1324	0.1745	-	0.4117	0.1287

Table 3.16 F_1 scores of SVM classifier on Reuters dataset

Feature / Class	TF	norTF	TP	TFIDF	CSDF	Semantic	TFRF
Coffee	0.9230	0.9820	0.9820	0.9820	0.9820	0.9820	0.9820
Corn	0.5370	0.6470	0.6410	0.6470	0.6410	0.6710	0.6410
Dlr	0.8670	0.8670	0.8970	0.8790	0.9090	0.8640	0.8970
Gnp	0.9010	0.8570	0.8920	0.8890	0.9170	0.8730	0.8920
Gold	0.8680	0.9290	0.9830	0.9290	0.9830	0.9830	0.9830
Money-supply	0.6140	0.8610	0.8990	0.8530	0.9120	0.8150	0.8990
Oilseed	0.3640	0.5000	0.5190	0.5260	0.5370	0.5110	0.5190
Ship	0.8440	0.8880	0.9220	0.8860	0.9270	0.8880	0.9220
Sugar	0.8060	0.8530	0.8770	0.8570	0.8920	0.8540	0.8770
Wheat	0.6800	0.6910	0.7160	0.6870	0.7260	0.6400	0.7160
Average	0.7404	0.8075	0.8328	0.8135	0.8426	0.7810	0.8328
T-test	**0.0034**	**0.0006**	**0.0054**	**0.0021**	**-**	**0.0198**	**0.0054**

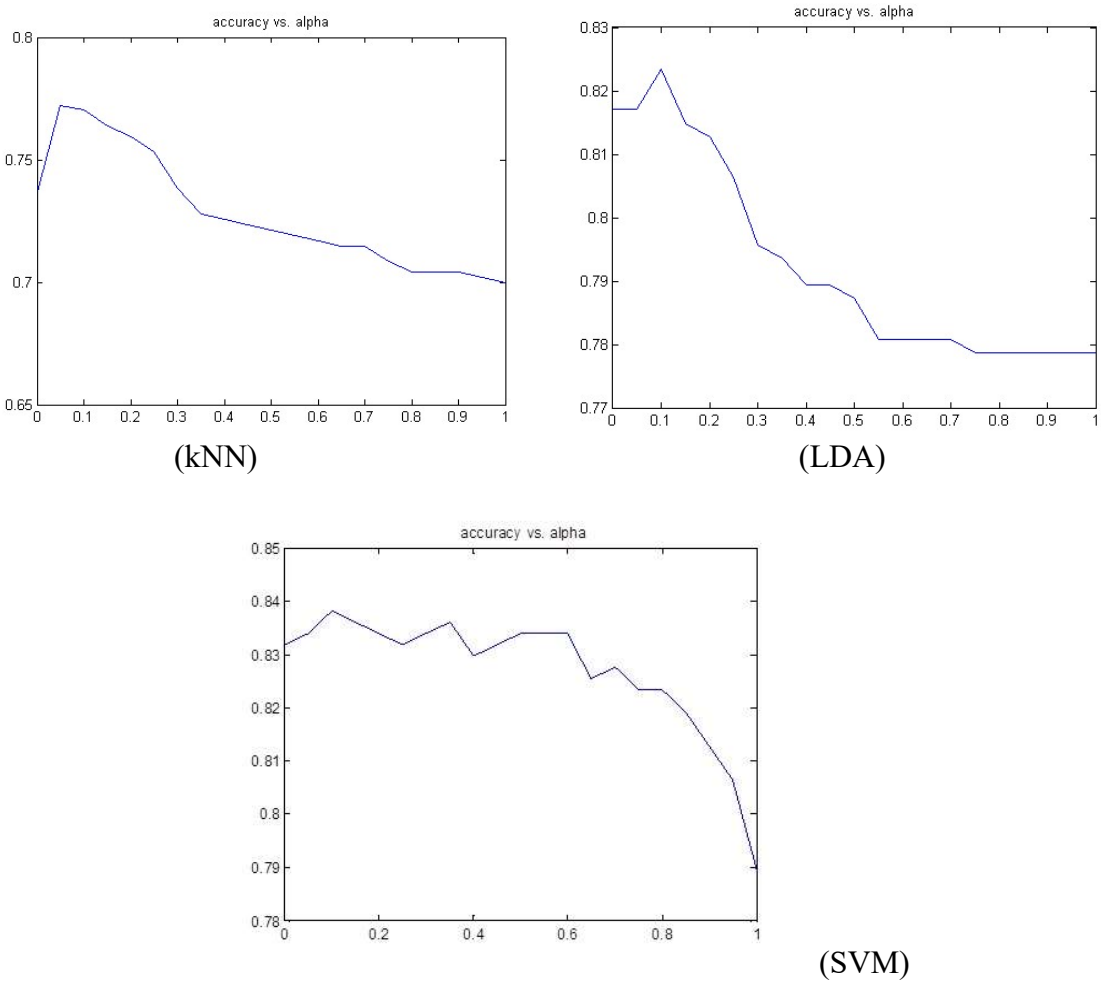
Table 3.17 F_1 scores of Naïve Bayes classifier on Reuters dataset

Feature / Class	TF	norTF	TP	TFIDF	CSDF	Semantic	TFRF
Coffee	0.7450	0.8770	0.9060	0.8070	0.9060	0.9230	0.9060
Corn	0.3430	0.4860	0.5380	0.4040	0.5380	0.5690	0.5380
Dlr	0.7950	0.8670	0.8510	0.8310	0.8510	0.8570	0.8510
Gnp	0.8460	0.8500	0.7640	0.8290	0.7820	0.7730	0.7640
Gold	0.9120	0.9330	0.9290	0.9490	0.9470	0.9090	0.9290
Money-supply	0.6670	0.8240	0.7690	0.7730	0.8000	0.7650	0.7690
Oilseed	0.2370	0.4130	0.5190	0.3610	0.5440	0.5160	0.5190
Ship	0.8020	0.8500	0.8950	0.8370	0.9010	0.8960	0.8950
Sugar	0.6130	0.6000	0.7650	0.6060	0.7830	0.8290	0.7650
Wheat	0.5600	0.4960	0.6770	0.5880	0.6770	0.6810	0.6770
Average	0.6360	0.6970	0.7520	0.6820	0.7610	0.7620	0.7520
T-test (Semantic)	**0.0068**	0.1355	0.1844	**0.0347**	0.9032	-	0.1844
T-test (CSDF)	**0.0041**	0.0821	**0.0124**	**0.0132**	-	0.9032	**0.0124**

Table 3.18 F_1 scores of logistic regression classifier on Reuters dataset

Feature / Class	TF	norTF	TP	TFIDF	CSDF	Semantic	TFRF
Coffee	0.7210	0.8070	0.8970	0.8280	0.9290	0.8280	0.9290
Corn	0.5740	0.5490	0.5250	0.5890	0.5310	0.5160	0.5250
Dlr	0.8180	0.8740	0.8080	0.7450	0.8130	0.8130	0.8200
Gnp	0.7690	0.8060	0.7890	0.9090	0.8120	0.7890	0.7890
Gold	0.9120	0.8210	0.9670	0.8770	0.9830	0.9120	0.9670
Money-supply	0.7620	0.8120	0.7620	0.8360	0.7690	0.7890	0.7620
Oilseed	0.4260	0.5120	0.5100	0.4680	0.5310	0.5090	0.5100
Ship	0.7980	0.8280	0.8290	0.8160	0.8320	0.8180	0.8330
Sugar	0.6670	0.7220	0.6880	0.7760	0.6770	0.6960	0.6880
Wheat	0.6190	0.6310	0.6810	0.5880	0.6720	0.6760	0.6810
Average	0.6970	0.7250	0.7310	0.7230	0.7370	0.7220	0.7330
T-test	0.0564	0.4373	0.0614	0.6689	-	0.1297	0.2769

Figure 3.7 shows the classification performances of the combined method TF-CSDF on the Reuters dataset with $\propto = 0$ to $\propto = 1$. For the kNN classifier, the starting point (only CSDF) achieved 73.62% classification accuracy while the ending point (only normalised TF) achieved 70% accuracy. The highest peak is at $\propto = 0.05$ (i.e., 0.05norTF+0.95CSDF), which reached 77.23% accuracy. At this point, the classification accuracy increased by 3.61% and 7.23% compared to the original CSDF and the normalised TF, respectively. For the LDA classifier, the starting point obtained 81.7% classification accuracy while the ending point obtained 77.87% accuracy. The highest peak is at $\propto = 0.1$ (i.e., 0.1norTF+0.9CSDF), which reached 82.34% accuracy. At this point, the classification accuracy increased by 0.64% and 4.47% compared to the original CSDF and the normalised TF, respectively. For the SVM classifier, the starting point obtained 83.19% classification accuracy while the ending point obtained 78.98% accuracy. The highest peak is at $\propto = 0.1$ (i.e., 0.1norTF+0.9CSDF) which reached 83.83% accuracy. At this point, the classification accuracy increased by 0.64% and 4.85%, compared to the original CSDF and the normalised TF, respectively.



(kNN) (LDA)

(SVM)

Figure 3.7 Experimental results of TF-CSDF

For semantic representation, the experimental results, which were comparable to the baseline representation methods and the CSDF method, are shown in Table 3.19.

Table 3.19 Experimental results of semantic representation

Classifier's performance / Document representation	Classification accuracy (%)	
	LDA	SVM
Semantic (1 kw)	80.85	80.64
Semantic (3 kw)	80.85	80.43
Semantic (prediction with training data)	73.83	73.40
Semantic (prediction without training data)	67.87	

The experimental results show that the classification accuracy of semantic representations was better than that of TF, norTF, and TF-IDF representations, which are shown in Tables 3.12, 3.13, and 3.19. However, it was a little bit lower than the accuracy of TP and CSDF. There is no significant difference between the performances of classification using one or three representative words of each class. Furthermore, the classification accuracies of the predicted classes with labelled training data achieved 73.83% and 73.40% with LDA and SVM classifier, respectively. However, it was less effective than using the normal classification process which achieved over 80% accuracy. The performance of class prediction without using labelled training data was 67.87%. Even though the performance was not too good, this prediction achieved an acceptable performance without using any labelled training data.

With regard to classifier fusion, Table 3.20 shows the classification accuracy of using five different types of features for document representation and some examples demonstrating classification performance improvement by classifier fusion. These features are CSDF, semantic (path similarity scores), TP, normalised TF, and normalised TF-IDF. They were used separately for five classifiers and the decisions of these five classifiers were then fused by majority voting. The performance of the decision fusion from the multiple document representation features was 82.77%, which was almost the same as that of using CSDF feature alone (82.55%), with only 0.43% improvement.

Table 3.20 An example of classifier fusion

Document	Actual	Predicted					
		CSDF	Semantic	TP	norTF	norTF-IDF	Fusion
…							
61	2	2	2	2	2	2	2
62	2	2	7	2	2	7	**2**
63	2	2	2	2	2	2	2
64	2	2	2	2	2	2	2
65	2	2	2	2	2	2	2
66	2	10	10	10	2	10	10
67	2	2	2	2	2	2	2
68	2	2	2	2	2	2	2
69	2	7	2	7	10	7	7
70	2	2	2	2	2	2	2
71	2	2	2	2	2	2	2
72	2	2	2	2	2	2	2
73	2	2	2	2	2	2	2
74	2	10	2	10	2	2	**2**
75	2	2	2	2	2	2	2
76	2	2	2	2	2	2	2
77	2	2	2	2	2	2	2
78	2	7	2	7	2	2	**2**
79	2	10	2	10	2	2	**2**
80	2	9	2	9	2	2	**2**
…							
	Accuracy	82.55	80.43	81.7	79.36	81.49	**82.77**

3.7.2.2 20newsgroups dataset

The 20newsgroups dataset can be separated into two sub-groups: 20 categories and 7 top-level categories. The feature selection of 20newsgroups is based on two criteria: using the top 100 document frequency scores of each class and using information gain (IG) scores. Table 3.21, Figures 3.8 and 3.9 show the classification accuracies of different types of features for document representation with SVM, LDA, and Naïve Bayes classifiers. Logistic regression classifier has not been tested on this dataset because it was too time consuming. The results show that TF-IDF and norTF-IDF were the best representations for SVM classifier, whilst norTF was the best representation for LDA classifier. However, compared with CSDF, the results were not significantly different. Furthermore, two supervised term weighting methods: CSDF and TFRF, which achieved same scores, were the best representations for Naïve Bayes classifier.

Table 3.21 The classification accuracy on 20newsgroups

Document representation	Classification accuracy (%)					
	SVM		LDA		NB	
	20 classes	7 classes	20 classes	7 classes	20 classes	7 classes
TF	50.62	68.56	58.67	60.94	38.13	45.25
norTF	63.30	78.69	**66.52**	**75.40**	54.49	65.44
TP	60.79	77.02	65.69	71.89	61.76	71.64
TF-IDF	**64.98**	78.20	63.87	68.26	52.04	63.41
norTF-IDF	64.87	**79.12**	66.22	74.56	54.78	62.41
CSDF	61.05	77.20	65.61	72.27	**61.78**	**71.91**
Semantic	60.33	74.97	65.06	71.44	61.58	71.75
TFRF	61.05	77.20	65.63	72.27	**61.78**	**71.91**

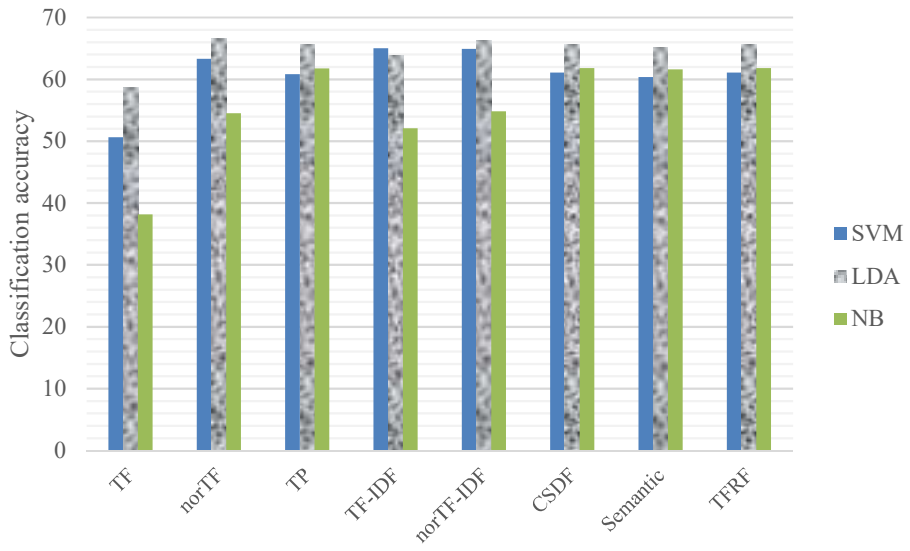


Figure 3.8 Classification accuracy with different types of features on 20-class 20newsgroup dataset

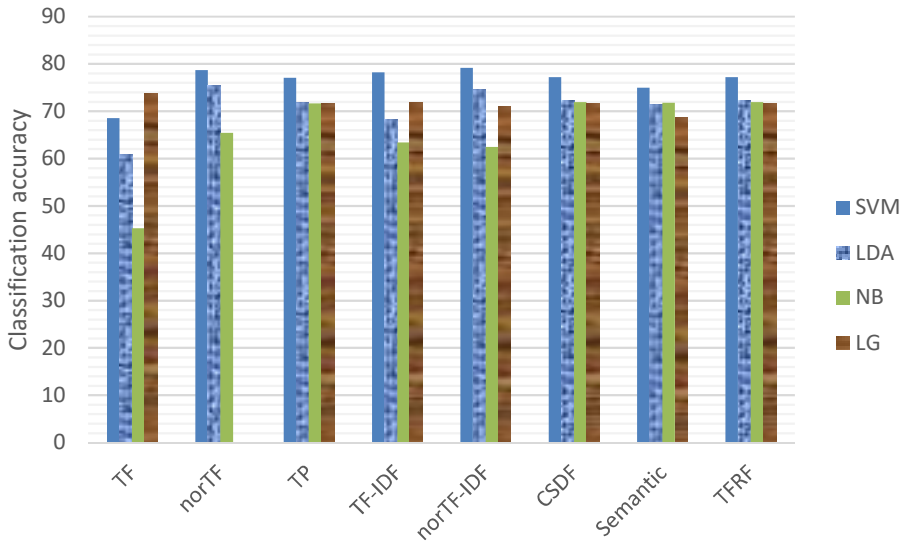


Figure 3.9 Classification accuracy with different types of features on 7-class 20newsgroup dataset

Tables 3.22 and 3.23 illustrate F_1 score of each class with different types of features on 7-class and 20-class 20newsgroup dataset using SVM classifier, respectively. The experimental results show that norTFIDF was the best representation for 7 classes; however, there was no significant difference in performance. On the other hand, TFIDF achieved the best F_1 score for 20 classes, which was significantly better than that of TF. In addition, CSDF and TFRF, which achieved the same score, were significantly better than TF as well.

Table 3.22 F_1 scores of SVM classifier on 20newsgroups (7 classes)

Feature / Class	TF	norTF	TP	TFIDF	norTFIDF	CSDF	Semantic	TFRF
alt	0.2560	0.5150	0.5160	0.5170	0.5550	0.5200	0.5340	0.5200
comp	0.7070	0.8460	0.8260	0.8270	0.8480	0.8310	0.7820	0.8310
misc	0.6700	0.7250	0.7320	0.7520	0.7370	0.7250	0.6880	0.7250
rec	0.7970	0.8680	0.8650	0.8650	0.8790	0.8650	0.8520	0.8650
Sci	0.6390	0.7440	0.7150	0.7350	0.7450	0.7180	0.6950	0.7180
Soc	0.4450	0.6640	0.6490	0.6420	0.6460	0.6480	0.6460	0.6480
Talk	0.6850	0.7620	0.7470	0.7610	0.7630	0.7450	0.7510	0.7450
Average	0.6730	0.7850	0.7700	0.7780	0.7890	0.7710	0.7510	0.7710
T-test	0.1202	0.9113	0.7769	0.8655	-	0.7800	0.5845	0.7800

Table 3.23 F_1 scores of SVM classifier on 20newsgroups (20 classes)

Feature / Class	TF	norTF	TP	TFIDF	norTFIDF	CSDF	Semantic	TFRF
Alt	0.4040	0.5110	0.4910	0.5140	0.5210	0.4890	0.4640	0.4890
Graphics	0.2870	0.5570	0.4780	0.5420	0.5610	0.5100	0.5120	0.5100
ms-windows	0.3760	0.5660	0.5410	0.5810	0.5680	0.5380	0.5310	0.5380
Ibm	0.4320	0.5370	0.4790	0.5570	0.5560	0.5030	0.4800	0.5030
Mac	0.4640	0.6120	0.5960	0.6570	0.6360	0.6000	0.5820	0.6000
window-x	0.3630	0.5670	0.5940	0.5720	0.5980	0.5770	0.5530	0.5770
Misc	0.6980	0.7170	0.7200	0.7410	0.6980	0.7220	0.7190	0.7220
Autos	0.5480	0.7460	0.7000	0.7360	0.7530	0.6950	0.6920	0.6950
Motorcycles	0.6400	0.7750	0.7670	0.8040	0.7980	0.7680	0.7530	0.7680
Baseball	0.7120	0.7600	0.7380	0.7740	0.7980	0.7360	0.7340	0.7360
Hockey	0.7580	0.8390	0.8200	0.8530	0.8610	0.8130	0.8150	0.8130
Crypt	0.6450	0.7640	0.7350	0.8030	0.7760	0.7390	0.7290	0.7390
Electronics	0.4190	0.4630	0.4410	0.4990	0.5120	0.4350	0.4260	0.4350
Med	0.4480	0.6280	0.5730	0.6250	0.6490	0.5850	0.5830	0.5850
Space	0.6260	0.7990	0.7170	0.7780	0.8090	0.7250	0.7180	0.7250
Christian	0.5160	0.6670	0.6750	0.6840	0.6510	0.6650	0.6710	0.6650
Guns	0.5750	0.6320	0.5990	0.6710	0.6480	0.6090	0.6210	0.6090
Mideast	0.6040	0.5920	0.5880	0.6140	0.6030	0.5790	0.5710	0.5790
Politics	0.4130	0.4680	0.4380	0.4820	0.4790	0.4490	0.4500	0.4490
Religion	0.2400	0.2990	0.3480	0.3590	0.3420	0.3440	0.3290	0.3440
Average	0.5160	0.6340	0.6100	0.6510	0.6500	0.6120	0.6050	0.6120
T-test (TFIDF)	**0.0041**	0.6829	0.3317	-	0.9723	0.3530	0.2747	0.3530
T-test (CSDF)	**0.0330**	0.6181	0.9579	0.3530	0.3735	-	0.8564	1

Tables 3.24, 3.25, 3.26 and 3.27 illustrate F_1 scores of each class with different types of features for 7 classes and 20 classes using LDA and Naïve Bayes classifiers, respectively. The experimental results show that semantic feature achieved the best average F_1 score for 7 classes, which was significantly better than TF with LDA classifier, whilst norTF was the best representation for 20 classes with LDA classifier. However, there was no significant difference in performance. With Naïve Bayes classifier, CSDF, semantic feature, and TFRF were the best representations, achieving F_1 scores significantly better than that of

TF for 7 classes. Furthermore, CSDF and TFRF achieved the best scores for 20 classes, which were significantly better than those of TF and TFIDF.

Table 3.24 F_1 scores of LDA classifier on 20newsgroups (7 classes)

Feature / Class	TF	norTF	TP	TFIDF	norTFIDF	CSDF	Semantic	TFRF
Alt	0.3641	0.4828	0.4711	0.4316	0.4702	0.4781	0.6275	0.4781
Comp	0.6959	0.8393	0.8064	0.7737	0.8297	0.8112	0.8890	0.8112
Misc	0.5291	0.6307	0.5997	0.5710	0.6369	0.6027	0.7513	0.6027
Rec	0.7616	0.8784	0.8587	0.8336	0.8776	0.8620	0.9000	0.8620
Sci	0.5711	0.7236	0.6813	0.6511	0.7213	0.6838	0.8393	0.6838
Soc	0.4273	0.5732	0.5311	0.4858	0.5522	0.5324	0.6122	0.5324
Talk	0.6254	0.7377	0.7033	0.6743	0.7274	0.7051	0.8121	0.7051
Average	0.5678	0.6951	0.6645	0.6316	0.6879	0.6679	0.7759	0.6679
T-test	**0.0116**	0.2696	0.1350	0.0655	0.2382	0.1456	-	0.1456

Table 3.25 F_1 scores of LDA classifier on 20newsgroups (20 classes)

Feature / Class	TF	norTF	TP	TFIDF	norTFIDF	CSDF	Semantic	TFRF
Alt	0.4226	0.5383	0.5627	0.5122	0.5314	0.5608	0.5654	0.5637
Graphics	0.4599	0.6372	0.5918	0.5810	0.6139	0.5903	0.5839	0.5921
ms-windows	0.5324	0.5776	0.5857	0.5744	0.5828	0.5758	0.5694	0.5751
Ibm	0.5219	0.5850	0.5591	0.5578	0.5814	0.5531	0.5091	0.5523
Mac	0.6222	0.6693	0.6810	0.6565	0.6658	0.6861	0.6458	0.6897
window-x	0.4979	0.5786	0.5760	0.5354	0.5746	0.5757	0.5656	0.5760
Misc	0.6813	0.7050	0.7172	0.7127	0.7097	0.7172	0.7173	0.7180
Autos	0.7117	0.7531	0.7522	0.7538	0.7448	0.7512	0.7488	0.7497
Motorcycles	0.7282	0.8034	0.8228	0.8265	0.8184	0.8228	0.8190	0.8228
Baseball	0.7509	0.7689	0.7917	0.7694	0.8058	0.7892	0.7901	0.7892
Hockey	0.7521	0.8090	0.8245	0.8054	0.8242	0.8225	0.8297	0.8245
Crypt	0.6787	0.8160	0.7965	0.7599	0.8113	0.7905	0.7925	0.7905
Electronics	0.4866	0.5236	0.5148	0.5065	0.5299	0.5237	0.5082	0.5229
Med	0.5364	0.6782	0.6713	0.6160	0.6604	0.6722	0.6639	0.6722
Space	0.6772	0.8148	0.7671	0.7503	0.8025	0.7694	0.7620	0.7684
Christian	0.5491	0.6457	0.6200	0.6083	0.6241	0.6193	0.6218	0.6193
Guns	0.6052	0.6619	0.6502	0.6363	0.6659	0.6494	0.6535	0.6494
Mideast	0.5950	0.6528	0.6336	0.6193	0.6240	0.6317	0.6332	0.6309
Politics	0.4203	0.4683	0.4392	0.4535	0.4675	0.4405	0.4484	0.4392
Religion	0.3050	0.3333	0.3750	0.3378	0.3403	0.3774	0.3737	0.3774
Average	0.5767	0.6510	0.6466	0.6287	0.6489	0.6459	0.6401	0.6462
T-test	0.0701	-	0.9135	0.5828	0.9598	0.8998	0.7878	0.9043

Table 3.26 F_1 scores of Naïve Bayes classifier on 20newsgroups (7 classes)

Feature / Class	TF	norTF	TP	TFIDF	norTFIDF	CSDF	Semantic	TFRF
Alt	0.3390	0.3240	0.5130	0.4220	0.3170	0.5130	0.5130	0.5130
Comp	0.5130	0.7780	0.7810	0.7360	0.7450	0.7860	0.7840	0.7860
Misc	0.2030	0.4730	0.7320	0.3390	0.4170	0.7300	0.7180	0.7300
Rec	0.6360	0.7770	0.7850	0.7600	0.7900	0.7860	0.7870	0.7860
Sci	0.4890	0.6130	0.6390	0.6360	0.5810	0.6430	0.6440	0.6430
Soc	0.4640	0.5580	0.6820	0.5200	0.5040	0.6820	0.6960	0.6820
Talk	0.4630	0.6660	0.6800	0.6280	0.6540	0.6830	0.6810	0.6830
Average	0.4990	0.6770	0.7160	0.6560	0.6560	0.7180	0.7180	0.7180
T-test	**0.0027**	0.2348	0.9755	0.1384	0.1495	1	-	1

Table 3.27 F_1 scores of Naïve Bayes classifier on 20newsgroups (20 classes)

Feature / Class	TF	norTF	TP	TFIDF	norTFIDF	CSDF	Semantic	TFRF
Alt	0.3110	0.4290	0.5110	0.4400	0.4160	0.5150	0.5020	0.5150
graphics	0.2150	0.4440	0.4740	0.4010	0.3560	0.4950	0.5000	0.4950
ms-windows	0.3140	0.3750	0.5050	0.3560	0.4110	0.5120	0.4880	0.5120
Ibm	0.3700	0.4230	0.5390	0.4480	0.4510	0.5300	0.5070	0.5300
Mac	0.3550	0.4660	0.6240	0.4940	0.4830	0.6100	0.5960	0.6100
window-x	0.2060	0.4840	0.5870	0.4210	0.4920	0.5740	0.5950	0.5740
Misc	0.3580	0.4920	0.6970	0.4930	0.5600	0.6940	0.7030	0.6940
Autos	0.4480	0.6420	0.7000	0.5590	0.6570	0.6990	0.6780	0.6990
motorcycles	0.3540	0.6470	0.7000	0.5390	0.6890	0.6940	0.6810	0.6940
baseball	0.5990	0.7210	0.7780	0.6330	0.7260	0.7770	0.7840	0.7770
Hockey	0.6360	0.7530	0.8630	0.8120	0.7610	0.8640	0.8650	0.8640
Crypt	0.4980	0.7520	0.7370	0.7300	0.7680	0.7390	0.7360	0.7390
electronics	0.3030	0.4100	0.4610	0.4130	0.4340	0.4650	0.4690	0.4650
Med	0.3810	0.6130	0.6100	0.5910	0.5890	0.6120	0.6090	0.6120
Space	0.5340	0.7120	0.6920	0.6800	0.7030	0.6920	0.6910	0.6920
christian	0.4340	0.6050	0.6780	0.5750	0.5650	0.6790	0.6910	0.6790
Guns	0.4060	0.6120	0.6560	0.5530	0.6340	0.6580	0.6600	0.6580
mideast	0.4690	0.5600	0.6290	0.5530	0.5760	0.6260	0.6240	0.6260
Politics	0.1960	0.3830	0.4580	0.3010	0.3550	0.4610	0.4670	0.4610
religion	0.1830	0.2320	0.3510	0.2400	0.2340	0.3520	0.3570	0.3520
Average	0.3850	0.5470	0.6200	0.5200	0.5520	0.6200	0.6180	0.6200
T-test	**0.0000**	0.0889	0.9980	**0.0219**	0.1204	-	0.9549	1

For finding an appropriate IG score threshold, this experiment compared the accuracy of different IG scores using LDA classifier with two-fold cross validation on training dataset for both 20 classes and 7 classes. Tables 3.28 and 3.29, and Figures 3.10 and 3.11 illustrate the classification accuracy of two-fold cross validation (CV) of different information gain scores for 20 classes and 7 classes, respectively. #attr. is the number of selected attributes or features. The experimental results show that the features which were selected using information gain score of 0.01 were the best. For 20 classes, semantic representation achieved the best classification accuracy followed by CSDF, norTFIDF, and norTF. For 7 classes, norTF achieved the best performance of classification followed by norTFIDF, CSDF, and the semantic method. However, there was no significant difference.

Table 3.28 The results of two-fold CV on 20newsgroups (20 classes)

IG score threshold	norTF		norTFIDF		CSDF		Semantic	
	accuracy	#attr.	accuracy	#attr.	accuracy	#attr.	Accuracy	#attr.
0.15	0	0	0	0	45.77	191	39.24	83
0.1	45.75	15	33.13	15	56.30	315	53.58	180
0.05	32.47	42	47.16	42	66.22	580	64.12	443
0.01	**72.67**	**738**	**72.95**	**750**	**73.94**	**1075**	**74.42**	**1036**
0.005	73.34	849	73.68	828	74.14	1096	74.23	1064

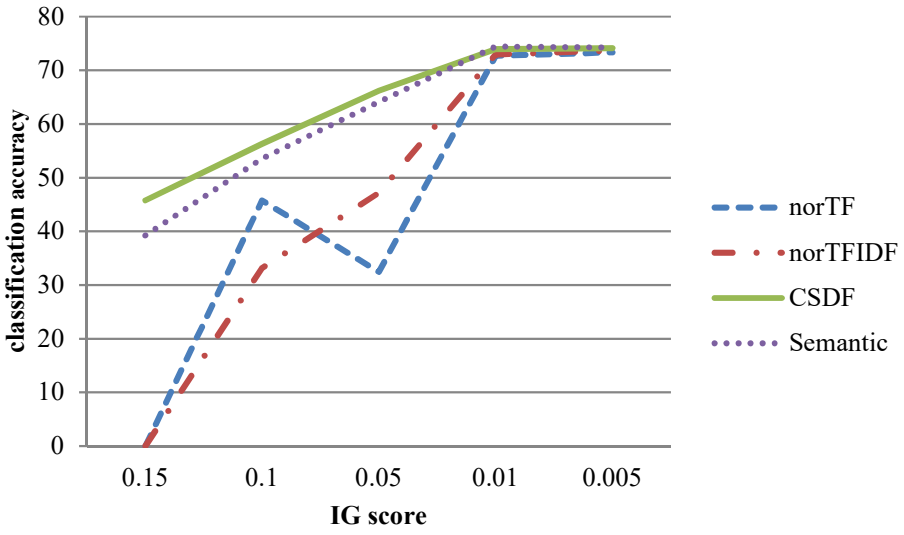


Figure 3.10 The classification accuracy of two-fold CV on 20newsgroups

(20 classes)

Table 3.29 The results of two-fold CV on 20newsgroups (7 classes)

IG score threshold	norTF		norTF-IDF		CSDF		Semantic	
	Accuracy	#attr.	accuracy	#attr.	accuracy	#attr.	Accuracy	#attr.
0.15	0	0	0	0	49.87	82	46.90	47
0.1	40.3	5	39.6	5	56.20	164	54.55	89
0.05	58.16	23	58.07	23	68.04	414	67.40	277
0.01	**76.26**	**425**	**75.02**	**420**	**73.70**	**994**	**72.64**	**956**
0.005	77.23	848	75.75	844	73.96	1087	73.29	1062

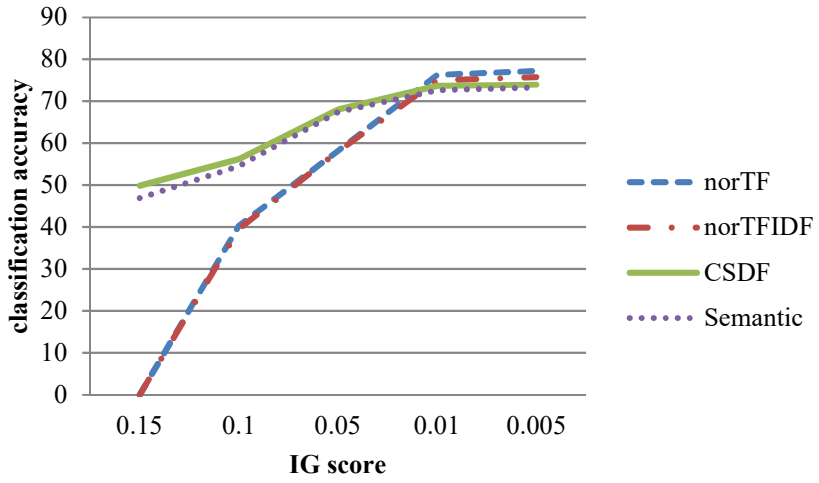


Figure 3.11 The classification accuracy of two-fold CV on 20newsgroups

(7 classes)

Using IG score of 0.01 as threshold to select features, Tables 3.30, 3.31, 3.32, and 3.33 show the training and testing accuracies for 20 classes and 7 classes using different types of features for document representation, respectively. Furthermore, Figures 3.12 and 3.13 illustrate the comparison of two accuracies for 20 classes and 7 classes, respectively. The experimental results show that CSDF representation achieved the best performance for 20 classes in terms of training accuracy and testing accuracy. For 7 classes, CSDF representation achieved over 82% training accuracy, but its testing accuracy dramatically decreased to lower than 72%. There is an overfitting problem most likely, which should be investigated further in future research.

Table 3.30 Training accuracy with 20 classes

IG score	norTF		norTF-IDF		CSDF		Semantic	
	accuracy	#attr.	accuracy	#attr.	Accuracy	#attr.	Accuracy	#attr.
0.1	33.42	15	32.87	15	62.52	315	56.54	180
0.05	47.05	42	47.14	42	74.04	580	70.47	443
0.01	**79.24**	**738**	**79.83**	**750**	**83.83**	**1075**	**83.26**	**1036**

Table 3.31 Testing accuracy with 20 classes

IG score	norTF		norTF-IDF		CSDF		Semantic	
	Accuracy	#attr.	Accuracy	#attr.	Accuracy	#attr.	Accuracy	#attr.
0.1	31.43	15	30.70	15	52.04	315	48.37	180
0.05	43.03	42	42.98	42	60.45	580	58.17	443
0.01	**67.23**	**738**	**67.37**	**750**	**67.70**	**1075**	**67.14**	**1036**

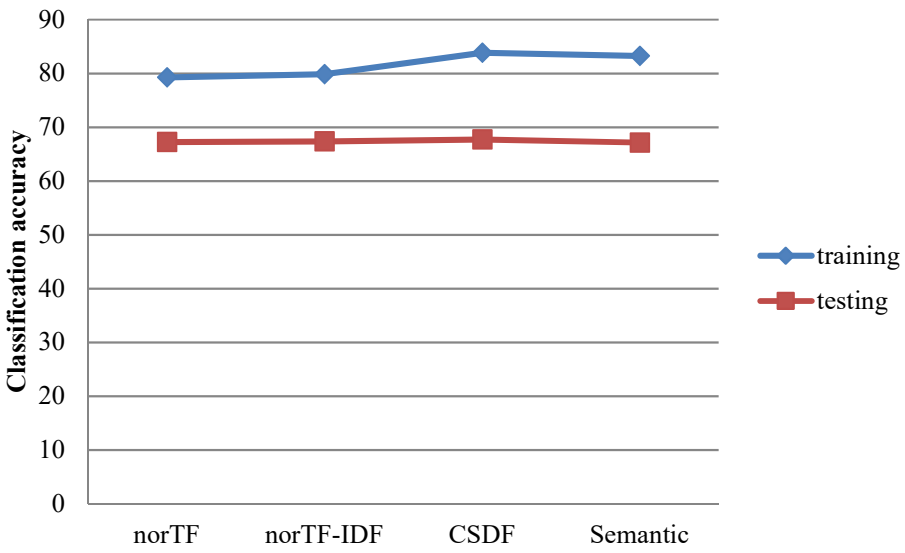


Figure 3.12 The training and testing accuracy with 20 classes using IG score 0.01

Table 3.32 Training accuracy with 7 classes

IG score	norTF		norTF-IDF		CSDF		Semantic	
	accuracy	#attr.	accuracy	#attr.	Accuracy	#attr.	Accuracy	#attr.
0.1	40.46	5	39.70	5	59.28	164	55.88	89
0.05	59.18	23	58.81	23	73.15	414	70.79	277
0.01	**80.04**	**425**	**78.91**	**420**	**82.53**	**994**	**82.61**	**956**

Table 3.33 Testing accuracy with 7 classes

IG score	norTF		norTF-IDF		CSDF		Semantic	
	accuracy	#attr.	Accuracy	#attr.	Accuracy	#attr.	Accuracy	#attr.
0.1	40.67	5	39.87	5	56.09	164	51.78	89
0.05	59.00	23	58.31	23	66.49	414	63.69	277
0.01	**74.03**	**425**	**73.04**	**420**	**71.75**	**994**	**70.71**	**956**

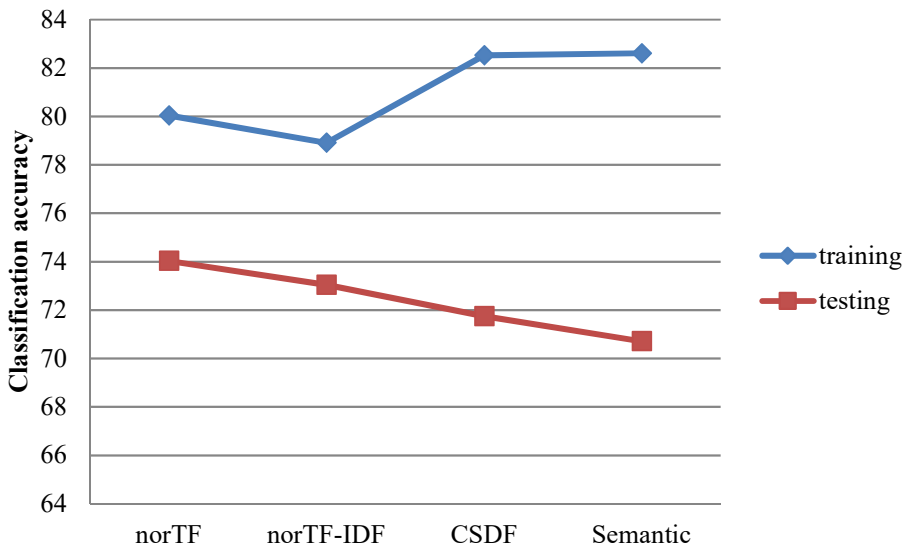


Figure 3.13 The training and testing accuracy with 7 classes using IG score 0.01

Furthermore, Table 3.34 and Figure 3.14 show the classification performances of the combined method TF-CSDF on 20newsgroups dataset with $\propto = 0$ to $\propto = 1$. For 20 classes, the starting point (only CSDF) achieved 66.52% classification accuracy while the ending point (only normalised TF) achieved 65.61% accuracy. The highest peak is at $\propto = 0.05$ (i.e., 0.05norTF+0.95CSDF), which reached 66.91% accuracy. At this point, the classification accuracy increased by 0.39% and 1.3% compared to the original CSDF and the normalised TF, respectively. For 7 classes, the starting point (only CSDF) achieved 75.4% classification accuracy while the ending point (only normalised TF) achieved 72.27% accuracy. The highest peak is at $\propto = 0.1$ (i.e., 0.1norTF+0.9CSDF), which reached 75.64% accuracy. At this point, the classification accuracy increased by 0.24% and 3.37% compared to the original CSDF and the normalised TF, respectively. To sum up, TF-CSDF, which is the combination between normalised TF and CSDF features, achieved higher classification accuracy than the individuals in this dataset.

Table 3.34 Experimental results of TF-CSDF

alpha	The classification accuracy of LDA classifier	
	20 classes	7 classes
	TF-CSDF	TF-CSDF
0	0.6561	0.7227
0.05	**0.6691**	0.7554
0.1	0.6681	**0.7564**
0.15	0.6677	0.7557
0.2	0.6672	0.7549
0.25	0.6660	0.7546
0.3	0.6657	0.7545
0.35	0.6656	0.7541
0.4	0.6657	0.7539
0.45	0.6657	0.7540
0.5	0.6656	0.7540
0.55	0.6657	0.7541
0.6	0.6654	0.7542
0.65	0.6657	0.7541
0.7	0.6654	0.7540
0.75	0.6656	0.7539
0.8	0.6656	0.7537
0.85	0.6654	0.7539
0.9	0.6654	0.7539
0.95	0.6654	0.7539
1	0.6652	0.7540

20 classes **7 classes**

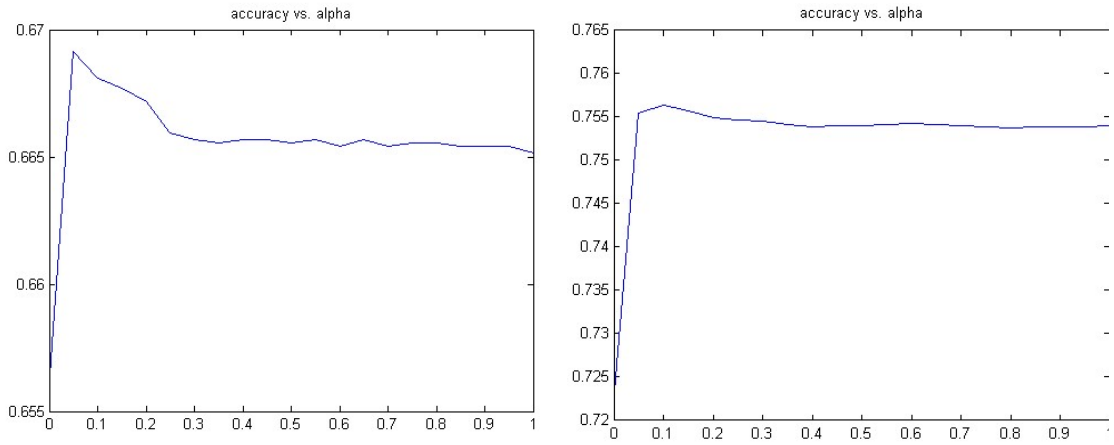


Figure 3.14 The classification accuracy of TF-CSDF

3.7.2.3 Web returned document dataset

For the web returned document dataset, both filter approach and wrapper approach were adopted for feature selection. Using the top 40 document frequency scores of each class, 258 features were initially selected in total by the filter approach. With the wrapper approach, the features were further selected using sequential forward floating search (SFFS) method with LDA classifier [71].

The classification accuracy on web returned documents is illustrated in Tables 3.35, 3.36, and 3.37. The experimental results show that TF, norTF, TF-IDF, norTF-IDF, and CSDF achieved almost similar performance; however, CSDF, TFRF, and semantic feature had a tendency of using fewer features when wrapper approach was adopted. CSDF and TFRF had the same number of features. On the other hand, semantic representation achieved the lowest performance. There may be two reasons for this issue. Firstly, the representative words might not be appropriate to present the classes. Secondly, there are a lot of proper nouns which are unknown to the NLKT path_similarity function.

Table 3.38 and Figure 3.15 illustrate the classification accuracy of using different numbers of features and different classifiers on web returned documents. The experimental results using filter-based feature selection show that TFIDF achieved the best classification accuracy with LDA and Naïve Bayes classifiers, whilst norTFIDF achieved the best accuracy with logistic regression classifier. Furthermore, TFRF achieved the best performance with all three classifiers using wrapper-based feature selection. This means that TFRF did not face too much overfitting problem.

Table 3.35 Experiment results on web returned documents (I)

Accuracy	TF		norTF		TF-IDF	
	Acc.	#attr.	Acc.	#attr.	Acc.	#attr.
Training (filter)	94.59	258	95.56	258	94.77	258
Training (wrapper)	89.44	88	88.47	87	88.09	89
2-fold cross validation (filter)	92.01	258	92.24	258	93.13	258
2-fold cross validation (wrapper)	87.99	88	87.69	87	86.41	89
Testing (filter)	92.66	258	92.94	258	93.05	258
Testing (wrapper)	88.52	88	87.84	87	88.24	89

Table 3.36 Experiment results on web returned documents (II)

Accuracy	norTF-IDF		CSDF		Semantic	
	Acc.	#attr.	Acc.	#attr.	Acc.	#attr.
Training (filter)	95.33	258	94.25	258	84.55	258
Training (wrapper)	89.36	87	88.09	85	75.59	79
2-fold cross validation (filter)	93.13	258	91.49	258	81.55	258
2-fold cross validation (wrapper)	87.83	87	86.48	85	73.04	79
Testing (filter)	92.94	258	92.38	258	80.73	258
Testing (wrapper)	87.90	87	87.28	85	72.55	79

Table 3.37 Experiment results on web returned documents (III)

Accuracy	TFRF	
	Acc.	#attr.
Training (filter)	94.44	258
Training (wrapper)	90.82	85
2-fold cross validation (filter)	89.93	258
2-fold cross validation (wrapper)	89.93	85
Testing (filter)	92.32	258
Testing (wrapper)	90.20	85

Table 3.38 The classification accuracy on web returned documents

Classifier / Feature	Filter			Wrapper		
	LDA	NB	LG	LDA	NB	LG
TF	92.66	88.12	88.07	88.52	87.28	87.34
norTF	92.94	85.77	89.64	87.84	85.60	87.11
TF-IDF	**93.05**	**88.57**	89.19	88.24	85.88	87.34
norTF-IDF	92.94	86.55	**90.25**	87.9	84.37	87.00
CSDF	92.38	87.17	89.47	87.28	84.31	87.00
Semantic	80.73	75.41	78.04	72.55	71.76	72.38
TFRF	92.32	87.17	89.58	**90.20**	**87.56**	**90.03**

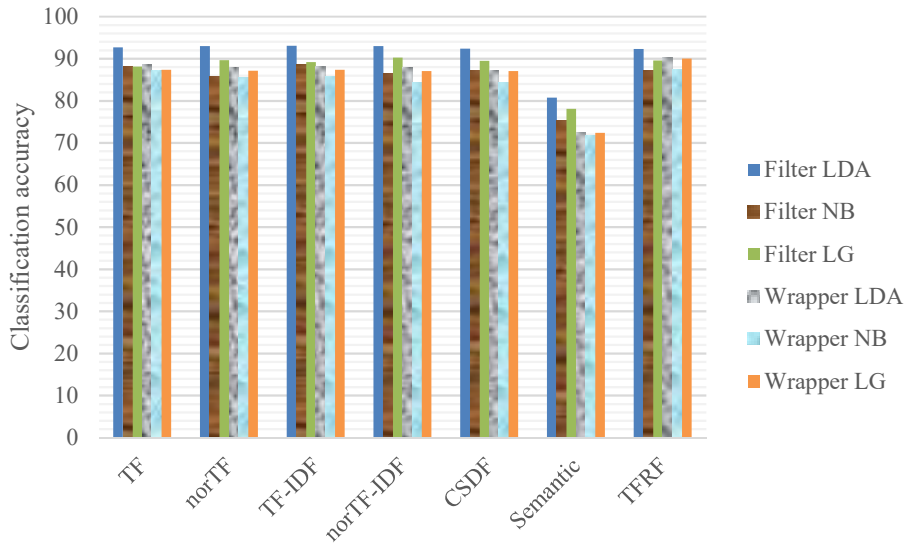


Figure 3.15 The classification accuracy on web returned documents

Tables 3.39 and 3.40 illustrate F_1 scores of each class with different types of features using LDA classifier. The experimental results show that TFIDF achieved the best F_1 score which was significantly better than that of semantic representation using filter-based feature selection. Furthermore, CSDF and TFRF achieved almost the same average score, which was significantly better than that of semantic feature. With the wrapper method, TFRF and CSDF was significantly better than semantic representation.

Tables 3.41 and 3.42 illustrate F_1 scores of each class with different types of features using Naïve Bayes classifier. The experimental results show that almost all the representations achieved similar scores except for semantic feature which achieved significantly lower score using both filter-based and wrapper-based feature selection. In addition, Tables 3.43 and 3.44 illustrate F_1 scores of each class with different types of features using logistic regression classifier. The experimental results follow the same trend as those of using Naïve Bayes classifier.

Regarding machine learning based methods, the performance of CSDF was better than or equal to that of TFRF on Reuters and 20newsgroup datasets. On web returned document dataset, they had the same performance when using filter-based feature selection; however, TFRF's performance was better than CSDF's performance when wrapper approach was adopted. Furthermore, these experimental results can confirm the conclusion of Lan et al. [47] that not all supervised term weighting methods are better than unsupervised methods. For example, the classification accuracies of TFIDF and norTFIDF were the best for both 20 classes and 7 classes when using SVM classifier. To sum up, CSDF and TFRF had almost the same performance in general; however, the overfitting problem was the main issue for CSDF representation.

Table 3.39 F_1 scores of LDA classifier on web returned document

Feature / Class	TF	norTF	TFIDF	norTFIDF	CSDF	Semantic	TFRF
Animal	0.8977	0.8956	0.9061	0.8894	0.9143	0.8691	0.9143
Art	0.9575	0.9600	0.9618	0.9556	0.9575	0.7855	0.9575
Flower	0.9289	0.9292	0.9296	0.9267	0.9257	0.9043	0.9257
Food	0.9865	0.9821	0.9888	0.9843	0.9800	0.9037	0.9800
Movie	0.8489	0.8548	0.8485	0.8577	0.8904	0.8201	0.8899
Shopping	0.9103	0.9312	0.9248	0.9269	0.9287	0.6088	0.9266
Sport	0.9144	0.9203	0.9186	0.9231	0.8400	0.8731	0.8383
Travel	0.9735	0.9669	0.9735	0.9756	0.9626	0.7613	0.9626
Average	0.9272	0.9300	0.9315	0.9299	0.9249	0.8157	0.9244
T-test (TFIDF)	0.8513	0.9469	-	0.9440	0.7725	**0.0133**	0.7558
T-test (CSDF)	0.9192	0.8158	0.7725	0.8217	-	**0.0178**	0.9813

Table 3.40 F_1 scores of LDA classifier on web returned document (wrapper)

Feature / Class	TF	norTF	TFIDF	norTFIDF	CSDF	Semantic	TFRF
Animal	0.9238	0.9017	0.9264	0.9242	0.6863	0.7817	0.9108
Art	0.9638	0.9550	0.6987	0.9618	0.9330	0.7283	0.9660
Flower	0.8878	0.8905	0.8932	0.9183	0.9135	0.8632	0.9201
Food	0.9865	0.9797	0.9655	0.9749	0.9821	0.5414	0.9704
Movie	0.8842	0.8956	0.8979	0.8544	0.8802	0.7737	0.7340
Shopping	0.9009	0.9284	0.9184	0.9167	0.9104	0.6682	0.9217
Sport	0.8972	0.8529	0.9005	0.8557	0.8159	0.8629	0.8894
Travel	0.6967	0.6892	0.9343	0.6905	0.9343	0.7368	0.9497
Average	0.8926	0.8866	0.8919	0.8871	0.8820	0.7445	0.9078
T-test (TFRF)	0.7163	0.6169	0.6920	0.6272	0.5512	**0.0036**	-
T-test (CSDF)	0.8162	0.9196	0.8236	0.9128	-	**0.0151**	0.5512

Table 3.41 F_1 scores of Naïve Bayes classifier on web returned document

Feature / Class	TF	norTF	TFIDF	norTFIDF	CSDF	Semantic	TFRF
Animal	0.8830	0.8520	0.8770	0.8740	0.8860	0.8140	0.8860
Art	0.9060	0.8620	0.9190	0.9050	0.9150	0.7430	0.9150
Flower	0.8090	0.8520	0.8140	0.8320	0.8080	0.8300	0.8080
Food	0.9660	0.8120	0.9690	0.8490	0.9360	0.8650	0.9360
Movie	0.8750	0.8600	0.8920	0.8770	0.8620	0.7520	0.8620
Shopping	0.8490	0.8560	0.8420	0.8340	0.8390	0.5830	0.8390
Sport	0.8600	0.8690	0.8580	0.8460	0.8330	0.7900	0.8330
Travel	0.9050	0.9040	0.9170	0.9050	0.8940	0.7190	0.8940
Average	0.8820	0.8580	0.8860	0.8650	0.8720	0.7620	0.8720
T-test (TFIDF)	0.8576	0.1868	-	0.3275	0.5472	**0.0049**	0.5472
T-test (CSDF)	0.6646	0.4736	0.5472	0.7385	-	**0.0094**	1

Table 3.42 F_1 scores of Naïve Bayes classifier on web returned document (wrapper)

Feature / Class	TF	norTF	TFIDF	norTFIDF	CSDF	Semantic	TFRF
Animal	0.8940	0.8740	0.8930	0.8690	0.6880	0.7940	0.8600
Art	0.9610	0.9320	0.6850	0.9570	0.9140	0.7290	0.9610
Flower	0.8570	0.8710	0.8700	0.8560	0.8750	0.8360	0.8520
Food	0.9820	0.9640	0.9560	0.9350	0.9630	0.5360	0.9540
Movie	0.8800	0.8760	0.8900	0.8390	0.8630	0.7720	0.7460
Shopping	0.8880	0.8840	0.8960	0.8660	0.8420	0.6830	0.8760
Sport	0.8900	0.8410	0.8480	0.8220	0.7640	0.8350	0.8610
Travel	0.6910	0.6710	0.9260	0.6710	0.8980	0.7170	0.9260
Average	0.8800	0.8640	0.8700	0.8520	0.8510	0.7380	0.8790
T-test (TF)	-	0.7152	0.8190	0.5227	0.5112	**0.0085**	0.9826
T-test (CSDF)	0.5112	0.7660	0.6503	0.9820	-	**0.0293**	0.4811

Table 3.43 F_1 scores of logistic regression classifier on web returned document

Feature / Class	TF	norTF	TFIDF	norTFIDF	CSDF	Semantic	TFRF
Animal	0.8350	0.8470	0.8460	0.8570	0.8550	0.8270	0.8610
Art	0.9230	0.9400	0.9400	0.9410	0.9530	0.7540	0.9520
Flower	0.8760	0.8790	0.8700	0.8940	0.8870	0.8760	0.8860
Food	0.9530	0.9620	0.9620	0.9640	0.9510	0.8580	0.9560
Movie	0.8320	0.8440	0.8340	0.8490	0.8340	0.8050	0.8340
Shopping	0.8720	0.9060	0.8920	0.9100	0.8850	0.5940	0.8790
Sport	0.8640	0.8810	0.8710	0.8860	0.8820	0.8460	0.8870
Travel	0.8930	0.9160	0.9270	0.9220	0.9130	0.7380	0.9140
Average	0.8810	0.8970	0.8920	0.9030	0.8950	0.7870	0.8960
T-test (norTFIDF)	0.2985	0.7734	0.6446	-	0.7062	**0.0090**	0.7471
T-test (CSDF)	0.5140	0.9304	0.9203	0.7062	-	**0.0133**	0.9584

Table 3.44 F_1 scores of logistic regression classifier on web returned document (wrapper)

Feature / Class	TF	norTF	TFIDF	norTFIDF	CSDF	Semantic	TFRF
Animal	0.9120	0.9040	0.9190	0.9170	0.6970	0.7760	0.9090
Art	0.9590	0.9520	0.6980	0.9590	0.9330	0.7220	0.9610
Flower	0.8700	0.8890	0.8880	0.9000	0.9130	0.8660	0.9140
Food	0.9800	0.9750	0.9590	0.9630	0.9750	0.5380	0.9630
Movie	0.8670	0.8740	0.8940	0.8430	0.8730	0.7720	0.7530
Shopping	0.8920	0.9130	0.8980	0.9140	0.9050	0.6780	0.9160
Sport	0.8820	0.8470	0.8870	0.8390	0.8050	0.8460	0.8740
Travel	0.6870	0.6860	0.9210	0.6870	0.9260	0.7440	0.9550
Average	0.8810	0.8800	0.8830	0.8780	0.8780	0.7430	0.9050
T-test (TFRF)	0.5475	0.5291	0.5502	0.4980	0.5039	**0.0029**	-
T-test (CSDF)	0.9513	0.9712	0.9135	0.9890	-	**0.0138**	0.5039

3.8 Summary

Document representation is critical in improving document classification performance. TF-IDF is still widely used by many search engines for information retrieval due to its simplicity, interpretability and effectiveness. Although there have been alternative term weighting schemes proposed in recent years, including machine learning based methods [169] [170], none of them have been as widely recognised and adopted as TF-IDF by search engines and document databases. This research proposes a simple but effective method for document representation based on term frequency and class specific document frequency under the VSM framework. The proposed features for document representation, CSDF and TF-CSDF, are based on the assumption that class specific document frequency contains very important information for class discrimination. The experimental results show that the CSDF based document representation was equal to or better than other widely used features for VSM representation, including TF-IDF in

terms of classification accuracy on three datasets. CSDF also needed a smaller number of features when selected by the wrapper approach. Compared to machine learning based methods such as TFRF, the performance of CSDF was better than or equal to that of TFRF in general; however, the experimental results show that overfitting is a major issue for CSDF method. In addition, the combination between term frequency and CSDF in appropriate proportion can achieve higher classification accuracy than the individual methods. The experimental results also show that not all supervised term weighting methods are better than unsupervised methods which are the same as in [47].

For semantic information, the experimental results show that the performance of semantic features is equal to or lower than that of the other tested methods on the three datasets. Semantic representation in this experiment is not as effective as expected for two reasons. Firstly, the representative words for each class may not be appropriately chosen. Secondly, there are a lot of proper nouns which are unknown to the NLKT path_similarity function.

As for classifier fusion, the classification accuracy of decision fusion was slightly better than the best accuracy achieved by individual classifiers. This means that there is not much new information added by the different types of features and classifiers.

TF-CSDF has similar simplicity and interpretability as TF-IDF, and it is more effective than TF-IDF for document representation and classification. Furthermore, our method is simpler than the class-indexing-based method [45]. It does not require large memory to store data, and the computation cost is very low. We expect that CSDF as a new term weighting technique would be widely used in search engines and document databases for document representation or indexing.

Chapter 4 GCrank: A new ranking method using document classification scores

4.1 Introduction

Apart from classification problems, one of the greatest challenges faced by search engines is web document ranking. In search engines, it often occurs that the top-ranked returned web documents may not contain information relevant to users' search intentions, and relevant fresh web pages may not get high ranks [98]. Baeza-Yates et al. [10] have shown that almost 80% of the users who use search engines are interested in only the top 3 returned results. Pan [171] has also found that high click-through rates appear only in the top ranked web pages. Therefore, it is very important to develop effective web document ranking algorithms. To automatically 99rganize documents into user's interesting topic groups is a solution to the ranking problem. In this chapter, classification and ranking of search engine returned web documents are two issues that will be addressed.

Document classification techniques have been applied to many areas, including IR. In this chapter, the LDA classifier is used to classify search returned documents into related topics and re-rank the documents using classification scores. The class specific document frequency (CSDF) weighting method for document representation presented in Chapter 3 is adopted. It has been demonstrated to be able to effectively improve the performance of document classification in comparison with other widely used vector space model (VSM) based document representations [158]. A new ranking method called GCrank is proposed in this chapter. It combines the original Google ranking scores and the LDA classification scores of the Google search returned documents to improve

ranking performance. The experimental results demonstrated that GCrank method improve the performance of ranking in terms of several widely used ranking performance criteria. It is expected that the top-ranked web documents would be most relevant to the user's search intent.

4.2 Document ranking criteria

4.2.1 Google ranking

Google search engine [1] uses a complex hyperlink-content-based ranking approach. It is PageRank method that makes use of the link structure of the web. For each webpage, a quality ranking is calculated by forming a probability distribution over webpages. The content-based features of webpages are used as well. The ranking systems are categorised into two types. Firstly, for a single word query, the hit lists of the query word in each webpage, such as title, anchor, URL, and large font for that word are considered. Each of these has its score. A final rank score is given by the combination of hit list scores and a PageRank score. Secondly, multiple hit lists are generated for a multi-word query. A proximity score is calculated based on how far apart the hits are in a webpage, and then combined with the scores for individual single word queries. Google applies several techniques to improve search quality, including PageRank, anchor text, and proximity information. Google ranking represents the state-of-the-art. This research tries to further improve Google ranking by using classification scores to re-rank Google returned web documents. Therefore, to evaluate the proposed method in this thesis, Google ranking is used as a baseline ranking method for comparison.

4.2.2 The proposed ranking method: GCrank

The objective of the proposed method aims to re-rank Google returned documents using classification scores to improve the performance of ranking. In this paper, LDA is adopted for classification due to its simplicity and resilience to overfitting. The LDA classification score *Cscore* is defined in this paper as follows:

$$\Sigma = (n_1\Sigma_1 + n_2\Sigma_2)/n \tag{4.1}$$

$$\mathbf{w}=\Sigma^{-1}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2) \tag{4.2}$$

$$w_0 = \mathbf{w}^T(n_1\boldsymbol{\mu}_1 + n_2\boldsymbol{\mu}_2)/n \tag{4.3}$$

$$Cscore = \mathbf{w}^T\mathbf{x} - w_0 \tag{4.4}$$

where Σ_1 and Σ_2 are the covariance matrices of the samples of class 1 and class 2 respectively, n_1, n_2 are the number of samples in class 1 and class 2 respectively, n is the total number of all the samples, $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$ are the means of the samples of class 1 and class 2 respectively.

From the idea that top-ranked webpages should belong to the same topic category as the one relevant to the query, the proposed method is emerged. That is, involving classifiers in the ranking process may improve webpage ranking performance. How much the webpages are relevant to the query usually is indicated by classification scores of web documents. In this paper, a reason for using LDA classifier is because one can visualize its operation as splitting a high-dimensional feature space with a hyperplane defined by *Cscore*=0. All points indicating web documents on one side of the hyperplane are classified into one class. The corresponding web document will have a high classification score, if a point is far away from the hyperplane. It is ensured that this web document is in that class with high confidence. Thus, this method believes that a web document with a high classification score should have a relatively high rank in the search

returned results. However, a returned web document's rank should be considerably reduced, if it is classified into a query irrelevant topic category. Google ranking has already been known as a superior ranking method. If Google ranking can be further improved by combining it with web document classification scores, it would be highly desirable. This paper proposes the GCrank method as described by equations (4.5), (4.6), and (4.7).

$$norGscore_j = \frac{1}{GoogleRank_j}, 0 \leq norGscore_j \leq 1 \tag{4.5}$$

$$norCscore_j = \begin{cases} \frac{Cscore_j}{\text{Max}Cscore}, & 0 \leq norCscore_j \leq 1 \\ 0, & \text{if document j is not in the same topic category as the query} \end{cases} \tag{4.6}$$

$$GCrank_j = \alpha \times norGscore_j + (1-\alpha) \times norCscore_j \tag{4.7}$$

where $GCrank_j$ is a new combined score of document j, $norGscore_j$ is a normalisation of Google ranking score of document j which is between 0 and 1. $GoogleRank_j$ is a Google's rank of document j, $norCscore_j$ is a normalisation of the classification score of document j which is between 0 and 1, $Cscore_j$ is a classification score of document j (belonging to the same topic category as the query), $MaxCscore$ is the maximum classification score of all documents in the query topic category, and α is a weighting factor. In our experiment, α has been investigated and its optimal value for the data used was found by evaluation using data from one category only. The details will be described in Section 4.3.2.

4.3 Experiments and results

4.3.1 Experimental procedure

There are two experiments in this chapter. The first experiment aims to evaluate the quality of the classifier and to define the classification score. The classifier in this experiment is LDA [73] [74] [174] [175] [176] [177] because LDA does not face too much of the overfitting problem. The sample web documents were created from analysing the top 56 Google search returned documents per query from Google API which was previously tested in Chapter 3. Each document was pre-processed as follows. First of all, we consider only the title and snippet content in each document. All HTML tags were removed. After that, all content was separated into tokens. Only nouns were considered, since the most selective terms should be nouns [10] [168]. Finally, stemming was used to derive words to their stem. In addition, the CSDF method was used for document representation. For evaluation purposes, 80 test queries were selected from eight popular search topics or categories, as shown in Table 3.4. Each topic consists of 10 queries containing one to three words that are generally known and easy for user evaluation. Approximately 4,500 returned web documents were used for these test queries in the experiment. These documents were randomly separated into 2,679 documents for training data and 1,785 documents for testing data. The details of this dataset are shown in Table 3.5. Feature selection was conducted using both filter and wrapper approaches [13] [68] [69] [71]. For the filter approach, the terms were selected from those that have the top 40 document frequency values in each class of documents. In this stage, only 258 features were selected from 320 terms with high document frequency values for the eight classes, with duplicate terms removed. The features selected by filtering were selected again using

sequential forward floating search (SFFS) method with LDA classifier as the wrapper [71].

The second experiment aims to evaluate and compare the ranking performance of the returned web documents directly from Google API and those re-ranked by the GCrank method to evaluate the effectiveness of the proposed ranking method. To evaluate ranking performance in the experiment, three widely used performance criteria were adopted: mean average precision (MAP), normalised discounted cumulated gain (nDCG), and precision at 20 (P@20). P@20 was used to measure the relevance of the top 20 returned web documents. MAP and nDCG can measure not only the relevance but also the ranking of relevance of the returned web documents. More details about these criteria are described in section 2.7. Integrating the evaluation results in terms of these three criteria may lead to more comprehensive evaluation. As ground truth, whether a web document is relevant or irrelevant and how important it is will be decided from user feedback. In the experiment, whether a returned web document is highly relevant, mildly relevant, or irrelevant for each test query was decided by three participants. Questionnaires were used to obtain users' evaluative feedback. The participants selected top 10 highly relevant web documents and then rank these documents in order.

4.3.2 Selection of a weighting factor (α)

To find a proper value of the weighting factor α, the ranking performance of web returned documents with different α values were evaluated by the three performance criteria. The 10 queries of the movie category were tested by GCrank method. The feedbacks from three participants (P1, P2, and P3) from the University of Essex gave the true ranks of web returned documents. Only the top

20 returned web documents were decided in three evaluation methods: MAP, nDCG, and P@20. Highly relevant documents were the top 10 documents, whose scores were multiplied by two in the nDCG evaluation method, while mildly relevant documents were the documents in the 11th to 20th rank, whose scores were kept unchanged.

Table 4.1 and Figure 4.1 show the evaluation results based on the true ranks given by the three participants with four different weighting factor values. The best ranking performance was a weighting factor of 0.9. Therefore, the following evaluations used α equal to 0.9 to evaluate the proposed method.

Table 4.1 Experimental results of using different weighting factor values

Method / A	nDCG			MAP			P@20			Avg
	P1	P2	P3	P1	P2	P3	P1	P2	P3	
0.80	0.8932	0.8482	0.5899	1.0000	0.8961	0.4467	0.7900	0.7400	0.4900	0.7438
0.85	0.9107	0.8589	0.6131	0.7814	0.7170	0.3825	0.8250	0.7650	0.5050	0.7065
0.90	1.0000	0.9464	0.6547	0.9224	0.8293	0.4293	1.0000	0.9350	0.5800	**0.8108**
0.95	0.9722	0.9222	0.6472	0.7290	0.6544	0.3529	0.9250	0.8650	0.5600	0.7364
Avg	0.9440	0.8939	0.6262	0.8582	0.7742	0.4029	0.8850	0.8263	0.5338	0.7494
	0.8214			0.6784			0.7483			

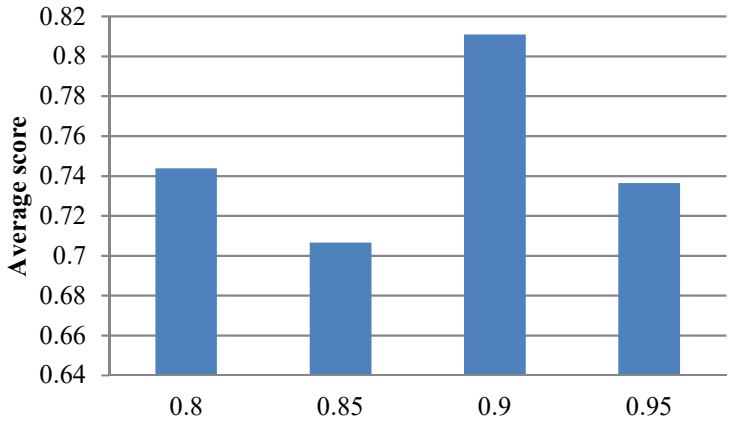


Figure 4.1 Average results with different weighting factors

4.3.3 Experimental results and evaluation

4.3.3.1 Evaluation of the performance of classification and classification scores

Table 4.2 shows the classification performances of the LDA classifier using the CSDF features for document representation as described in Chapter 3. The experimental results show that the performance of classification was very good. It achieved over 80% accuracy on both training set and testing set. In addition, there was not much of an overfitting problem. At this stage, the classification scores of all documents were defined and saved to use for web document ranking later. Table 4.3 and Figure 4.2 illustrate the classification performance of each category. The experimental results show that "animal" and "flower" categories achieved the two lowest classification accuracy, which was lower than 90%, while "food" category had the highest accuracy.

Table 4.2 Classification accuracy of LDA classifier

Accuracy	The top 40 (258 features)	Wrapper (85 features)
Training	94.25%	88.09%
Testing	92.38%	87.28%

Table 4.3 Classification accuracy of the LDA classifier for each category

Category	Accuracy (%)
Animal	**88.19**
Arts	95.54
Flower	**89.43**
Food	**99.64**
Movie	90.71
Shopping	90.70
Sport	95.52
Travel	98.38
Average	93.51

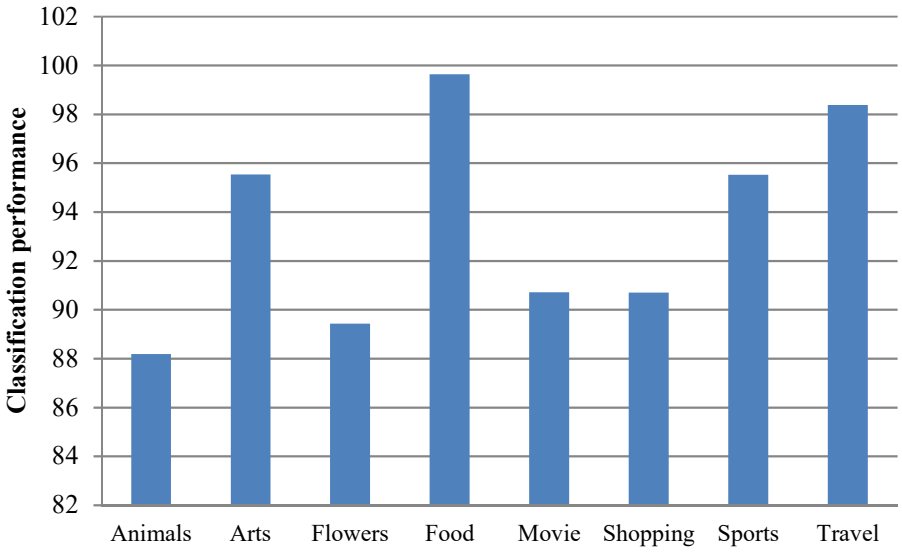


Figure 4.2 Classification accuracy of the LDA classifier for each category

4.3.3.2 Evaluation of the effectiveness of GCrank method

The GCrank method was used to re-rank Google returned web documents, aiming to improve the relevance ranking of the top 56 Google search returned documents per query in comparison with that of the original rank from Google API. The performances of ranking the web documents returned from the 80 test queries in eight categories were evaluated by three performance criteria. The true relevance of the returned web documents was decided by three participants from The University of Essex (P1, P2, and P3). Wilcoxon rank-sum test [178] [179] is a nonparametric statistical significance test method that does not require the assumption of normal distribution. It was adopted for statistical test of the integration of three evaluation methods which were decided by three participants with the p value ≤ 0.05 as a significant level in this experiment. The experimental results of the original Google method and the GCrank method are shown in Tables 4.4 and 4.5, respectively.

Table 4.4 Evaluation results of the original Google ranking

	nDCG			MAP			P@20			Avg
Category	P1	P2	P3	P1	P2	P3	P1	P2	P3	
Animal	0.9873	0.8924	0.7106	0.9668	0.8779	0.5981	0.9750	0.8950	0.6700	**0.8415**
Art	0.9629	0.9424	0.8773	0.9583	0.9583	0.7708	0.9750	0.9750	0.8400	**0.9178**
Flower	0.7456	0.6928	0.4559	0.6484	0.6048	0.3171	0.7550	0.5950	0.5350	**0.5944**
Food	0.9753	0.9763	0.8948	0.9625	0.9248	0.7919	0.9700	0.9650	0.8500	**0.9234**
Movie	0.9231	0.9424	0.6586	0.8617	0.9208	0.7417	0.9250	0.9700	0.5650	**0.8343**
Shopping	0.9814	0.9323	0.9203	0.9208	0.8699	0.8599	0.9650	0.9000	0.9100	**0.9177**
Sport	0.9684	0.9847	0.9399	0.9514	0.9749	0.8864	0.9800	0.9900	0.9300	**0.9562**
Travel	0.9858	0.9539	0.8559	0.9769	0.9249	0.8483	0.9900	0.9500	0.7800	**0.9184**
Avg	0.9412	0.9147	0.7892	0.9059	0.8820	0.7268	0.9419	0.9050	0.7600	**0.8630**
	0.8817			**0.8382**			**0.8690**			

Table 4.5 Evaluation results of the GCrank method

	nDCG			MAP			P@20			Avg
Category	P1	P2	P3	P1	P2	P3	P1	P2	P3	
Animal	0.9809	0.9048	0.7253	0.9437	0.9006	0.6220	0.9500	0.9050	0.6850	**0.8464**
Art	1.0000	1.0000	0.8861	1.0000	1.0000	0.7902	1.0000	1.0000	0.8500	**0.9474**
Flower	0.7865	0.7031	0.4714	0.7090	0.6176	0.3419	0.7900	0.6350	0.5700	**0.6249**
Food	0.9850	0.9840	0.9040	0.9917	0.9724	0.8128	0.9750	0.9850	0.8650	**0.9416**
Movie	0.9453	1.0000	0.6547	0.8987	1.0000	0.7248	0.9350	1.0000	0.5800	**0.8598**
Shopping	1.0000	0.9425	0.9164	0.9895	0.9175	0.8861	0.9750	0.9250	0.9300	**0.9424**
Sport	0.9964	1.0000	0.9530	0.9900	1.0000	0.9170	0.9900	1.0000	0.9500	**0.9774**
Travel	1.0000	0.9808	0.8679	1.0000	0.9568	0.8327	1.0000	0.9650	0.8150	**0.9354**
Avg	0.9618	0.9394	0.7973	0.9403	0.9206	0.7409	0.9519	0.9269	0.7806	**0.8844**
	0.8995			**0.8673**			**0.8865**			

The average evaluation results of the GCrank method based on the true ranks of each participant were better than those of the original Google ranking by about 2%. The best improvement was obtained in terms of the MAP evaluation criterion. The best improvement by the GCrank method was in the “art”, “flower”, and “shopping” categories. On the other hand, there was no obvious improvement in the “animal” category. Thus, the ranking of the returned web documents using the GCrank method was better than that of the original Google ranking based on the

true ranks from the three participants in terms of three evaluation methods.

To see whether the ranking performance improvement by the GCrank method is statistically significant, the integration (average) of the MAP, nDCG, and P@20 evaluation results of the two ranking methods were compared using the Wilcoxon rank-sum test. Table 4.6 shows the statistical test results, with the *p* value for eight categories being 0.0427 which is less than 0.05. The “animal” and “flower” categories had the lowest classification accuracy as shown in Tables 4.4 and 4.5. If these categories are disregarded, the statistical difference between the two ranking methods should be larger. Table 4.6 illustrates that the *p* value for seven categories after the “animal” category had been removed was 0.0243. After the “animal” and “flower” categories had been removed, the *p* value for six categories was 0.0059. This demonstrates that the GCrank method was significantly better than the original Google ranking. If only the categories with higher classification accuracy were considered, it was more significantly better.

Table 4.6 Statistical significance test results: GCrank vs. Google ranking

Number of categories	Statistical test results (*p* value)
8 categories	0.0427
7 categories (no animal)	0.0243
6 categories (no animal and flower)	0.0059

This experiment has indicated that the proposed method is significantly helpful for ranking the returned web documents in general. Table 4.7 shows four examples of re-ranking using the GCrank method that upgrades the ranking of some documents that are classified into the category relevant to the query with high scores and downgrades the ranking of some documents that are not classified into the category relevant to the query. In Tables 4.7 and 4.8, a new rank, a class number, a classification score, an original rank, and the content of a web

document are shown in each web document, respectively. Classes 1, 4, 5, 7 and 8 are for animal, food, movie, sport, and travel, respectively. The first example shows the query “avatar” which was a famous movie. A highlighted web returned document was classified into “sports” category by the LDA classifier. The GCrank method ranked this web document down to the bottom of the list from a Google rank of 49. As a matter of fact, this web document was not directly related to “avatar” movie based on the ranks of the three participants. The second example illustrates some results returned by the query “frozen”, which was a famous movie as well. Due to a highlighted web document was classified to the movie category with a high score, the GCrank method upgraded a highlighted web document related to the reviews of this movie from 55^{th} to 41^{st} position. In the third example, the query was “taj mahal” from “travel” category. A highlighted web returned document was not directly related to “taj mahal” due to it was truly a restaurant name, and the LDA classifier classified it into “food” category. Therefore, this document was downgraded to 53^{rd} from 44^{th} position using GCrank method. Finally, the query “great wall” was a famous attraction in China. Due to a highlighted web document had a high classification score, the proposed method raised the rank of about the history of this place from the bottom of the list up to 23^{rd} position. These four examples demonstrated that the GCrank method can create interpretable results.

Table 4.7 Examples of re-ranking using GCrank (I)

Query	Avatar
Original ranking	46 5 38.767 46 Avatar Secrets: An Interactive Documentary for th 47 5 42.261 47 Avatar & Aliens are the same movie - The O 48 5 39.332 48 Avatar | Film | The Guardian Working on live-actio 49 7 0 49 Requests are closed This is a simple blog dedicated to A 50 5 42.261 50 Avatar & Aliens are the same movie - The O 51 5 38.234 51 FaceYourManga: Home Download and Print your avatar 52 5 43.293 52 Avatar Movie Review (2009) | Plugged In Plugged 53 5 38.767 53 AvatarHD - Android Apps on Google Play Trong thế giới 54 5 38.655 54 Xbox Avatars – Windows Apps on Microsoft Store Use the X 55 5 38.767 55 VUDU - Avatar From Academy Award(R) winning 56 5 43.51 56 Avatar Fortress Fight 2 | 1000 Free Flash Games
Re-ranking using GCrank	46 5 39.332 48 Avatar | Film | The Guardian Working on live-a 47 5 38.726 45 Avatar Vectors, Photos and PSD files | Free Dc 48 5 38.767 46 Avatar Secrets: An Interactive Documentary fc 49 5 37.737 44 Avatar ... Woman's Invisible Jet) – Hugo 50 5 36.953 41 2045 Initiative The Dalai Lama Supports 2045's <b 51 5 38.767 53 AvatarHD - Android Apps on Google Play Trong thế giới < 52 5 38.234 51 FaceYourManga: Home Download and Print your ava 53 5 38.655 54 Xbox Avatars – Windows Apps on Microsoft Store Use th 54 5 38.767 55 VUDU - Avatar From Academy Award(R) winnir 55 5 5.5026 42 What is avatar? A Webopedia Definition (1) A 56 7 0 49 Requests are closed This is a simple blog dedicated to <b
Query	Frozen
Original ranking	46 5 4.5183 46 Frozen Games Frozen Games, Play t 47 5 3.4454 47 Frozen Food and Power Outages: When to S 48 5 10.085 48 Frozen (Widescreen) - Walmart.com Bring h 49 5 3.4454 49 Frozen Food and Power Outages: When to S 50 5 19.951 50 Huffy 16" Girls' Disney Frozen Bil 51 5 10.085 51 Frozen (Widescreen) - Walmart.com Bring h 52 5 4.5183 52 Save 50% on Frozen Cortex on Steam A hard 53 5 19.309 53 Frozen Toys, Costumes, Gifts & Mercha 54 5 23.728 54 Shop for Disney Frozen & Licensed Cha 55 5 9.2977 55 Frozen | Film | The Guardian Film blog Let i 56 5 19.143 56 Frozen / Disney - TV Tropes Frozen
Re-ranking using GCrank	41 5 9.2977 55 Frozen | Film | The Guardian Film bl 42 5 3.6486 27 'Frozen' director crushes a 43 5 3.661 28 'Frozen' director on Tarzan 44 7 0 21 Frozen | zulily Frozen has be 45 5 4.5183 38 Frozen Synapse: A Simultaneous Tur 46 5 3.661 34 'Frozen' director on Tarzan 47 5 3.6486 35 'Frozen' Director Finally Cl 48 5 3.6486 40 frozen - Wiktionary frozen 49 5 4.5183 46 Frozen Games Frozen Games 50 5 4.5183 52 Save 50% on Frozen Cortex on Steam 51 5 3.4454 47 Frozen Food and Power Outages: Wh

Table 4.8 Examples of re-ranking using GCrank (II)

Query	Taj mahal				
Original ranking	44	4	0	44	Taj Mahal Indian Restaurant The
	45	8	60.782	45	India Agra Taj Mahal - YouTube A
	46	8	60.538	46	Taj Mahal NewTaj.EOL
	47	8	106.1	47	Taj Mahal needs nine-year mud pa
	48	8	62.437	48	taj mahal: Latest News, Videos an
	49	8	60.538	49	Rockport Music ℥€" Taj Mahal Aug
	50	8	59.86	50	Taj Mahal Gardens Found to Align
	51	8	61.849	51	Taj Mahal - A Tribute to Beauty -
	52	8	67.372	52	Tourist reportedly dies at Taj Mahal
	53	7	0	53	Tajmahal: The True Story - The H
	54	8	59.121	54	Tourist falls to death while posing for selfie at
Re-ranking using GCrank	44	8	62.437	48	taj mahal: Latest News, Videos
	45	8	60.782	45	India Agra Taj Mahal - YouTub
	46	8	58.715	42	Taj Mahal | Biography, Albums
	47	8	60.538	46	Taj Mahal NewTaj.EOL
	48	8	61.849	51	Taj Mahal - A Tribute to Beaut
	49	8	60.538	49	Rockport Music ℥€" Taj Mahal
	50	8	59.86	50	Taj Mahal Gardens Found to A
	51	8	59.121	54	Tourist falls to death while posing for selfi
	52	1	0	27	Tajmahal AR ... Sat - 5 pm to
	53	4	0	44	Taj Mahal Indian Restaurant T
	54	7	0	53	Tajmahal: The True Story - Th
Query	Great wall				
Original ranking	46	8	55.034	46	Great Wall Szechuan House - Chinese Re
	47	7	0	47	GW Supermarket ... Store Locator; |; Employmer
	48	8	41.448	48	The Great Wall: From Beginning to End:
	49	8	55.181	49	Great Wall at Mutianyu (Beijing, China):
	50	8	59.358	50	Great Wall Chinese Restaurant - Order C
	51	8	55.063	51	GREAT WALL CHINESE RESTAURANT-FAR
	52	8	54.285	52	BrainPOP | Social Studies | Learn about Grea
	53	8	50.992	53	Great Wall - Order Online - Prince Georg
	54	8	48.724	54	Great Wall - Order Online - Palm Coast -
	55	8	40.494	55	Great Wall The first emperor of the Qin
	56	8	73.987	56	Ancient China for Kids: The Great Wall -
Re-ranking using GCrank	23	8	73.987	56	Ancient China for Kids: The Great Wall
	24	8	54.949	25	Great Wall Marathon Annual maratho
	25	8	50.992	22	Great Wall - Order Online - Danville -
	26	8	58.163	32	Hiking China's Great Wall @ Nati
	27	8	47.22	24	GREAT WALL - greatwall37.com GR
	28	8	53.532	34	The Great Wall, China - Lonely Planet
	29	8	57.581	43	Great Wall Chinese Restaurant - Orde
	30	8	54.949	37	Great Wall of China - Enchanted Learn
	31	8	59.358	50	Great Wall Chinese Restaurant - Orde
	32	8	57.177	44	Great Wall Hiking Tours: Hike WILD <b
	33	8	54.661	39	Off the Great Wall - YouTube Get you

4.4 Summary

Ranking web returned documents is one of the most important tasks of a search engine, since almost 80% of the users who use search engines are only interested in the top 3 results. This chapter proposes GCrank, an effective web document ranking method using LDA classification scores to re-rank Google search returned

documents. These documents that have low classification scores or whose classes are not in the same category as the one related to the query are downgraded by GCrank method. On the other hand, this method increases the ranks of web documents that have high classification scores. The experimental results report that the ranking of the returned web documents by the GCrank method was significantly better than the original Google ranking in terms of the ranking performance criteria, as indicated in Table 4.6. There is also proof that the GCrank method can rank web documents more specific to user's information need. Thus, our hypothesis about the LDA hyperplane has been successfully evaluated by the experiment, which states that if a point representing a web document is far away from the LDA hyperplane, this document should have a relatively high rank among the search returned documents.

It should be noted that this chapter focuses on improving the original Google ranking only, without comparing with other ranking methods. Subjective bias in the performance evaluation is another main concern. For instant, the performance evaluation often depends on the queries used in the experiment and the decisions on the relevance of web documents with the original queries. Multiple evaluation criteria from different perspectives were adopted to ensure a trustworthy comparison and evaluation. However, further work should be investigated to overcome the limitations in this aspect of performance. It is noteworthy that with a limited number of topic categories and limited size of web documents tested in the experiment, this thesis presents preliminary but promising results of re-ranking Google search returned documents using classification scores. Deeper investigation and more extensive testing would be required in future research.

Chapter 5 A hybrid method for term ranking and its applications in automatic query suggestion

5.1 Introduction

Normally, search engine users submit only a couple of words as a query. To understand more precisely users' information need is one of the greatest challenges faced by search engines. Most existing search engines retrieve information by finding exact keywords. Users sometimes do not know the certain vocabulary of the searched topic, and they do not know how to produce appropriate queries because they do not know how search algorithms work [4]. One solution to these issues is to devise a query suggestion section in search engines, which helps users in their searching activities. Kelly et al. [12] have pointed out that when users run out of ideas or are faced with a cold-start problem, query suggestions are necessary. Kato et al. [108] have investigated three types of logs in Bing (the Microsoft's search engine). They have found that query suggestions are usually used when the original query is a single-term query, or a uncommon query, or after the user has clicked on several URLs in the first search result page. Furthermore, Carpineto and Romano [180] have reported that an advantage of query suggestion will increase a chance to return a related document that does not consist of the original query terms. Niu and Kelly [181] have reported that users save significantly more documents retrieved by query suggestions than by user-generated queries, when searching for difficult topics.

There are many query suggestion methods that extract features or query relevant terms from implicit feedback such as log files, ontologies, and documents returned from search engines. After that, these features are used to generate query suggestions. It is very hard to use log files for generating query suggestions due to

privacy protection. Therefore, web returned documents were chosen as the sources of query suggestion terms in this thesis.

A new query suggestion method combining two ranked retrieval methods: TF-IDF and Jaccard coefficient is proposed. In addition, this chapter applies the ranking idea of the GCrank method described in Chapter 4 to ranking the generated query suggestions as well. The experiment was conducted for comprehensive performance evaluation of the proposal method using multiple criteria emphasizing different perspectives.

5.2 Query suggestion methods

Document-based features are used as a source to generate and rank query suggestions in this thesis. He and Ounis [11] have presented a measure value which estimates how the existence of a query term spreads over different subsets of returned documents. The higher value is, the more the returned document is linked to the topic. Their results show that the entropy measure for relevant documents ranked in the top 5 is very high, while it decreases rapidly when the ranking becomes lower. Web returned documents form pseudo relevance feedback, assuming that the top ranked documents are relevant to the query and can be used as sources for generating query suggestions [2]. Various ranking methods have been used for ranking query suggestions. Ranked retrieval model is the traditional ranking model based on the VSM framework. Typical ranked retrieval methods include term frequency, Jaccard coefficient, and TF-IDF [2] [53].

5.2.1 TF-IDF

TF-IDF is the most famous term weighting technique in IR [10]. The TF-IDF score of a term in a set of documents used in this chapter is calculated by equations (5.1) and (5.2):

$$TFIDF_i = \sum_{j=1}^{N} w_{i,j} \tag{5.1}$$

$$w_{i,j} = \begin{cases} (1 + \log_2 TF_{i,j}) \times \log_2 \frac{N}{DF_i}, & \text{if } TF_{i,j} > 0 \\ 0, & \text{otherwise} \end{cases} \tag{5.2}$$

where $TF_{i,j}$ is the frequency of term i in document j, DF_i is the number of documents in which term i appears, N is the total number of documents.

TF-IDF has been used for measuring word relatedness [37]. Therefore, it can be applied to identify terms in the web documents returned from Google search as query suggestions, which are mostly relevant to the original query.

5.2.2 Jaccard coefficient

Jaccard coefficient [53] measures the overlap of two returned documents D_1 and D_2, which are represented as vectors of terms. They may not have the same size. The Jaccard coefficient for a length-normalised model is calculated by equation (5.3):

$$Jaccard(D_1, D_2) = \frac{|D_1 \cap D_2|}{|D_1 \cup D_2|} \tag{5.3}$$

where $\cap$ represents intersection and $\cup$ represents union. In this research, D_1 and D_2 are bags of words which contain a query suggestion candidate. They are selected from words which appear in at least two returned documents. Furthermore, the notion of ‘multiset’ or ‘bag’ in mathematics is a generalisation

of the notion of set, in which members can appear more than once. In general, a multiset is a result of the intersection or union of multisets [182].

If a query suggestion candidate is from more than two returned documents, its Jaccard coefficient can be extended as equation (5.4).

$$Jaccard(D_1, D_2, \dots D_M) = \frac{|D_1 \cap D_2 \cap \dots D_M|}{|D_1 \cup D_2 \cup \dots D_M|} \quad (5.4)$$

In this research, *M* documents that contain this suggestion term are identified by each query suggestion candidate. Jaccard coefficient is calculated as the score for ranking this candidate.

Jaccard coefficient has been applied for measuring the similarity between search texts, and computed semantic relatedness between two concept clouds [126] [183].

5.2.3 Cosine similarity

Cosine similarity [53] is one of the most commonly used methods to rank returned documents. In this chapter, cosine similarity is used to measure the similarity between a query suggestion candidate and the original query. For length-normalised vectors, cosine similarity is simply a dot product, i.e.,

$$\cos(\vec{q}, \vec{s}) = \vec{q}.\vec{s} = \sum_{i=1}^{B} q_i s_i \quad (5.5)$$

where q_i is the term frequency of the original query in returned document *i*, s_i the term frequency of a query suggestion candidate in returned document *i*, and *B* is the number of documents in which both the original query and the query suggestion candidate appear.

5.2.4 A new method based on the combination of TF-IDF and Jaccard coefficient

By adaptation and combination of the TF-IDF, Jaccard coefficient, and cosine similarity methods, six query suggestion methods as shown in Table 5.1 were investigated. These different methods were used to extract features as query suggestion and to rank them in different ways. The performance of query suggestion methods generated in this experiment will be evaluated using multiple performance criteria.

In the proposed combinations of methods, the query suggestions were selected from the top 10 TF-IDF scores or Jaccard coefficient scores, depending on which scores are more important or reflect better relevance. After that, these suggestions may be re-ranked in descending order by cosine similarity scores.

Table 5.1 Query suggestion methods to be investigated

No.	QS methods	Feature extraction and ranking (selection)	Suggestion re-ranking
1	Tfidf	TF-IDF score	-
2	Tfcos	TF-IDF score	Cosine similarity score
3	Jac	Jaccard coefficient score	-
4	Jacos	Jaccard coefficient score	Cosine similarity score
5	Tfjac	TF-IDF score and Jaccard coefficient score	-
6	Tfjacos	TF-IDF score and Jaccard coefficient score	Cosine similarity score

Our previous experiment reported that the TF-IDF method was capable of generating relevant suggestions of the user's original query, whilst Jaccard coefficient was the best method in ranking query suggestions. Therefore, the Tfjac method proposed in this chapter selects terms from the combination of the top 10 candidate terms from the TF-IDF method and the Jaccard coefficient method [184] [185]. The algorithm starts from finding duplicate words from both

methods. If the amount of these words is less than 10, more candidate terms from the Jaccard coefficient method are added. If the number of terms is still lower than 10, more candidate terms from the TF-IDF method are added till 10 query suggestions are selected. Figure 5.1 illustrates the overall process of the Tfjac method.

For the Tfjacos method, the selection process is the same as the Tfjac method; however, the generated query suggestions are re-ranked in descending order by their cosine similarity scores.

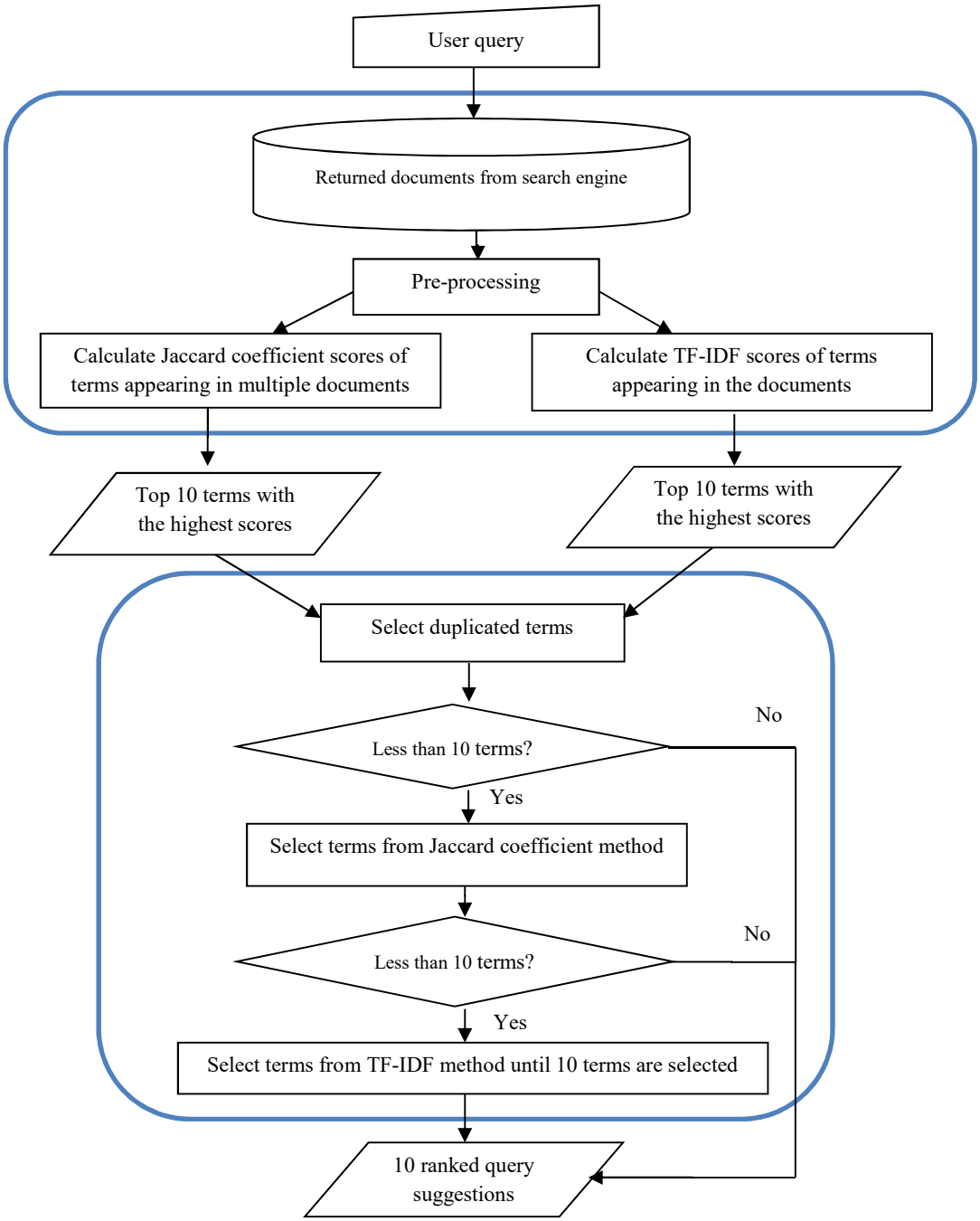


Figure 5.1 Diagram of the Tfjac method

5.2.5 Using LDA classification scores for ranking query suggestions

The basic idea is to use the GCrank method to find the ranking scores of the web documents where query suggestion terms appear and use these scores to rank the query suggestions. For self-containedness, the equations of the GCrank method are repeated as follows:

$$norGscore_j = \frac{1}{GoogleRank_j}\ ,\ 0 \le norGscore_j \le 1 \tag{5.6}$$

$$norCscore_j = \begin{cases} \frac{Cscore_j}{\text{Max}Cscore}\ ,\ 0 \le norCscore_j \le 1 \\ 0, \quad \text{if document j is not in the same topic category as the query} \end{cases} \tag{5.7}$$

$$GCrank_j = \begin{cases} \alpha \times norGscore_j + (1-\alpha) \times norCscore_j \\ 0, \quad \text{if document j is not in the same topic category as the query} \end{cases} \tag{5.8}$$

where $GCrank_j$ is a combined ranking score of document j, $norGscore_j$ is the normalised Google ranking score of document j, $GoogleRank_j$ is the original Google's rank of document j, $norCscore_j$ is the normalised classification score of document j, $Cscore_j$ is the original classification score of document j, $MaxCscore$ is the maximum classification score of all the web documents returned by a query, and α is a weighting factor. To generate and rank query suggestions, this chapter proposes the following sGCrank and mGCrank methods, which are based on the above mentioned document ranking method. The first method is based on the assumption that if a term appears in many documents belonging to the same class as the original query, this term will be a good suggestion. For query suggestion i, its ranking score is defined by equation (5.9):

$$sGCrank_i = \sum_{j=1}^{n} GCrank_{ij} \quad (5.9)$$

where $sGCrank_i$ is a *GCrank* score of query suggestion *i*, $GCrank_{ij}$ is a *GCrank* score of document *j* in which query suggestion *i* appears. n is the number of documents in which query suggestion *i* appears.

The mGCrank method assumes that a term will be a good suggestion if it appears frequently in documents belonging to the same class as the original query. The mGCrank method is described by equation (5.10):

$$mGCrank_i = \sum_{j=1}^{n} (GCrank_{ij} \times TF_{ij}) \quad (5.10)$$

where $mGCrank_i$ is a *GCrank* score of query suggestion *i*, $GCrank_{ij}$ is a *GCrank* score of document *j* in which that query suggestion *i* appears, n is the number of documents in which query suggestion *i* appears. TF_{ij} is a term frequency of query suggestion *i* in document *j*. These two proposed methods were compared with query suggestion methods described in Section 5.2.4.

5.3 Evaluation methods

In the experiment, four widely used performance criteria: MRR, MAP, nDCG, and P@k were adopted to evaluate ranking performance. The experiments in this chapter focus on precision scores at the top 5 or top 10 query suggestions: P@5 or P@10. MRR is used for measuring the performance of ranking, whilst P@k is used for measuring the performance of generating relevant query suggestions. MAP and nDCG can measure the performance of both ranking and producing relevant query suggestions. nDCG can distinguish highly relevant suggestions from mildly relevant suggestions. Whether a query suggestion is relevant or irrelevant will be decided by user feedback. The integrated evaluation results from the four methods may lead to a more comprehensive evaluation.

User evaluation will be conducted as well to check whether the evaluation using the above criteria is acceptable by real users. Questionnaires are used to obtain users' evaluative feedback. The participants select a top query suggestion respectively from the query suggestions made by each query suggestion method for each of the 80 test queries, and then rank these top query suggestions in order.

5.4 Experiments and results

5.4.1 Experimental design

There are three major experiments in this chapter. The first experiment is called Tfjac experiment. Five state-of-the-arts and the proposed query suggestion methods listed in Table 5.1 are compared. The second experiment, which is called GCrank experiment, compares some methods in the Tfjac experiment with the proposed methods using classification scores for ranking query suggestions: sGCrank and mGCrank. Finally, the third experiment is called the QS experiment. This experiment evaluates and compares the relevance of the returned documents in interactive web search. To evaluate the effectiveness of the proposed query suggestion methods, these documents are retrieved from the original query with and without using query suggestion respectively.

Based on the findings of He and Ounis [11], it has been decided that query suggestions are generated from analysing the top 8 Google search returned documents in the Tfjac experiment and QS experiment. That would be sufficient to generate highly relevant suggestions with respect to the original query. On the other hand, query suggestions are generated from the top 56 Google returned documents in the GCrank experiment due to classification purpose. The web documents used in this chapter are the same as those used in Chapter 3 and Chapter 4. The six query suggestion methods and two classification-score-based

methods described in the previous section have been investigated in these experiments. Table 3.4 illustrates 80 test queries which were selected from eight popular search categories for evaluation purposes. Each category consists of 10 queries containing of one to three words which are commonly known and easy for user evaluation.

5.4.2 User's selection of suggested queries and assessment of relevance of search results

User evaluation has been implemented as well to confirms whether the evaluation results are satisfactory to real users. It is important to know whether a query suggestion is truly good in the performance evaluation, questionnaires were given to users for obtaining users' evaluative feedback.

For the Tfjac experiment, for each test query, highly relevant, mildly relevant and irrelevant suggestions were decided by two approaches. Firstly, 50% of the decisions were based on the suggestions by Google search engine, which has been widely known. Secondly, another 50% of the decisions were made by 5 participants who were PhD students studying in different fields at University of Essex. This aims to reduce subjective bias and make the expected results more reliable. Only the top 10 query suggestions were considered in the evaluation methods. Five highly relevant suggestions and five mildly relevant suggestions or irrelevant suggestions were chosen by the participants. They were asked to select 3 best suggestions from the lists of query suggestions made by each query suggestion method for each of the 80 test queries, and then rank these 3 chosen suggestions for each test query using a scale from 1 to 3, with 3 indicating the most important suggestion. A total score for each of these suggestion terms was obtained by adding up the scores given to the term by all the participants. The

term with the highest total score was regarded as the most important suggestion. The top 5 terms with the highest total scores were regarded as the highly relevant suggestions in this experiment. In the nDCG evaluation method, these highly relevant suggestion scores were multiplied by two.

For the GCrank experiment, top 5 suggestions from the 10 query suggestions made by each query suggestion method for each of the 80 test queries were chosen and ranked by 3 participants who were PhD students studying in different fields at University of Essex. These top 5 suggestions for each test query were ranked using a scale from 1 to 5, with 5 indicating the most important suggestion. The 5 highly relevant suggestions for each test query were chosen based on the participants' scores in the same way as in the Tfjac experiment. In the nDCG method, the highly relevant suggestion scores were multiplied by two to six depending on how important that suggestion was, as decided by the users. In this case, six is for the most important suggestion.

Finally, for evaluating the relevance of the search results in the QS experiment, questionnaires were also used. There are 16 returned websites for each test query in each questionnaire, half of which were returned by using the original query and the other half returned by using query suggestion. Eight participants at University of Essex were asked to select and rank the top 3 most relevant returned webpages for each of the 80 test queries using a scale from 1 to 3, with 3 indicating the most relevant webpage. A total score for each returned webpage was obtained by adding up the scores given to the webpage by all the participants. Based on the total scores of the returned webpages, highly relevant, mildly relevant and irrelevant webpages were determined. In the nDCG evaluation method, the highly relevant webpage scores were multiplied by two.

For mildly relevant suggestions/webpages or irrelevant suggestions/webpages, the scores were kept unchanged or set to zero, respectively.

5.4.3 Experimental results and evaluation

Comprehensive comparative experiments have been conducted to demonstrate the effectiveness of the methods developed in this chapter. The experimental results are shown in the following tables and figures, where an asterisk indicates that the related score differs significantly from the best one with the p value ≤ 0.05. The methods for statistical significance test in the Tfjac experiment and QS experiment are t-test, and in the GCrank experiment is the Wilcoxon rank-sum test. Conroy [186] has suggested that a tested dataset which has equal medians between two groups should avoid the Wilcoxon rank-sum test. According to this, the performance data in both Tfjac and QS experiments were not suitable for the Wilcoxon rank-sum test, therefore t-test was used instead in these experiments.

5.4.3.1 The Tfjac experimental results

The experimental results are given in Table 5.2 and Figure 5.2. The results of evaluation using MRR report that the best query suggestion methods were Tfjac and Jacos, followed by Tfjacos. The ranking score of Tfidf was significantly lower than those of the best methods. Similarly, the results of evaluation using MAP show that Tfjac was the best method to generate query suggestions in terms of producing and ranking related words. The results of evaluation using nDCG also show that Tfjac was the best method to rank and produce highly relevant suggestions followed by Jacos and Tfjacos. However, its score was not significantly different from the others. Finally, the results of evaluation using P@10 show that Tfjac, Jac, and Jacos had the same score and outperform other

methods in terms of generating relevant suggestions. On the other hand, the scores of Tfidf and Tfcos methods are significantly lower than those of the best methods.

Table 5.2 Experimental results of the Tfjac experiment

QS methods	MRR scores	Rank	MAP scores	Rank	nDCG scores	Rank	P@10 scores	Rank
Tfidf	0.2934*	6*	0.9544	4	0.8927	6	0.9145*	6*
Tfcos	0.3254	4	0.9519	5	0.8989	5	0.9147*	5*
Jac	0.3211	5	0.9485	6	0.9209	4	0.9524	1
Jacos	0.3846	1	0.9695	2	0.9509	2	0.9524	1
Tfjac	0.3846	1	0.9712	1	0.9542	1	0.9524	1
Tfjacos	0.3687	3	0.9609	3	0.9347	3	0.9232	4

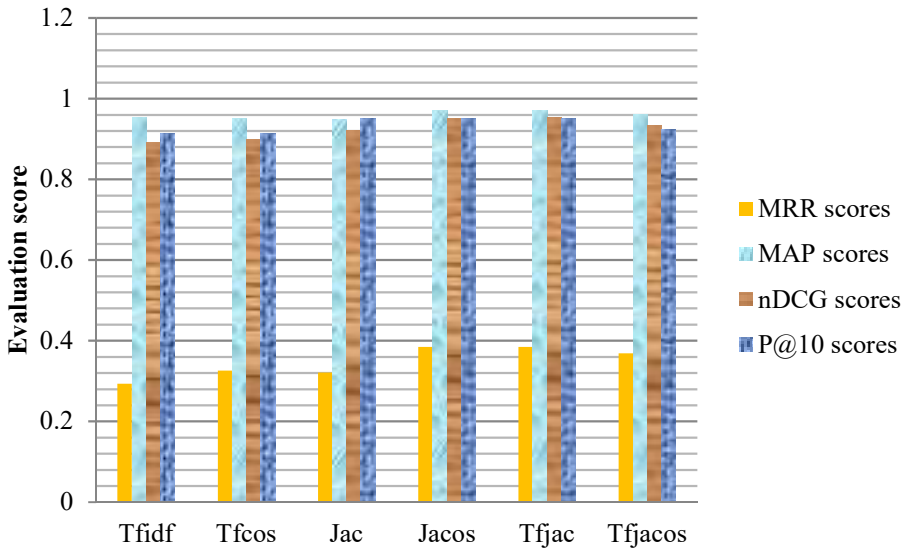


Figure 5.2 Experimental results of the Tfjac experiment

Table 5.3 Statistical test results of the Tfjac experiment

Evaluation method	QS method	Statistical test results (*p* value)
MRR	Tfjac vs Tfidf	0.0121
P@10	Tfjac vs Tfidf	0.0296
P@10	Tfjac vs Tfcos	0.0292

For the integrated evaluation, Table 5.4 illustrates the ranking orders of the six query suggestion methods in terms of the four evaluation methods respectively.

For the two methods, whose rankings are significantly lower than the others, the ranks are multiplied by two. The rankings in Table 5.4 can be transferred into MRR scores as shown in Table 5.5 and Figure 5.3. Overall, Tfjac is the best method to generate query suggestions followed by Jacos and Jac. The performances of Tfjacos, Tfcos and Tfidf methods were significantly worse than the other three methods as shown in Table 5.6.

Table 5.4 Summary of evaluation results

QS methods	MRR ranking	MAP ranking	nDCG ranking	P@10 ranking
Tfidf	6*(12)	4	6	6*(12)
Tfcos	4	5	5	5*(10)
Jac	5	6	4	1
Jacos	1	2	2	1
Tfjac	1	1	1	1
Tfjacos	3	3	3	4

Table 5.5 Integrated evaluation in MRR scores

QS methods	MRR scores	Rank
Tfidf	0.1458	6*
Tfcos	0.1875	5*
Jac	0.4042	3
Jacos	0.7500	2
Tfjac	1.0000	1
Tfjacos	0.3125	4*

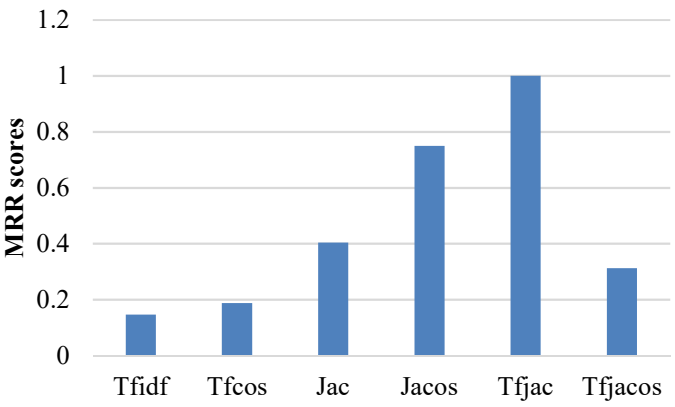


Figure 5.3 Integrated evaluation in MRR scores

Table 5.6 Statistical test results of integrated evaluation in MRR scores

QS method	Statistical test results (*p* value)
Tfjac vs Tfidf	0.0002
Tfjac vs Tfcos	0.0001
Tfjac vs Tfjacos	0.0000

Five PhD students studying in different fields participated in the user evaluation. Table 5.7 and Figure 5.4 illustrate the results of the user rankings in MRR scores. The majority of participants pointed out that the query suggestions made by Jacos were the best, followed by Tfjacos and Tfjac. However, they are not significantly different. It should be noted that only one top suggestion for each query was considered in the user evaluation here, which might lead to biased results and should be improved in future work.

Table 5.7 User evaluation in MRR scores

QS methods	MRR scores	Rank
Tfidf	0.6495	5
Tfcos	0.6549	4
Jac	0.6157	6
Jacos	0.7027	1
Tfjac	0.6732	3
Tfjacos	0.6909	2

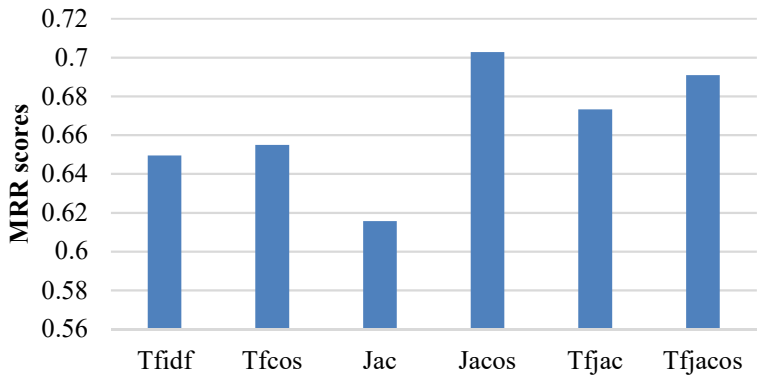


Figure 5.4 User evaluation in MRR scores

5.4.3.2 The GCrank experimental results

sGCrank and mGCrank methods which rank query suggestions using classification scores have been investigated and compared with Tfidf, Jac, and Tfjac in the GCrank experiment. The experimental results are shown in the following tables. The method for statistical significance test is the Wilcoxon rank-sum test with the p value ≤ 0.05 as significance level.

Five good suggestions from 10 query suggestions for each query were chosen and ranked by 3 participants. The results of evaluation are given in Table 5.8 and Figure 5.5. The results of evaluation using MRR show that sGCrank was the best method followed by mGCrank. The results of evaluation using MAP show that Tfjac was the best method for generating query suggestions in terms of ranking and producing relevant words, whilst mGCrank and sGCrank scored the lowest. The results of evaluation using nDCG show that Jac was the best method for ranking and producing highly relevant suggestions, followed by mGCrank and sGCrank. Finally, the results of evaluation using P@5 show that Tfjac and Tfidf have the same score and outperform other methods in terms of generating relevant suggestions. On the other hand, the scores from the mGCrank and sGCrank methods were the two lowest. However, there was no significant difference.

Table 5.8 Experimental results of GCrank experiment

QS methods	MRR scores	Rank	MAP scores	Rank	nDCG scores	Rank	P@5 scores	Rank
Tfjac	0.3429	4	0.9537	1	0.8003	4	0.9850	1
Jac	0.3566	3	0.9421	2	0.8221	1	0.9750	3
Tfidf	0.3418	5	0.9397	3	0.7974	5	0.9850	1
mGCrank	0.3627	2	0.9311	5	0.8191	2	0.9625	5
sGCrank	0.3674	1	0.9360	4	0.8153	3	0.9675	4

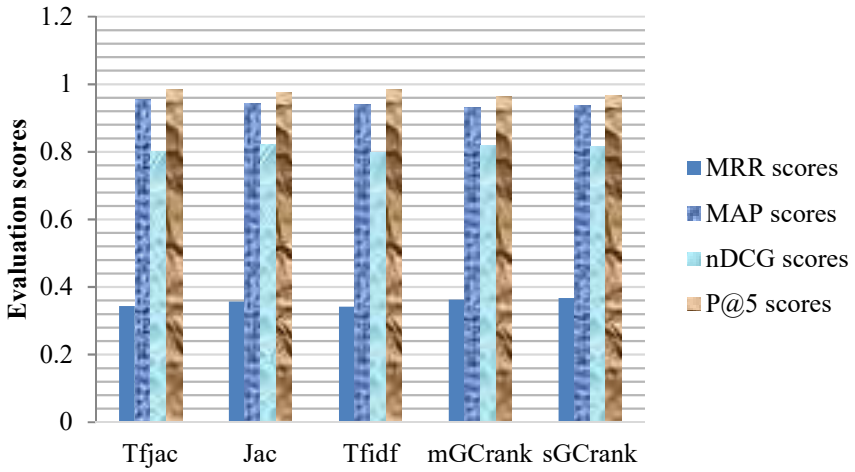


Figure 5.5 Experimental results of GCrank experiment

For an integrated evaluation, Table 5.9 illustrates the rankings of the five query suggestion methods in terms of the four evaluation methods respectively. The rankings in Table 5.9 can be transferred into MRR scores as shown in Table 5.10 and Figure 5.6. It is obvious that Tfjac is the best method overall to generate query suggestions followed by Jac and sGCrank. Even though the proposed GCrank based methods were not the best, their scores were not significantly different from the best one. In addition, the sGCrank method was better than the mGCrank method overall for producing query suggestions.

Table 5.9 Summary of evaluation results

QS methods	MRR ranking	MAP ranking	P@5 ranking	nDCG ranking
Tfjac	4	1	1	4
Jac	3	2	3	1
Tfidf	5	3	1	5
mGCrank	2	5	5	2
sGCrank	1	4	4	3

Table 5.10 Integrated evaluation in MRR scores

QS methods	MRR	MAP	P@5	nDCG	Avg	Rank
Tfjac	0.25	1	1	0.25	0.63	1
Jac	0.33	0.50	0.33	1	0.54	2
Tfidf	0.20	0.33	1	0.20	0.43	4
mGCrank	0.50	0.20	0.20	0.50	0.35	5
sGCrank	1	0.25	0.25	0.33	0.46	3

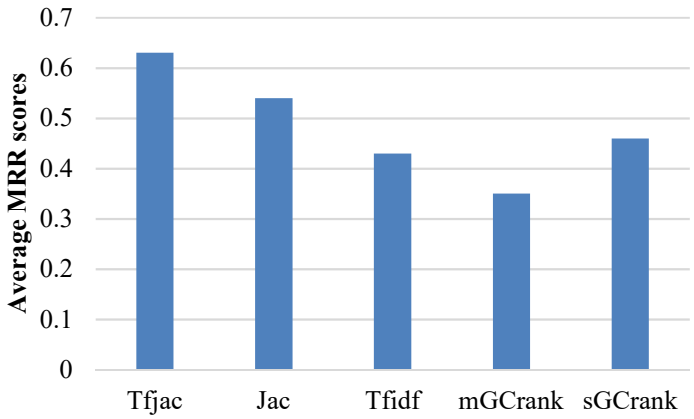


Figure 5.6 Integrated evaluation in average MRR scores

It seems that GCrank scores cannot help to improve the performance of query suggestions in the GCrank experiment. To confirm this conclusion, an additional experiment was investigated, which re-ranked query suggestions from the Tfjac method using GCrank scores. A document which has zero GCrank score does not belong to the same category as the original query. Therefore, if any query suggestion term appears in a document which has zero GCrank score, the rank of this term will be downgraded to the bottom of the list. The experimental results which are given in Table 5.11 and Figure 5.7 show that re-ranking query suggestions from the Tfjac method using GCrank scores does not result in improvement in terms of almost all evaluation criteria.

Table 5.11 Additional results

Methods	MRR	MAP	nDCG	P@5
Tfjac	0.3429	0.9537	0.8003	0.9850
Re-ranking using GCrank scores	0.3233	0.9541	0.7721	0.9825

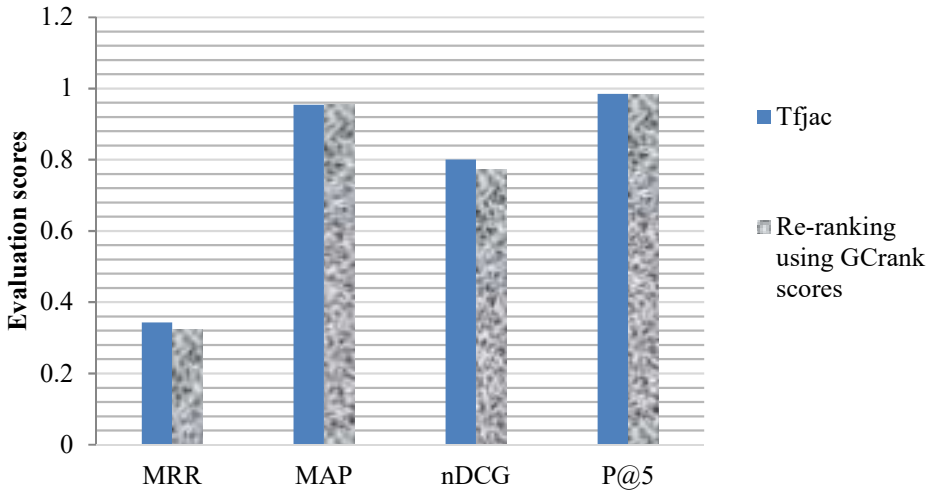


Figure 5.7 Additional results

5.4.3.3 The QS experimental results

This experiment aims to evaluate the effect of query suggestion on the search returned results. The Tfjac method was used for generating query suggestions for interactive web search. To evaluate the relevance of the top 8 Google search returned documents, query suggestions made from Tfjac method were used in comparison with that by using the original query only. The 80 test queries and the performance criteria were used here in the same way as the previous two experiments. However, the ranking is based on the relevance of the returned web documents rather than the quality of query suggestions directly. The relevance ranks of the returned web documents were obtained by 8 participants.

Table 5.12 Experimental results of QS experiment

Methods	MRR scores	MAP scores	nDCG scores	P@10 scores
Query	0.4618	0.9435*	0.9116	0.8422*
Query + suggestion	0.4452	0.9740	0.9402	0.9531

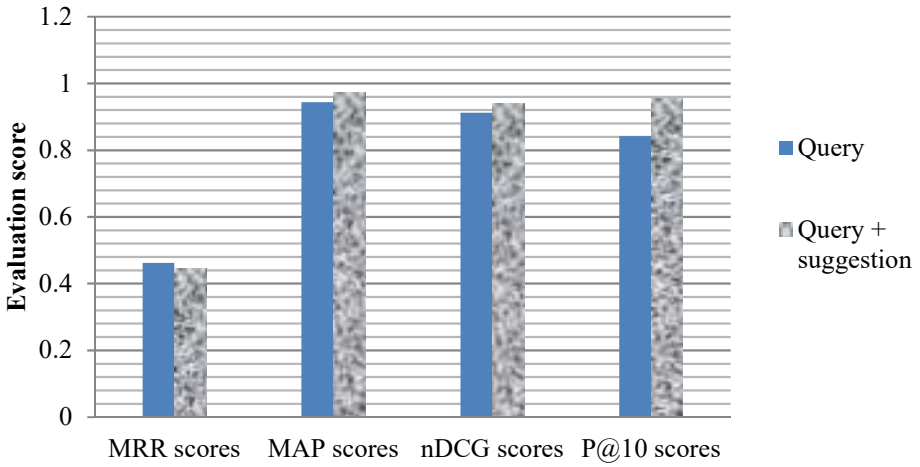


Figure 5.8 Experimental results of QS experiment

Table 5.13 Statistical test results of QS experiment

Methods	MRR scores	MAP scores	nDCG scores	P@10 scores
Query vs Query + suggestion	0.6199	0.0500*	0.2565	0.0000*

The results of evaluation are given in Table 5.12 and Figure 5.8, which show that in terms of the MRR scores the documents returned using the original query only were better ranked on average than those using query suggestions. However, there is no significant difference between the two MRR scores. In terms of MAP score and P@10 scores, the results show that the returned documents using query suggestion were significantly better than those using the original query only, as shown in Table 5.13. The results of evaluation using nDCG show that the nDCG score achieved by using query suggestion is higher than that of using the original query only, although there is no significant difference. In general, this experiment

determined that the proposed method, Tfjac, is effective and improves the relevance of the web returned documents through interactive web search.

5.5 Summary

This chapter of the thesis has investigated several ranked retrieval methods, and adapted and combined them for query suggestion. Six query suggestion methods, including the combined methods developed, have been evaluated using four performance criteria and user evaluation. The first experimental results show that Tfjac was the best to generate query suggestions among the six methods evaluated in terms of ranking and relevance. It is demonstrated that Tfjac can combine the best query suggestions from both TF-IDF and Jaccard coefficient methods. However, this combined method may deserve further investigation, and there may be room for further improvement by using better combination strategies.

It is also found that query suggestions re-ranking using cosine similarity help to generate better query suggestions in general. For example, the majority of the experimental results show that Jacos was the second-best method, which selected the query suggestion candidates from Jaccard coefficient and re-ranked the selected query suggestions using cosine similarity. Its top query suggestion was better than that of Tfjac, as shown in the user evaluation results. It should be noted that in the user evaluation conducted here, only the top suggestion for each query was evaluated. This is a limitation of the user evaluation when conducted in this way and should be further investigated. Furthermore, the experimental results also indicate that the query suggestions made by the Tfjac method significantly improved the relevance of the returned documents in interactive web search in terms of increasing the number of highly relevant documents or the precision.

A new approach to generate and rank query suggestions using classification scores has been proposed and investigated in the second experiment. The experimental results show that Tfjac was still the best to generate query suggestions among the five methods evaluated in terms of ranking and relevance, followed by Jaccard coefficient and sGCrank. However, two classification-score-based methods were among the top 2 best methods when evaluated using MRR and nDCG. From these results, it can be seen that these methods can generate many good and relevant suggestions ranked on the top of the lists selected by users. However, they also produced some irrelevant suggestions too. Furthermore, the sGCrank method was better than the mGCrank method overall for query suggestion. Therefore, term frequency does not help to improve the performance of query suggestions in this case. From these experiments, we can summarise that even though GCrank scores can improve the ranking of web returned documents from a search engine, they cannot help improve the performance of ranked query suggestions.

The queries used in the experiment and the judgment on the relevance of query suggestions with the original queries are the main factors for performance evaluation. 80 queries related to eight categories based on Google search results and users' suggestions have been designed in this thesis. Multiple evaluation criteria from different perspectives have been adopted to ensure a fair comparison and evaluation. However, further work should be investigated to overcome the limitations in this aspect of the performance and user evaluations.

Chapter 6 Conclusion

6.1 Summary of contributions

In this chapter, we summarise how this thesis work has achieved the research objectives set up in Section 1.2.

Firstly, a new term weighting technique has been proposed to improve the performance of document classification by using features sensitive to class memberships. The proposed weighting features for document representation, CSDF and TF-CSDF, are based on the assumption that class specific document frequency contains very important information for class discrimination under the VSM framework. The experimental results show that the CSDF based document representation is equal or better than other widely used VSM representations, including TF-IDF, in terms of classification accuracy on three datasets. Compared to the machine learning based method TFRF, the performance of CSDF was better than or equal to that of TFRF in general; however, the experimental results show that overfitting is a major issue for CSDF method. Furthermore, the experimental results show that not all supervised term weighting methods are better than unsupervised methods which are the same results as in [47].

In addition, the combination between TF and CSDF in appropriate proportion can achieve higher classification accuracy than the individual methods. TF-CSDF has similar simplicity and interpretability as TF-IDF and is more effective than TF-IDF for document representation for classification. We expect that CSDF as a new term weighting technique would be widely used in search engines and document databases for document representation or indexing.

In the investigation of using semantic information in term weighting, when comparing semantic features using the NLKT path_similarity scores, the results

show that the performance of semantic features is equal or lower than that of the other methods on three datasets. Semantic representation in this experiment is not as good as expected because the representative words may not be appropriate to present the classes, and there are a lot of proper nouns that are unknown to the NLKT path_similarity function.

Decision fusion has also been investigated for combining different term weighting techniques to improve document representation and classification. However, the combination of features and multiple classifiers yield only a small improvement on the performance of document classification. This means that there is not much new information added in the different features and multiple classifiers.

Secondly, a new ranking method called GCrank is proposed to improve the performance of web returned document ranking and thus user's satisfaction, which combines the Google ranking scores with LDA classification scores. It aims to downgrade a document which has a low classification score or is not classified as the same topic category as the query, and increase the rank of a document which has a high classification score. The experimental results show that the ranking of web returned documents by the GCrank method is significantly better than the original Google rank in terms of the integration of three evaluation criteria and the integrated evaluation. It means that this method can rank web returned documents more specific to user's information needs.

Thirdly, in order to improve the performance of query suggestion, the state-of-the-art ranked retrieval methods are investigated, adapted, and combines for effective query suggestion. This research proposes and investigates several ranked retrieval methods and combined them for query suggestion, which have been

evaluated using four performance criteria and user evaluation as well. The experimental results show that the proposed method, Tfjac, is the best for generating query suggestions among the six methods evaluated in terms of relevance and ranking. It is demonstrated that Tfjac is capable of combining good query suggestions from both TF-IDF and Jaccard coefficient methods. It is also found that query suggestions re-ranking using cosine similarity helps generate better query suggestions in general. Furthermore, a new approach to rank query suggestions using the classification scores is proposed and investigated. Two query suggestion methods, sGCrank and mGCrank, evaluated by comparing with other query suggestion methods such as Tfjac, using the same performance criteria. The experimental results show that Tfjac, is better than sGCrank and mGCrank for generating query suggestions. However, sGCrank and mGCrank can generate many good and relevant suggestions which are among those selected by users. It is also found that sGCrank method is better than mGCrank method for query suggestion in general. Therefore, term frequency does not help to improve the performance of query suggestions in this situation.

To sum up, this PhD thesis has resulted in new methods that help improve search results from search engines or document databases and thus increase user's satisfaction with search results. Specifically, CSDF is a new term weighting technique for document representation which can improve the performance of document classification in general. GCrank can be used to improve web documents ranking in the sense that the documents mostly meeting user's information needs appears first. Finally, Tfjac is a new method for query suggestion, which can provide useful query suggestions for effective interactive web search.

6.2 Limitations and future work

Although the research has achieved its goals in general, there are some limitations. First of all, because of the time limit, the number of human participants and the number of documents adopted in the experiments are small. Secondly, in the performance evaluation, subjective bias is another main concern. For example, the evaluation of performance usually relies on the queries used in the experiment, and the judgment on the relevance of query suggestions with the original queries. This thesis adopted multiple evaluation criteria from different perspectives to ensure a fair comparison and evaluation. However, the future work should be conducted to overcome the limitation on this aspect of the performance evaluation and on the user evaluation conducted in the experiment. Thirdly, due to the limited time and the complexity of some state-of-the-art methods, such as machine learning based ranking methods, they were not considered in the experiment for evaluating the proposed methods. Finally, most experiments were conducted offline, without emphasizing the time complexity of the proposed methods required for online applications.

Due to the limitations mentioned above, some ideas for future work are suggested in the remaining part of this section. Firstly, the document representation in this research focused on VSM representation with single word only as basic unit. Other grammatical units such as phrase, clause, or sentence, may have more representation power. In addition, graph-based representation can reveal the structure of the documents in the graph. This type of representation can provide more information necessary for document classification than the VSM framework. Although the semantic representation investigated in this research did not improve the performance of the document classification, there are many

ontologies or knowledge bases available such as Wikipedia or YAGO2s, which might be more effective.

Secondly, regarding document ranking methods, learning to rank has not been investigated in this thesis, but it is a worthy topic for future research. In addition, the number of topic categories, the size of web documents, and the number of participants in user evaluation are relatively small in the experiments in this thesis, more extensive testing and deeper investigation would be required in future research in order to draw more convincing conclusions.

Thirdly, regarding query suggestion methods, there are other sources available for producing query suggestions, such as log files and clickthrough logs, which would be useful for making better query suggestions, especially for personalised query suggestion. Most experiments in this thesis have been done with offline processing. It would be interesting to extend the experiments with online applications such as for adaptive search engine and adaptive query suggestion.

References

[1] S. Brin and L. Page, “The anatomy of a large-scale hypertextual web search engine,” Google, [Online]. Available: http://infolab.stanford.edu/backrub/google.html. [Accessed 2 January 2016].

[2] C. D. Manning, P. Raghavan and H. Schutze, Introduction to Information Retrieval, Cambridge: Cambridge University Press, 2008.

[3] B. M. Fonseca, P. B. Golgher, E. S. de Moura and N. Ziviani, “Using association rules to discover search engines related queries,” in *The First Latin American Web Congress*, USA, 2003, pp. 66-71.

[4] M. Delgado, M. Martin-Bautista, D. Sanchez, J. Serrano and M. Vila, “Association rules and fuzzy association rules to find new query terms,” in *EUSFLAT*, Lisbon, Portugal, 2009, pp. 49-53.

[5] S. Robertson, “On the history of evaluation in IR,” *Information Science,* vol. 34, no. 4, pp. 439-456, 2008.

[6] A. Bratko, G. V. Cormack, B. Filipic, T. R. Lynam and B. Zupan, “Spam filtering using statistical data compression models,” *Machine Learning Research,* vol. 7, pp. 2673-2698, 2006.

[7] S. Busemann, S. Schmeier and R. G. Arens, “Message classification in the call center,” in *The 6th Applied Natural Language Processing Conference*, Stroudsburg, PA, USA, 2000, pp. 158-165.

[8] S. Marina and M. Rosso, “Testing a genre-enabled application: a preliminary assessment,” in *The BCS IRSG Symposium: Future Directions in Information Access*, London, UK, 2008, pp.54-63.

[9] T. Strzalkowski, “Document representation in natural language text retrieval,” in *The Workshop on Human Language Technology*, Plainsboro, NJ, USA, 1994, pp. 364-369.

[10] R. Baeza-Yates and B. Ribeiro-Neto, Modern Information Retrieval: The Concept and Technology behind Search, England: Pearson Education Limited, 2011.

[11] B. He and I. Ounis, “Studying query expansion effectiveness,” in *The 31th Eruopean conference on IR research on advances in IR*, Toulouse, France, 2009, pp. 611-619.

[12] D. Kelly, A. Cushing, M. Dostert, X. Niu and K. Gyllstrom, “Effects of popularity and quality on the usage of query suggestions during information search,” in *SIGCHI Conference on Human Factors in Computing Systems*, Atlanta, USA, 2010, pp. 45-54.

[13] J. Han, M. Kamber and J. Pei, Data Mining: Concepts and Techniques, Elsevier, 2011.

[14] M. Clark, Y. Kim, U. Kruschwitz, D. Song, D. Albakour, S. Dignum, U. C. Beresi, M. Fasli and A. D. Roeck, “Automatically structuring domain knowledge from text: an overview of current research,” *Information Processing and Management,* vol. 48, no. 3, pp. 552-568, 2012.

[15] L. Todorovski and S. Dzeroski, “Intergrating knowledge-driven and data-driven approachs to modeling,” *Ecological Modelling,* vol. 194, no. 1, pp. 3-13, 2006.

[16] D. Solomatine, L. M. See and R. Abrahart, “Data-driven modelling: concepts, approaches and experiences,” in *Practical Hydroinformatics*, Springer Berlin heidelberg, 2008, pp. 17-30.

[17] C. T. Meadow, Text Information Retrieval Systems, Academic Press, 1992.

[18] S. M. Weiss, N. Indurkhya and T. Zhang, Fundamentals of Predictive Text Mining, NY, USA: Springer Science and Business Media, 2010.

[19] H. Kim, R. Xiang, S. Yizhou, C. Wang and J. Han, “Semantic frame-based document representation for comparable corpora,” in *IEEE 13th International Conference on Data Mining*, Dallas, TX, USA, 2013, pp. 350-359.

[20] M. Keikha, A. Khonsari and F. Oroumchian, “Rich document representation and classification: an analysis,” *Knowledge-Based Systems,* vol. 22, no. 1, pp. 67-71, 2009.

[21] A. Markov, M. Last and A. Kandel, “The hybrid representation model for web document classification,” *Intelligent Systems,* vol. 23, no. 6, pp. 654-679, 2008.

[22] K. K. Phukon, “A composite graph model for web document and the MCS technique,” *International Journal of Multimedia and Ubiquitous Engineering,* vol. 7, no. 1, pp. 45-52, 2012.

[23] K. Valle and P. Ozturk, “Graph-based representations for text classification,” in *India-Norway Workshop on Web Concepts and Technologies*, Trondheim, Norway, 2011, pp. 2363-2366.

[24] S. K. George and S. Joseph, “Text classification by augmenting bag of words (BOW) representation with co-occurrence feature,” *IOSR Journal of Computer Engineering,* vol. 16, no. 1, pp. 34-38, 2014.

[25] B. Harish, S. A. Kumar and S. Manjunath, “Classifying text documents using unconventional representation,” in *Big Data and Smart Computing (BIGCOMP)*, Bangkok, Thailand, 2014, pp. 210-216.

[26] P. Turney and P. Pantel, “From frequency to meaning: vector space models of semantics,” *Artificial Intelligence Research,* vol. 37, pp. 141-188, 2010.

[27] B. Haris, D. Guru and S. Manjunath, “Representation and classification of text document: a brief review,” *Computer Applications, Special Issue on Recent Trends in Image Processing and Pattern Recognition,* vol. 2, no. 2, pp. 110-119, 2010.

[28] J. Dobsa, “Algorithm for classification of textual documents represented by tandem analysis,” in *Data Mining and Data Warehouses*, Ljubljana, Slovenia, 2014, pp. 9-12.

[29] Q. Le and T. Mikolov, “Distributed representations of sentences and documents,” in *The 31th International Conference on Machine Learning*, Beijing, China, 2014, pp. 1188-1196.

[30] J. K. C. Chung, C.-E. Wu and R. T.-H. Tsai, “Improve polarity detection of online reviews with bag-of-sentimental-concepts,” in *Semantic Web Evaluation Challenge*, Crete, Greece, 2014, pp. 379-420.

[31] M.-S. Paukkeri, M. Ollikainen and T. Honkela, “Assessing user-specific difficulty of documents,” *Information Processing and Management,* vol. 49, no. 1, pp. 198-212, 2013.

[32] M. Radovanovic and M. Ivanovic, “Document representations for classification of short web-page describtions,” *Data Warehousing and Knowledge Discovery Lecture Notes in Computer Science,* vol. 4081, pp. 544-553, 2006.

[33] K. Supreeth and E. Prasad, “A novel document representation model for clustering,” *Computer Science and Communication (IJCSC),* vol. 1, no. 2, pp. 243-245, 2010.

[34] N. C. Thanh and K. Yamada, “Document representation and clustering with WordNet based similarity rough set model,” *Computer Science Issues (IJCSI),* vol. 8, no. 5, 2011.

[35] M. Keller and S. Bengio, “Theme topic mixture model for document representation,” in *PASCAL Workshop on Text mining and Understanding*, Meylan (Grenoble), France, 2004.

[36] X. He, D. Cai, H. Liu and W.-Y. Ma, “Locality preserving indexing for document representation,” in *The 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, South Yorkshire, UK, 2004, pp. 96-103.

[37] W. Yih and V. Qazvinian, “Measuring word relatedness using heterogeneous vector space models,” in *The North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, Montreal, Canada, 2012, pp. 616-620.

[38] F. Raja, M. Keikha, M. Rahgozar and F. Oroumchian, “Effectiveness of rich document representation in XML retrieval,” in *The 8th RIAO Conference on Large-Scale Semantic Access to Content (Text, Image, Video and Sound)*, Pittsburgh, PA, USA, 2007, pp. 241-250.

[39] G. Paltoglou and M. Thelwall, “More than bag-of-words: sentence-based document representation for sentiment analysis,” in *Recent Advances in Natural Language Processing*, Hissar, Bulgaria, 2013, pp. 546-552.

[40] S. Laroum, N. Bechet, H. Hamza and M. Roche, “HYBRED: an OCR document represenatation for classification tasks,” *Computer Science Issues (IJCSI),* vol. 8, no. 3, pp. 1-8, 2011.

[41] L. Ying, On Document Representation and Term Weights in Text Classification, Hong Kong: The Hong Kong Polytechnic University, 2009.

[42] I. A. El-Khair, “Term weighting,” in *Encyclopedia of Database Systems*, USA, Springer, 2009, pp. 3037-3040.

[43] S. E. Robertson, “On term selection for query expansion,” *Journal of Documentation,* vol. 46, pp. 359-364, 1990.

[44] N. Nanas, V. Uren, A. D. Roeck and J. Dominque, “A comparative study of term weighting methods for information filtering,” in *The 15th International Workshop on Database and Expert Systems Applications*, Zaragoza, Spain, 2004, pp. 13-17.

[45] F. Ren and M. G. Sohrab, “Class-indexing-based term weighting for automatic text classification,” *Information Science,* vol. 236, pp. 109-125, 2013.

[46] Q. Luo, E. Chen and H. Xiong, “A semantic term weighting scheme for text categorization,” *Expert Systems with Applications,* vol. 38, pp. 12708-12716, 2011.

[47] M. Lan, C. L. Tan, J. Su and Y. Liu, “Supervised and traditional term weighting methods for automatic text categorization,” *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 31, no. 4, pp. 721-735, 2009.

[48] J. Gautam and E. Kumar, “An integrated and improved approach to terms weighting in text classification,” *Computer Science Issues (IJCSI),* vol. 10, no. 1, pp. 310-314, 2013.

[49] B. Trstenjaka, M. Sasa and D. Donko, “KNN with TF-IDF based framework for text categorization,” *Procedia Engineering,* vol. 69, pp. 1356-1364, 2014.

[50] F. Debole and F. Sebastiani, “Supervised term weighting for automated text categorization,” *Text Mining and Its Applications,* vol. 138, pp. 81-97, 2004.

[51] K. S. Jones, “A statistical interpretation of term specificity and its application in retrieval,” *Journal of Documentation,* vol. 28, pp. 11-21, 1972.

[52] S. Robertson, “Understanding inverse document frequecy: one theoretical arguments for IDF,” *Journal of Documentation,* vol. 60, no. 5, pp. 503-520, 2004.

[53] D. Jurafsky and J. H. Martin, Speech and Language Processing: An Introduction to Natural Language Processing, Prentice Hall, 2008.

[54] G. Salton and C. Buckley, “Term-weighting approaches in automatic text retrieval,” *Information Processing and Management,* vol. 24, pp. 513-523, 1988.

[55] S. Flora and T. Agus, “Experiments in term weighting for novelty mining,” *Expert Systems with Applications,* vol. 38, pp. 14094-14101, 2011.

[56] L. Yang, C. Lia, Q. Dingb and L. Lib, “Combining lexical and semantic features for short text classification,” *Procedia Computer Science,* vol. 22, pp. 78-86, 2013.

[57] S. Bloehdorn and A. Hotho, “Boosting for text classification with semantic features,” in *International Workshop on Knowledge Discovery on the Web* , Springer Berlin Heidelberg, 2004, pp. 149-166.

[58] L. F. Lai, C. C. Wu, P. Y. Lin and L. T. Huang, “Developing a fuzzy search engine based on fuzzy ontology and semantic search,” in *IEEE Internation Conference on Fuzzy Systems*, Taiwan, 2011, pp. 2684-2689.

[59] M. Boicu, G. Tecuci, B. Stanescu, G. C. Balan and E. Popovici, “Ontologies and the knowledge acquisition bottleneck,” in *The 17th International Joint Conference on Artificial Intelligence (IJCAI)*, Seattle, USA, 2001, pp. 2684-2689.

[60] F. Suchanek, J. Hoffart, E. Kuzey and E. Lewis-Kelham, “YAGO2s:Modular high-quality information extraction with an application to flight planning,” in *The German Computer Science Symposium (BTW)*, Magdeburg, Germany, 2013, pp. 515-518.

[61] F. Suchanek, G. Kasneci and G. Weikum, “YAGO: a core of semantic knowledge unifying WordNet and Wikipedia,” in *World Wide Web*, Banff,Alberta, Canada, 2007, pp. 697-706.

[62] X. Peng and B. Choi, “Document classifications based on word semantic hierarchies,” *AI and Applications,* vol. 5, pp. 362-367, 2005.

[63] E. Ferretti, M. Errecalde and P. Rosso, “Does semantic information help in the text categorization task?,” *Intelligent Systems,* vol. 17, no. 1, pp. 91-106, 2008.

[64] R. Nagaraj, V. Thiagarasu and P. Vijayakumar, “A novel semantic level text classification by combining NLP and thesaurus concepts,” *Computer Engineering (IOSR-JCE),* vol. 16, no. 4, pp. 14-26, 2014.

[65] M. Lan and H.-b. Low, “A comprehensive comparative study on term weighting schemes for text categorization with support vector machines,” in *The 14th International World Wide Web Conference*, Chiba, Japan, 2005, pp. 1032-1033.

[66] M. Lan, C.-L. Tan and H.-B. Low, “Proposing a new term weighting scheme for text categorization,” *AAAI,* vol. 1, pp. 763-768, 2006.

[67] M. Mohri, A. Rostamizadeh and A. Talwalkar, Foundations of Machine Learning, The MIT Press, 2012.

[68] R. O. Duda, P. E. Hart and D. G. Stork, Pattern Classification, Wiley-Interscience, 2000.

[69] I. Guyon and A. Elisseeff, “An introduction to variable and feature selection,” *Machine Learning Research,* vol. 3, pp. 1157-1182, 2003.

[70] R. G. Osuna, “Sequential Feature Selection,” Texas A&M, [Online]. Available: http://www.facweb.iitkgp.ernet.in/~sudeshna/courses/ML06/featsel.pdf. [Accessed 1 December 2015].

[71] J. Q. Gan, B. A. Shiekh and C. S. L. Tsui, “A filter-dominating hybrid sequential forward floating search method for feature subset selection in high-dimensional space,” *International Journal of Machine Learning and Cybernetics,* vol. 5, no. 3, pp. 413-423, 2014.

[72] N. Altman, “An introduction to kernel and nearest-neighbor nonparametric regression,” *The American Statistican,* vol. 46, no. 3, pp. 175-185, 1992.

[73] G. Mclachlan, Discriminant Analysis and Statistical Pattern Recognition, John Wiley & Sons, 2004.

[74] J. Kalina and J. D. Tebbens, “Algorithm for regularized linear discriminant analysis,” in *Biomedical Engineering Systems and Technologies*, Lisbon, Portugal, 2015, pp. 128-133.

[75] J. D. Mcauliffe and D. M. Blei, “Supervised topic models,” *Advances in Neural Information Processing Systems,* pp. 121-128, 2008.

[76] S. Theodoridis and K. Koutroumbas, Pattern Recognition, 4th Edition, Academic Press, 2009.

[77] D. Hall and J. Llinas, “An introduction to multi-sensor data fusion,” in *IEEE International Symposium on Circuits and Systems*, Monterey, CA, USA, 1998, pp. 6-23.

[78] F. Castanedo, “A review of data fusion techniques,” *The Scientific Word Journal,* vol. 2013, pp. 1-19, 2013.

[79] H. F. Durrant-Whyte, “Sensor model and multisensor integration,” *Robotics Research,* vol. 7, no. 6, pp. 97-113, 1988.

[80] M. S. Mahmoud and Y. Xia, Networked filtering and fusion in wireless sensor networks, CRC Press, 2014.

[81] B. V. Dasarathy, “Sensor fusion potential exploitation-innovative architectures and illustrative applications,” *Proceedings of the IEEE,* vol. 85, no. 1, pp. 24-38, 1997.

[82] U. Mangai, S. Samanta, S. Das and P. Chowdhury, “A survery of decision fusion and feature fusion strategies for pattern classification,” *IETE Technical Review,* vol. 27, no. 4, pp. 293-307, 2010.

[83] D. Ruta and B. Gabrys, “An overview of classifier fusion methods,” *Computing and Information Systems,* vol. 7, no. 1, pp. 1-10, 2000.

[84] J. Yang, J. Yang, D. Zhang and J. Lu, “Feature fusion: parallel strategy vs. serial strategy,” *Pattern Recognition,* vol. 36, no. 6, pp. 1369-1381, 2003.

[85] M. Bhowmik, P. Saha, G. Majumder and D. Bhattacharjee, “Decision fusion of multisensor images for human face identification in information security,” in *Handbook of Research on Computational Intelligence for Engineering, Science and Business*, USA, IGI Global, 2012, pp. 571-591.

[86] V. Dasigi, R. C. Mann and V. A. Protopopescu, “Information fusion for text classification-- an experimental comparison,” *Pattern Recognition,* vol. 34, no. 12, pp. 2413-2425, 2001.

[87] A. Danesh, B. Moshiri and O. Fatemi, “Improve text classification accuracy based on classifier fusion methods,” in *The 10th International Conference on Information Fusion*, Quebec, Canada, 2007, pp. 1-6.

[88] X. D. Zhang, “A general decision layer text classification fusion model,” in *The 2nd International Conference on Education Technology and Computer*, Shanghai, China, 2010, pp. V5-239.

[89] A. Mohan, Z. Chen and K. Q. Weinberger, “Web-search ranking with initialized gradient boosted regression trees,” in *Yahoo, Learning to Rank Challenge*, 2011, pp. 77-89.

[90] Y. Du and Y. Hai, “Semantic ranking of web pages based on formal concept analysis,” *Systems and Software,* vol. 86, pp. 187-197, 2013.

[91] B. Xiang, D. Jiang, J. Pei, X. Sun, E. Chen and H. Li, “Context-aware ranking in web search,” in *The 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, Geneva, Switzerland, 2010, pp. 451-458.

[92] V. Derhami, E. Khodadadian, M. Ghasemzadeh, A. Mohammad and Z. Bidoki, “Applying reinforcement learning for web pages ranking algorithm,” *Applied Soft Computing,* vol. 13, pp. 1686-1692, 2013.

[93] Y. Lu, Y. Li, M. Xu and W. Hu, “A user model based ranking method of query results of meta-search engines,” in *Network and Information Systems for Computers (ICNISC)*, Wuhan, China, 2015, pp. 426-430.

[94] H. Wang, X. He, M. Chang, Y. Song, R. White and W. Chu, “Personalized ranking model adaptation for web search,” in *The 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, USA, 2013, pp. 323-332.

[95] V. Jindal, S. Bawa and S. Batra, “A review of ranking approaches for semantic search on web,” *Information Processing and Management,* vol. 50, pp. 416-425, 2014.

[96] J. GarciA, M. Junghans, D. Ruiz, S. Agarwal and A. Ruiz-CorteS, “Integrating semantic web service ranking mechanisms,” *Knowledge-Based Systems,* vol. 49, pp. 22-36, 2013.

[97] J. Lee, J. Min, A. Oh and C. Chung, “Effective ranking and search techniques for web resources considering semantic relationships,” *Information Processing and Manangement,* vol. 50, pp. 132-155, 2014.

[98] Z. Zhuang and S. Cucerzan, “Re-ranking search results using query logs,” in *The 15th ACM International Conference on Information and Knowledge Management*, Virginia, USA, 2006, pp. 860-861.

[99] E. Alkhalifa, “Investigating bias in the page ranking approach,” in *IEEE Information and Communication Technology Research (ICTRC)*, 2015, pp. 294-297.

[100] R. Baezy-Yates and E. Davis, “Web page ranking using link attributes,” in *World Wide Web*, New York, USA, 2004, pp. 328-329.

[101] R. Baeza-yates and B. Ribeiro-Neto, Modern Information Retrieval: The Concepts and Technology behind Search, England: Addison Wesley Longman Limited, 1999.

[102] R. Nallapati and C. Shah, “Evaluating the quality of query refinement suggestion,” in *Information and Knowledge Management (CIKM)*, Virginia, USA, 2006.

[103] M. Costa, J. Miranda, D. Cruz and D. Gomes, “Query suggestion for web archive search,” in *Preservation of Digital Objects (iPres)*, Lisbon, Portugal, 2013.

[104] M. Kato, T. Sakai and K. Tanaka, “Structured query suggestion for specialization and parallel movement,” in *World Wide Web*, Lyon, France, 2012, pp. 389-398.

[105] J. Xu and W. B. Croft, “Query Expansion Using Local and Global Document Analysis,” in *The 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* , Zurich, Switzerland, 1996, pp. 4-11.

[106] U. Kruschwitz, D. Lungley, M.-D. Albakour and D. Song, “Deriving query suggestion for site search,” *The Association for Information Science and Technology (JASIST),* vol. 64, no. 10, pp. 1975-1994, 2013.

[107] I. Adeyanju, D. Song, M. D. Albakour, U. Kruschwitz, A. D. Roeck and M. Fasli, “Learning from users' querying experience on intranets,” in *World Wide Web*, Lyon, France, 2012, pp. 755-764.

[108] M. P. Kato, T. Sakai and K. Tanaka, “Query session data vs clickthrough data as query suggestion resources,” in *The 33rd European Conference on Information Retrieval (ECIR)*, Dublin, Ireland, 2011.

[109] R. Baeza-Yates, C. Hurtado and M. Mendoza, “Query recommendation using query logs in search engines,” in *International Conference on Extending Database Technology*, Heraklion, Crete, Greece, 2004, pp. 588-596.

[110] P. Boldi, F. Bonchi, C. Castillo, D. Donato and S. Vigna, “Query suggestion using query flow graphs,” in *Workshop on Web Search Click Data*, Milan, Italy, 2009, pp. 56-63.

[111] P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis and S. Vigna, “The query flow graph: model and applications,” in *International Conference on Information and Knowledge Management (CIKM)*, California, USA, 2008, pp. 609-618.

[112] H. Cao, D. Jiang, J. Pei, Q. He, Z. Liao, E. Chen and H. Li, “Context-aware query suggestion by mining click-through and session data,” in *The 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Nevada, USA, 2008, pp. 875-883.

[113] C. Huang, L. Chien and Y. Oyang, “Relevant term suggestion in interactive web search based on contextual information in query session logs,” *American Society for Information Science and Technology,* vol. 54, no. 7, pp. 638-649, 2003.

[114] Z. Liao, Y. Song, Y. Huang, L. He and Q. He, “Task trail: an effective segmentation of user search behavior,” *IEEE Transactions on Knowledge and Data Engineering,* vol. 26, no. 12, pp. 3090-3102, 2014.

[115] Q. Mei, D. Zhou and K. Church, “Query suggestion using hitting time,” in *The 17th ACM Conference on Information and Knowledge Management (CIKM)*, California, USA, 2008, pp. 469-478.

[116] Z. Gong, C. Cheang and L. Hou, “Web query expansion by WordNet,” *Lecture Notes in Computer Science (LNCS),* vol. 3588, pp. 166-175, 2005.

[117] J. Wan, W. Wang, J. Yi, C. Chu and K. Song, “Query expansion approach based on ontology and local context analysis,” *Applied Sciences, Engineering and Technology,* vol. 4, no. 16, pp. 2839-2843, 2012.

[118] H. Hu, M. Zhang, Z. He, P. Wang and W. Wang, “Diversifying query suggestions by using topics from Wikipedia,” in *Web Intelligence and Intelligent Agent Technology*, Atlanta, USA, 2013, pp. 139-146.

[119] J. Biega, E. Kuzey and F. Suchanek, “Inside YAGO2s: A transparent information extraction architechture,” in *World Wide Web* , Rio de Janeiro, Brazil, 2013, pp. 325-328.

[120] J. Hoffart, F. Suchanek, K. Berberich, E. Lewis-Kelham, G. Melo and G. Weikum, “YAGO2: exploring and querying world knowledge in time, space, context, and many languages,” in *World Wide Web* , Hyderabad, India, 2011, pp. 229-232.

[121] J. Yang, R. Cai, F. Jing, S. Wang, L. Zhang and W. Ma, “Search-based query suggestion,” in *The 17th ACM Conference on Information and Knowledge Management* , California, USA, 2008, pp. 1439-1440.

[122] Y. Song, D. Zhou and L. He, "Query suggestion by constructing term-transition graphs," in *The 5th ACM International Conference on Web Search and Data Mining (WSDM)*, Seattle, Washingtion, USA, 2012, pp. 353-362.

[123] M. H. Hsu, M. F. Tsai and H. H. Chen, "Query expansion with conceptnet and wordnet: An intrinsic comparison," in *The 3rd Asia Information Retrieval Symposium (AIRS)*, Singapore, 2006, pp. 1-13.

[124] R. Navigli and P. Velardi, "An analysis of ontology-based query expansion strategies," in *The ECML Workshop on Adaptive Text Extraction and Mining (ATEM)*, Cavtat Dubrovnik, Croatia, 2003, pp. 42-49.

[125] L. Van der Plas and J. Tiedemann, "Using lexico-semantic information for query expansion in passage retreival for question answering," in *The COLING Workshop on Information Retrieval for Question Answering (IRQA)*, Manchester, UK, 2008, pp. 50-57.

[126] R. Zanon, S. Albertini, M. Carullo and I. Gallo, "A new query suggestion algorithm for taxonomy-based search engines," in *International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (KDIR)*, Barcelona, Spain, 2012, pp. 151-156.

[127] S. Rieh and H. Xie, "Analysis of multiple query reformulations on the web: the interactive information retrieval context," *Information Processing and Management,* vol. 42, no. 3, pp. 751-768, 2006.

[128] D. Beeferman and A. Berger, “Agglomerative clustering of search engine query log,” in *The Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, New York, USA, 2000, pp. 407-416.

[129] I. Adeyanju, D. Song, M. Albakour, U. Kruschwitz, A. D. Roeck and M. Fasli, “Adaptation of the concept hierarchy model with search logs for query recommendation on Intranets,” in *The 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, USA, 2012, pp. 5-14.

[130] M. Sanderson and B. Croft, “Deriving concept hierarchies from text,” in *The 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, CA, USA, 1999, pp. 206-213.

[131] H. Joho, M. Sanderson and M. Beaulieu, “Hierarchical approach to term suggestion device,” in *The 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* , Tampere, Finland, 2002, pp. 454-454.

[132] D. Poshyvanyk and A. Marcus, “Combining formal concept analysis with information retrieval for concept location in source code,” in *The 15th IEEE Internation Conference on Program Comprehension (ICPC)*, Banff, Alberta, Canada, 2007, pp. 37-48.

[133] C. Zhai and J. Lafferty, “A study of smoothing methods for language models applied to adhoc information retrieval,” in *The 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* , New Orleans, LA, USA, 2001, pp. 334-342.

[134] J. Huang and E. N. Efthimiadis, “Analyzing and evaluating query reformulation strategies in web search logs,” in *The 18th ACM Conference on Information and Knowledge Management*, China, 2009, pp. 77-86.

[135] L. A. Tuan and J. J. Kim, “Automatic suggestion for PubMed query reformulation,” *Computing Science and Engineering,* vol. 6, no. 2, pp. 161-167, 2012.

[136] I. Rish, “An empirical study of the naive Bayes classifier,” in *International Joint Conferences on Artificial Intelligence (IJCAI)* , New York, 2001, pp. 41-46.

[137] A. McCallum, D. Freitag and F. Pereira, “Maximum entropy Markov models for information extraction and segmentation,” in *The 17th International Conference on Machine Learning*, Stanford, CA, USA, 2000, pp. 591-598.

[138] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine Learning,* vol. 20, no. 3, pp. 273-297, 1995.

[139] Y. Kim, J. Seo and W. B. Croft, “Automatic boolean query suggestion for professional search,” in *ACM SIGIR Special Interset Group on Information Retrieval*, China, 2011, pp. 825-834.

[140] U. Ozertem, O. Chapelle, P. Dommez and E. Velipasaoglu, “Learning to suggest: a machine learning framework for ranking query suggestions,” in *The 35th International ACM SIGIR Conference on Research and Development in Information Retrieval* , Portland, Oregon, USA, 2012, pp. 25-34.

[141] G. Bordogna, P. Carrara and G. Pasi, “Fuzzy approaches to extend boolean information retrieval,” in *Fuzziness in Database Management Systems*, Germany, Physica Verlag, 1995, pp. 231-274.

[142] G. Bordogna and G. Pasi, “A fuzzy linguistic approach generalizing boolean information retrieval: a model and its evaluation,” *The American Society for Information Science,* vol. 44, no. 2, pp. 70-82, 1993.

[143] D. Broccolo, O. Frieder, F. M. Nardini, R. Perego and F. Silvestri, “Incremental algorithm for effective and efficient query recommendation,” in *String Processing and Information Retrieval (SPIRE)*, Los Cabos, Mexico, 2010, pp. 13-24.

[144] S. Muthukrishnan, “Data streams: algorithm and applications,” *Foundations and Trends in Theoretical CS,* vol. 1, no. 2, pp. 117-236, 2005.

[145] M. Dorigo, G. Caro and L. Gambardella, “Ant colony algorithms for discrete optimization,” *Artificial Life,* vol. 5, no. 3, pp. 137-172, 1999.

[146] D. Martens, M. D. Backer, J. Vanthienen, M. Snoeck and B. Baesens, “Classification with Ant Colony Optimization,” *IEEE Transactions on Evaluationary Computation,* vol. 11, pp. 651-665, 2007.

[147] S. Dignum, U. Kruschwitz, M. Fasli, Y. Kin, D. Song, U. Cervino and A. D. Roeck, “Incorporating seasonality into search suggestions devired from Intranet query logs,” in *IEEE/WIC/ACM International Conferences on Web Intelligence*, USA, 2010, pp. 425-430.

[148] S. Dignum, Y. Kim, U. Kruschwitz, D. Song, M. Fasli and A. D. Roeck, “Using Domain model for context-rich user logging,” in *Workshop on Understanding the User - Logging and Interpreting User Interactions in Information Search and Retrieval (UIIR)*, Boston, USA, 2009, pp.48-61.

[149] U. Kruschwitz, M. D. Albakour, J. Niu, J. Leveling, N. Nanas, Y. Kim, D. Song, M. Fasli and A. D. Roeck, “Moving towards adaptive search in digital libraries,” in *International Conference on Advanced Language Technologies for Digital Libraries*, Italy, 2009, pp. 41-60.

[150] M. Okabe and S. Yamada, “Semi-supervised query expansion with minimal feedback,” *IEEE Transaction on Knowledge and Data engineering,* vol. 19, no. 11, pp. 1585-1589, 2007.

[151] A. Otegi, X. Arregi and E. Agirre, “Query expansion for IR using knowledge-based relatedness,” in *The 5th International Joint Conference on Natural Language Processing*, Changmai,Thailand, 2011, pp. 1467-1471.

[152] L. Dybkjaer, H. Hemsen and W. Minker, Evaluation of Text and Speech Systems, Netherland: Springer Science & Business Media, 2007.

[153] W. B. Powell, Approximate Dynamic Programming: Solving the Curses of Dimensionality, Wiley-Interscience, 2007.

[154] A. Simundic, “Measures of diagnostic accuracy: basic definitions,” *Journal of Medical and Biological Sciences ,* vol. 22, no. 4, pp. 61-5, 2008.

[155] M. H. Ebell, “Free Online Course in Evidence-Based Practice,” Institute for Evidence-based Health Professions Education, [Online]. Available: http://ebp.uga.edu/courses/Chapter%204%20-%20Diagnosis%20I/6%20-%20Likelihood%20ratios.html. [Accessed 12 12 2016].

[156] P. McNicholas, T. Murphy and M. O'Regan, “Standardising the lift of an association rules.,” *Computational Statistics and Data Analysis,* vol. 52, no. 10, pp. 4712-4721, 2008.

[157] K. Leung, “Association rules,” Polytechnic University, 7 December 2007. [Online]. Available: http://cis.poly.edu/~mleung/FRE7851/f07/AssociationRules3.pdf. [Accessed 12 12 2016].

[158] S. Plansangket and J. Q. Gan, “A new term weighting scheme based on CSDF for document representation and classification,” in *The 7th Computer Science and Electronic Engineering Conference (CEEC)*, Essex, UK, 2015.

[159] Max-Planck-Institute, “YAGO2s: A High-Quality Knowledge Base,” Max Planck Institute for Informatics and the Telecom ParisTech University, 2013. [Online]. Available: http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/. [Accessed 11 April 2013].

[160] T. Lippincott and R. Passonneau, “Semantic clustering for a functional text classification task,” in *International Conference on Intelligent Text Processing and Computational Linguistic*, Mexico City, Mexico, 2009, pp. 509-522.

[161] M. Stede, “The hyperonym problem revisited: Conceptual and lexical hierarchies in language generation,” in *Proceedings of The First International Conference on Natural Language Generation*, Mitzpe Ramon, Israel, 2000, pp. 93-99.

[162] N. Project, “WordNet Interface,” NLTK Project, [Online]. Available: http://www.nltk.org/howto/wordnet.html. [Accessed 13 August 2015].

[163] M.-W. Chang, “Importance of semantic representation: dataless classification,” *AAAI,* pp. 830-835, 2008.

[164] D. D. Lewis, “Reuters21578,” [Online]. Available: http://www.daviddlewis.com/resources/testcollections/reuters21578/. [Accessed 2014 April 4].

[165] J. Rennie, “Jason Rennie's Home Page,” [Online]. Available: http://qwone.com/ jason/20Newgroups. [Accessed 26 November 2015].

[166] D. Martens and F. Provost, “Explaining data-driven document classifications,” *MIS Quarterly,* vol. 38, no. 1, pp. 73-99, 2014.

[167] M. l. g. a. t. U. o. Waikato, “Weka 3 Data mining software in Java,” The University of Waikato, [Online]. Available: http://www.cs.waikato.ac.nz/ml/weka/. [Accessed 21 December 2016].

[168] S. Bordag, “A comparison of co-occurrence and similarity measures as simulations of context,” in *The 9th International Conference on Computational Linguistics and Intelligent Text Processing*, Haifa, Israel, 2008, pp. 52-63.

[169] Y. C. Chang, “A new query reweighting method for document retrieval based on genetic algorithm,” *IEEE Transaction on Evolutionary Computation,* vol. 10, no. 5, pp. 617-622, 2006.

[170] S. Danso, E. Atwell and O. Johnson, “A comparative study of machine learning methods for verbal autopsy text classification,” *International Journal of Computer Science Issues,* vol. 10, no. 6, 2013.

[171] B. Pan, “The power of search engine ranking for tourist destinations,” *Tourism Management,* vol. 47, pp. 79-87, 2015.

[172] S. Balakrishnama and A. Ganapathiraju, “Linear discriminant analysis - a brief tutorial,” in *International Symposium on Information Processing*, 1998.

[173] R. A. Fisher, “The user of multiple measurements in taxonomic problems,” *Annals of eugenics,* vol. 7, no. 2, pp. 179-188, 1936.

[174] S. Balakrishnama and A. Ganapathiraju, “Linear discriminant analysis-a brief tutorial,” in *International Symposium on Information Processing*, 1998.

[175] S. Sayad, Real Time Data Mining, Self-Help Publishers, 2011.

[176] A. M. Martinez and A. Kak, “PCA versus LDA,” *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 23, no. 2, pp. 228-233, 2001.

[177] R. A. Fisher, “The use of multiple measurements in taxonomic problems,” *Annals of Eugenics,* vol. 7, no. 2, pp. 179-188, 1936.

[178] F. Wilcoxon, “Individual comparisons by ranking methods,” *Biometrics Bulletin,* vol. 1, no. 6, pp. 80-83, 1945.

[179] M. P. Fay and M. A. Proschan, “Wilcoxon-Mann-Whitney or t-test? On assumptions for hypothesis tests and multiple interpretations of decision rules,” *Statistics Surveys,* vol. 4, pp. 1-39, 2010.

[180] C. Carpineto and G. Romano, “A survey of automatic query expansion in information retrieval,” *ACM Computing Surveys,* vol. 44, no. 1, pp. 1-50, 2012.

[181] X. Niu and D. Kelly, “The use of query suggestion during information search,” *Information Processing and Management,* vol. 50, no. 1, pp. 218-234, 2014.

[182] W. D. Blizard, “Multiset theory,” *Notre Dame Journal of Formal Logic,* vol. 30, no. 1, pp. 36-66, 1989.

[183] S. Kulkarni and D. Caragea, “Computation of the semantic relatedness between words uing concept clouds,” in *International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (KDIR)*, Madeira, Portugal, 2009, pp. 183-188.

[184] S. Plansangket and J. Q. Gan, “A query suggestion method combining TF-IDF and Jaccard coefficient for interactive web search,” *Artificial Intelligence Research,* vol. 4, no. 2, p. 119, 2015.

[185] S. Plansangket and J. Q. Gan, “Performance evaluation of state-of-the-art ranked retrieval methods,” in *International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (KDIR)*, Rome, Italy, 2014.

[186] R. M. Conroy, “What hypotheses do "nonparametric" two-group tests actually test?,” *The Stata Journal,* vol. 12, no. 2, pp. 182-190, 2012.

Appendix A

List of Publications

A.1 Journal Papers

Suthira Plansangket, and John Q. Gan. "A query suggestion method combining TF-IDF and Jaccard Coefficient for interactive web search," *Artificial Intelligence Research*, vol. 4, no. 2, pp.119-125, 2015

Suthira Plansangket, and John Q. Gan. "Re-ranking google search returned web documents using document classification scores," *Artificial Intelligence Research*, vol. 6, no. 1, pp.59-68, 2017

A.2 Conference Papers

Suthira Plansangket, and John Q. Gan. “Performance evaluation of state-of-the-art ranked retrieval methods and their combinations for query suggestion,” in *The 6th International Conference on Knowledge Discovery and Information Retrieval*, Rome, Italy, 2014, pp.141-148.

Suthira Plansangket, and John Q. Gan. “A new term weighing scheme based on class specific document frequency for document representation and classification,” in *The 7th Computer Science and Electronic Engineering Conference (CEEC)*, Essex, UK, 2015, pp. 5-8.