The Hong Kong Polytechnic University
Department of Computing

New Document-context Term Weights And Clustering For Information Retrieval

Edward Kai Fung Dang

A Thesis submitted for the partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

March 2010

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

_____(Signed)

Edward Kai Fung Dang

Abstract

In this thesis we investigate new methods to deal with the polysemy and word mismatch problems in information retrieval (IR).

We tackle polysemy by using 'document-contexts', which are text windows centred on query terms in a document. Analysis of the words in the vicinity of a query term can identify its specific meaning in the context. In IR, many of the commonly used term weights are variants of the TF-IDF form. The tradition TF-IDF weight of a term depends only on the occurrence statistics of the term itself. We have studied a novel 'context-dependent' term weight, which incorporates information based on the words found in the document-contexts of a term. These term weights are generated by a Boost and Discount (B&D) procedure, which utilizes any relevance information that is available to estimate the probability of relevance of a context. Such relevance information may come from actual relevance judgments that a user makes on a (small) number of documents, as in 'relevance feedback' (RF). The theoretical justification of our scheme to calculate the new term weights is provided by a probabilistic non-relevance decision model of IR. We present experiments in the RF setting to test the context-dependent term weights. We demonstrate that using the new term weights can yield statistically significant improvement in retrieval compared with the traditional weights.

Regarding the word mismatch problem, one plausible solution is to use clustering techniques. A traditional clustering evaluation measure used in IR is the MK1, which is a score calculated for the single 'optimal cluster' that can be extracted from the clustering result. MK1 is appropriate if a single retrieved cluster is desired. However, in some applications it may be desirable for the retrieval results to be presented in multiple clusters according to sub-topics. For this case, we introduce a new evaluation measure, called CS, which corresponds to finding an optimal combination of clusters. We define a sub-class of CS, called CS1, applicable when the clusters are disjoint. By reformulating the optimization to a 0-1 linear fractional programming problem, we

demonstrate that an exact solution of CS1 can be obtained by a linear time algorithm. We discuss how our approach can be generalized to overlapping clusters, and present greedy algorithms to obtain optimal estimates. We claim that one particular 'cost effectiveness' algorithm yields the global optimal solution for clusters that overlap only by nesting. A mathematical proof of this claim by induction is presented.

We have also investigated whether clustering techniques can further improve the retrieval effectiveness in relevance feedback using context-dependent term weights. B&D utilizes information extracted from the judged documents to provide evidence of relevance or non-relevance in the unseen documents. We use clustering to seek contexts from unseen documents that are similar to those in the judged documents. In this way, additional relevance information can be obtained for B&D. Experiments on the TREC-2005 collection show that a 'clustered SVM' scheme is effective in further improving relevance feedback effectiveness as compared to standard B&D, yielding small but statistically significant improvements in MAP. Thus, this is a promising direction for further research.

List of Publications

DANG, E. K. F., LUK, R. W. P., LEE, D. L., HO, K. S., AND CHAN, S. C. F. Query-specific clustering of search results based on document-context similarity scores. In *Proceedings of the 15th Intl. Conf. on Information and Knowledge Management (CIKM'06)* (2006), P. S. Yu, V. Tsotras, E. Fox, and B. Liu, Eds., pp. 886–887.

DANG, E. K. F., LUK, R. W. P., HO, K. S., CHAN, S. C. F., AND LEE, D. L. A New Measure of Clustering Effectiveness: Algorithms and Experimental Studies. *Journal of the American Society for Information Science and Technology 3*, 59 (2008), 390–406.

DANG, E. K. F., LUK, R. W. P., LEE, D. L., HO, K. S., AND CHAN, S. C. F. Optimal Combination of Nested Clusters by a Greedy Approximation Algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence 31*, 11 (2009), 2083–2087.

DANG, E. K. F., WU, H. C., LUK, R. W. P., AND WONG, K. F. Building a framework for the probability ranking principle by a family of expected weighted rank. *ACM Transactions on Information Systems 27*, 4, Article 20 (2009), 1–37.

DANG, E. K. F., LUK, R. W. P., ALLAN, J., HO, K. S., CHAN, S. C. F., CHUNG, K. F. L., AND LEE, D. L. A New Context-Dependent Term Weight Computed by Boost and Discount Using Relevance Information. To appear in *Journal of the American Society of Information Science and Technology.*

Acknowledgements

First, I wish to thank my supervisor Dr. Robert W.P. Luk for introducing me to the field of Information Retrieval and for giving me the opportunity to participate in IR research. I am indebted to him for his guidance and support throughout my study. I have benefited greatly from Robert's 'hands-on' supervision — from the suggestions of insightful ideas, to his help with debugging my programs, as well as the essential maintenance of the Matrix. His insistence in striving for excellence is one of the attitudes that I learn from him.

I am grateful to Dr. Stephen Chan for letting me into computing research in the first place while I was an 'outsider' in the field. I thank my co-supervisors Dr. Edward K.S. Ho and Prof. Dik Lun Lee for their support in their respective ways. In particularly, Prof. Lee's many helpful comments to my research paper manuscripts are much appreciated.

I also thank the past and present members of the Information Retrieval group whose period of study has overlapped with mine – Karen W.S. Wong, Li Yinghao, Jack Wu Ho Chung and Wang Dayu. Special mention goes to Jack who always provides quick and helpful answers to my many technical questions.

I am grateful to various people in my church for their support and encouragement, including Grace Wong and others in Jonathan Fellowship, as well as Rev. Sung Po King and members of the 2009/2010 'Gold' Disciple class.

Last, but most of all, I thank my parents and my wife Fiona, who is the person I am most thankful to God for placing in my life – for their constant love, encouragement, understanding, prayers and belief in me.

Contents

Abstract **i**

List of Publications **iii**

Acknowledgements **iv**

List of Figures **viii**

List of Tables **x**

Notations **xii**

1 Introduction **1**

1.1 Research problems and motivation 4

1.1.1 Context-dependent term-weighting 4

1.1.2 Clustering evaluation measure 6

1.1.3 Enhancing retrieval effectiveness by clustering techniques . . 7

1.2 Contributions and their Significance 7

1.3 Experimental environment 9

1.3.1 Test collections 9

1.3.2 Retrieval evaluation measures 11

1.4 Outline 12

2 Literature Review and Background **14**

2.1 The polysemy and word mismatch problems in IR 15

2.2 Relevance Feedback 18
2.3 Term Weights 21
2.3.1 The TF-IDF weighting 22
2.3.2 Probabilistic retrieval model and the BM25 weighting 25
2.3.3 Context-dependent term weights 27
2.4 Clustering 27
2.4.1 Clustering algorithm 28
2.4.2 Similarity score 31
2.4.3 Clustering in IR 32
2.4.4 Clustering evaluation 37

3 Context-Dependent Term Weights **40**
3.1 Document-context based probabilistic non-relevance decision model . 43
3.2 Computing context-dependent term weights by Boost and Discount . 47
3.2.1 Modeling the probability of relevance of a context 48
3.2.2 Calculating the context-dependent term weights 54
3.3 Experiments 59
3.3.1 Experimental environment and setup 60
3.3.2 Calibration of model parameters 62
3.3.3 Comparison of RF performance using context-dependent term weights vs baseline 76

4 Clustering Evaluation **81**
4.1 Clustering effectiveness measure based on a combination of subclusters 81
4.1.1 Micro-average F-measure 83
4.2 Optimal combination of disjoint clusters – CS1 88
4.2.1 Reformulation of the optimization problem 90
4.2.2 Experiments on CS1 92
4.3 Optimal combination of overlapping clusters – CS2 103
4.3.1 Experiments – overlapping clusters 109
4.4 The MMF Problem and Optimality of GAA for Nested Clusters . . . 112

4.4.1 MMF problem and related work 112
4.4.2 Worst case time-space complexity of GAA 122

5 Clustering for Relevance Feedback **124**
5.1 Clustering approach 126
5.2 Experimental settings 128
5.3 Experimental results 131

6 Conclusion and Future Work **143**
6.1 Summary and contributions 144
6.2 Future research 147

Bibliography **149**

List of Figures

2.1 Distribution of similarity scores between Relevant-Relevant and Relevant-Irrelevant pairs. 33

3.1 The logistic function, $f(z) = 1/(1 + e^{-\gamma z})$. 49

3.2 Flow diagram of relevance feedback with standard query expansion or with B& D. The parameters used in the various steps are indicated in curly brackets. 58

3.3 Calibration of Query Expansion via RF parameters, based on averaging over 50 title queries of TREC-2005. 69

3.4 Difference between residue MAP values obtained for QE with $\beta_{QE} = 1.0$ and $\beta_{QE} = 0.8$, averaged over queries with the same value of N_R@20 for TREC-2005, plotted against N_R@20. 71

3.5 Residue MAP averaged over 50 queries of TREC-2005, obtained by setting $\beta_{QE} = 1.0$ for queries with N_R@20 $\leq N_{split}$ and setting $\beta_{QE} = 0.8$ for queries with N_R@20 $> N_{split}$. 72

3.6 Calibration of B&D parameters, for TREC-2005 queries with N_R@20 ≤ 3. 74

3.7 Calibration of B&D parameters, for TREC-2005 queries with N_R@20 > 3. 75

3.8 Difference of the residue MAP values obtained by B&D and the QE baseline (with Split) for the 50 title queries of TREC 2005. The queries are sorted in increasing order of N_R@20, which is indicated by the X-axis. 77

4.1 Dendrogram representing group-average clustering of the top 40 retrieved documents of TREC-7, query 351 (Falkland petroleum exploration). 83
4.2 Algorithm to calculate MK1 87
4.3 Illustration of clusters used in (a) Algorithm 1 and (b) Algorithms 2A, 2B, 2C as applied to the same hierarchical system. Each leaf node is a document, and relevant documents are denoted by 'r'. 88
4.4 Algorithm to calculate CS1 89
4.5 Plots of CS1 vs. MK1 for top 1000 retrieved documents for the queries of TREC-7, with $\beta = 1.0$. 99
4.6 Plots of the average values of CS1 and MK1 vs β, for the top 1000 retrieved documents for the queries of TREC-7, corresponding to three types of hierarchical clustering algorithms as indicated 102
4.7 Greedy approximation algorithms to estimate optimal effectiveness measure for overlapping clusters 108
4.8 Greedy approximation algorithm (GAA) for MMF problem 115

5.1 Flow of our general approach of applying clustering in RF 131
5.2 Difference of the residue MAP values obtained by various Method C conditions and standard B&D, for the 50 title queries of TREC 2005. The queries are sorted in increasing order of N_R@20, which is indicated by the X-axis. 140

List of Tables

1.1 Some statistics of the TREC collections used in our experiments. . . . 10

3.1 List of parameters used in pseudo-relevance feedback (PRF), query expansion and 'Boost & Discount' 62

3.2 Summary of parameters used for pseudo relevance feedback (PRF). . 65

3.3 Summary of the parameters calibrated to obtain the best residue MAP for TREC-2005, using the baseline or B&D models 73

3.4 Summary of the residue MAP values obtained by the baseline and B&D term weights in RF with 20 or 10 relevance judgments, for various TREC collections 78

4.1 Evaluation measures for $\beta = 1.0$, averaged over 50 queries for each collection. 98

4.2 Averaged evaluation measures for $\beta = 1.0$. Retrieved documents that do not have relevance judgment information are discarded. 100

4.3 Evaluation measures for different values of β, averaged over 50 queries. 101

4.4 Statistics of the optimal clusters for TREC-7. 102

4.5 Estimates of optimal combination of bottom level clusters, using several greedy algorithms (Algorithms 2A, 2B and 2C), averaged over the queries of each TREC collection 109

5.1 RF (N_{RF}=20) performance with various settings of Method A, averaged over 50 queries of TREC-2005. 135

5.2 RF (N_{RF}=20) performance of Method B, averaged over 50 queries of TREC-2005 136
5.3 RF (N_{RF}=20) performance with various settings of Method C, averaged over 50 queries of TREC-2005 141
5.4 MAP values obtained in RF (N_{RF}=20) with Method C3, averaged over 50 queries of various TREC collections 141

Notations

Symbol	Description
q_i	i^{th} query term
$f(t,d)$	term frequency of term t in document d
$df(t)$	document frequency of term t in the corpus
$idf(t)$	inverse document frequency of term t in the corpus
$d[k]$	term at location k in document d
$c(d,k,m)$	document-context in document d, centred at location k, with a window size of m words
$Loc(t,d)$	Set of locations in document d where term t occurs
$V(\cdot)$	set of distinct terms (vocabulary) contained in the argument
$card(\cdot)$	cardinality of the argument
N_{PRF}	Number of top passages assumed to be relevant in PRF
N_{RF}	Number of relevance judgments made in RF
$N_R@n$	Number of relevant documents among the n judged documents
N_{QE}	Number of query expansion terms selected from judged relevant documents
$N_{QE,irr}$	Number of query expansion terms from judged irrelevant documents
α_{QE}	Weight of original query vector in the expanded query vector
β_{QE}	Weight of positive vs. negative components of the query-expansion vector
k	Scaling in the BM25 term-frequency factor
b	Slope in the BM25 term-frequency factor
N_{rerank}	Number of passages returned by an initial retrieval to be re-ranked in RF
C_B	Size of contexts in relevant documents for ”Boost” terms extraction
C_D	Size of contexts in irrelevant documents for ”Discount” terms extraction
C_m	Size of contexts in an unseen document for matching B&D terms
γ_B	Logistic coefficient for Boost terms
γ_D	Logistic coefficient for Discount terms
D	Multiplicative factor controlling the strength of B&D (Eq.3.11)

$S_B(q_i)$	Set of boost terms for query term q_i
$S_D(q_i)$	Set of discount terms for query term q_i
E	Cluster effectiveness measure (E-measure, Eq.2.14)
F	F-measure, equals to 1-E
β	parameter governing relative importance of recall and precision in E-measure
MK1	Optimal clustering effectiveness measure, equals thd best E-measure attainable by selecting a single cluster
CS	Optimal micro-average E-measure attainable by selecting a combination of subclusters
CS1	A subclass of CS applicable to a family of disjoint clusters
CS2	A subclass of CS applicable to a family of clusters that overlap only by nesting

Chapter 1

Introduction

The popularization of the personal computer in the 1980s and the advent of the World Wide Web in the 1990s truly heralded the Era of Information. Nowadays people are in constant need of information, whether to accomplish a task demanded by their job, to fulfill their interests, or even to meet the requirements of everyday life, such as finding about available choices of household consumables. With more and more information being stored electronically, the electronic computer itself provides people with a powerful tool to search for information that they need. To harness this power, it is necessary to develop methodologies to find relevant information both effectively and efficiently. This is the aim of Information Retrieval (IR), which has been an active area of research since the 1950s. IR research is multidisciplinary, encompassing fields such as computer science, mathematics and linguistics. An important pioneering implementation of an IR system was the SMART system developed by Salton and his co-workers [71] in the 1960s, first at Harvard University and later at Cornell University. Such IR systems allowed early IR experiments to be performed. By the 1970s, a

number of retrieval methods were already developed using small corpora [73].

While IR research has been ongoing for half a century, even state-of-the art retrieval systems today are still far from achieving perfect effectiveness. With the constant exponential growth of the available information, there is for one the question whether techniques developed for small corpora can efficiently scale up to the huge corpora which are the norm today. Furthermore, with more and more available data to search from, it is conceivable that it is getting harder to distinguish between what is truly relevant and what is not. Besides, there are intrinsic lexical problems associated with the way people express their information need. A person's information need is generally stated in the form of a 'query' which consists of one or more 'query terms'. There is firstly the problem of polysemy, i.e. the existence of multiple meanings of a word. An example is the query term 'blackberry', whereby the term itself does not indicate whether the person desires information about blackberry the fruit or the mobile device. Polysemy is a cause for query term ambiguity (e.g Spärck Jones et al. [77]). Because of this problem, the IR system may return many irrelevant documents related to a topic not desired by the user, resulting in poor *precision*, which is equal to the percentage of a retrieved list of documents that are relevant. Precision is an important evaluation measure in IR. Another lexical problem is word mismatch (e.g. Xu and Croft [93]), wherein the same concept may be referred to by different words, i.e. synonyms. For example, if 'automobile' is one of the query terms, there may be relevant documents that do not contain the word 'automobile' at all, but the word 'car' instead. In this case, the IR system may fail to find some relevant documents if it only returns documents containing the exact query terms, resulting in poor *recall*, which is equal to the

percentage of all relevant documents in the corpus that are retrieved. Recall is another important evaluation measure in IR.

This thesis investigates new methods that deal with the polysemy and word-mismatch problems mentioned above. Past research has found query expansion via relevance feedback (RF) to be an effective way to tackle these lexical problems [65],[26]. Hence in our work, we seek methods that improve the performance of standard query expansion methods in the RF setting. Another way to deal with the polysemy problem, was the use of 'document-contexts' (or simply 'context' for brevity) by Wu et al. [92],[91], within a probabilistic retrieval model. The document-context of a term t is defined as a text window of a fixed-size (i.e. fixed number of words) centred on t [92]. The idea is that the words in the neighbourhood of a term that occurs in a document may indicate the specific usage of the term. For example, if there are words such as 'fruit', 'nutrition' or 'vitamin' in the vicinity of the word 'blackberry', then one may infer the document is about the fruit rather than the device. Following Wu et al., the new methods that we introduce in this thesis are based on document-contexts.

In regard to the word mismatch problem, one plausible solution is the use of clustering techniques, which automatically classifies objects into groups according to their similarity. According to the Cluster Hypothesis of Jardine and van Rijsbergen [35], documents relevant to the same query tend to be similar to one another. Suppose a document A contains a synonym of an actual query term instead of the term itself (e.g. 'car' instead of 'automobile'). If the text of A is quite similar to a known relevant document, then it is a good indication that A is relevant as well. Forming clusters of documents will group the document A with other relevant documents, enabling it to

be identified as relevant event though it does not contain a query term. The use of clustering in IR has been investigated by various researchers (e.g., Salton [71], Jardine and van Rijsbergen [35]). A review of this topic will be included in Chapter 2.

1.1 Research problems and motivation

In this section we describe the main research problems that are investigated in this thesis and the motivation for studying these problems.

1.1.1 Context-dependent term-weighting

In information retrieval and text data mining, an important element is the weighting of terms. A set of weights for all the terms that occur in a document constitutes a representation of the document. It is well established in IR that the retrieval effectiveness depends on appropriate term-weighting (Salton and Buckley [68]). A well-known and common term weight in use is the TF-IDF (e.g. Robertson and Sparck-Jones [62]). For example, the successful BM25 term weights introduced in the Okapi system (Robertson et al. [63]) are essentially TF-IDF weights. In general, the TF-IDF value of a term t in a document d depends on the occurrence statistics of t in d or in the corpus, but does not depend on the other terms appearing in the document. In other words, TF-IDF is independent of the 'document-contexts' of t, using the definition of a document-context as a fixed-size text window centred on t [92]. In this thesis, we investigate the new class of context-dependent term weights, and address the following research problem: Whether context-dependent term weights can improve the retrieval performance in RF

compared to the traditional TF-IDF term weights?

Our study is motivated by the recent work of Wu et al. [91] who showed that assigning TF-IDF weights to terms in a document can be interpreted as making relevance decisions in information retrieval. They introduced a probabilistic nonrelevance decision model in IR. Their model mimics a human making a series of 'local relevance decisions' by reading texts in the vicinity of all the query terms that occur in a document and deciding whether these portions of the document are individually relevant to the query. Thus the model is based on document-contexts centred on query terms. They derived a ranking formula that has a form similar to the BM25 weight (e.g. Robertson and Walker [59]), provided that the term frequency weighting of the query terms are adjusted according to the local evidence of relevance extracted from the text windows. Their derivation involves making several assumptions, in particular the Minimal Context assumption, which states that for any query, the local relevance at a location k in a document d is determined only by the single term occurring at location k. As pointed out by Wu et al. [91], this assumption is not realistic because it is expected that the words occurring close to a query term, and not the query term alone, should affect the local relevance decision. They commented that such an unrealistic assumption may cause performance limitations of TF-IDF term weights. Hence, we are motivated to relax this assumption and investigate whether there is any advantage in term weights that depend on contexts with a size larger than unity. Furthermore, because of the common use of TF-IDF, it is of interest to test how effective are the context-dependent term weights as compared with the traditional TF-IDF.

1.1.2 Clustering evaluation measure

We investigate the use of clustering techniques as a solution to the word mismatch problem. Numerous clustering algorithms have been developed in the past and applications of them are found in a broad range of disciplines. Therefore, we need to find an effective clustering algorithm for our purpose. An important issue that arises is how to define an appropriate measure to quantitatively evaluate the clustering results. This is the problem that we first address.

In order to quantify the quality of a document cluster, Jardine and van Rijsbergen [35] introduced the E-measure, which is a composite measure that combines the precision and recall values of the cluster. They also defined the MK1 measure which is equal to the best E-measure attainable by retrieving a single cluster based on the clustering of a set of documents. This is a natural benchmark measure appropriate to applications where a single retrieved cluster is desired. However, it is possible that better retrieval effectiveness is attained by returning several clusters rather than a single cluster (Griffiths et al. [22]). For example if a search query is too general, it may cover several sub-topics, and documents relevant to the query may fall into different sub-topic categories (Chik et al. [8]). Conventional similarity scores based on word statistics do not necessarily yield a high value across different relevant sub-topics. In this case, clustering algorithms may produce isolated clusters where relevant documents are concentrated. Correspondingly, an appropriate retrieval strategy would be to identify the multiple 'high precision' clusters and return all the documents contained in them as a pool. The MK1 measure, which is associated with a single optimal cluster,

would not be an appropriate benchmark for this strategy. In fact, there are applications where multiple clusters are naturally desired. For example, the web search engine Vivisimo (Koshman, Spink and Jansen [43]) returns search results in clusters corresponding to different sub-topics, and a user may find relevant information in more than one of these clusters. We are therefore motivated to seek a new measure which will more truly reflect the effectiveness of a clustering algorithm for applications where multiple clusters are desired.

1.1.3 Enhancing retrieval effectiveness by clustering techniques

Our experiments have demonstrated the effectiveness of the new context-dependent term weights in a RF task being demonstrated by our experiments. In relevance feedback, the new term weights are obtained by utilizing relevance information being extracted from a small number of judged documents. We are motivated to investigate whether further performance improvement can be obtained by discovering additional relevance information via clustering techniques.

1.2 Contributions and their Significance

In this section we briefly state the main contributions of our work and their significance.

1. Improvement in retrieval effectiveness with new context-dependent term weights

We have investigated novel context-dependent term weights in a relevance feedback (RF) task. These weights are computed by a Boost and Discount (B&D) procedure.

As such, this work represents the first experimental instantiation of context-dependent term weights that are used for retrieval. These new term weights are shown to be effective in enhancing the performance of RF, compared to using the traditional BM25 weights which are context-independent. Apart from RF, the new term weights may also be used in other applications such as text categorization (e.g. Sebastiani [72] and Yang [95]). Because our B&D procedure generates the new term weights by calculating shifts to the widely used BM25 term weights, the method can readily be implemented in systems that use the BM25 weights.

2. New clustering evaluation measure

We have introduced a new clustering evaluation measure, called CS (which stands for 'Combination of Subclusters'), based on seeking an optimal combination of clusters rather a single optimal cluster as in the traditional MK1 measure. As such, CS is an extension of the MK1 measure. We show that calculating this measure can be reformulated as an 0-1 optimization problem. For cases when clusters are disjoint, the problem becomes a linear fractional 0-1 optimization problem which can be solved by linear time algorithms. However, for arbitrarily overlapping clusters, the optimization problem is NP-hard. In this case, we have shown how the optimal solution can be estimated by greed algorithms. While the greedy algorithms can be applied to arbitrarily overlapping clusters, we present a mathematical proof that one particular algorithm, based on 'cost effectiveness', yields the global optimal solution for clusters that overlap only by nesting.

Clustering has applications in many disciplines apart from IR, including pattern

recognition (e.g. Jain [34]), data mining (e.g. Judd et al. [38]) and machine learning (e.g. Carpineto et al. [6]). Our new measure for clustering effectiveness could be useful in future developments of cluster analysis in IR and other disciplines. Commercial applications of clustering for retrieval, such as Vivisimo, suggest that there is interest for further research in this area. In our experiments, the better evaluation measures obtained by combining clusters as compared with the traditional MK1 measure also reveals a greater but latent potential of the clustering algorithms in grouping relevant documents together. While we have focused on hierarchical clustering in our study, the concept can readily be applied to any clustering algorithm, such as K-means.

1.3 Experimental environment

In this section, we briefly describe the test collections that we use in our experiments, and also the retrieval evaluation measures that we use.

1.3.1 Test collections

Many of the early experiments in IR were conducted using small text corpora such as the Cranford collection with several thousand documents [73]. In 1992, the Text REtrieval Conference (TREC) was started (Harman [27]). TREC is an ongoing series of IR workshop which provides a platform for developing IR techniques by making available large text collections. It also supplies sets of queries which come with relevance judgements made by human experts on a pool of retrieved documents for each query. By using TREC's common evaluation package, it is possible for different TREC

participants to compare the effectiveness of their different techniques.

Over the years, the size of the TREC test collections has expanded. In our work, we have performed our experiments using various TREC collections, including TREC-2, -6, -7, -8 and -2005. Some statistics of the collections are given in Table 1.1. Each of these collections has 50 'topics', consisting of 'title', 'description' and 'narrative' fields, which describe the topics in varying details. In our experiments, we use 'title queries' extracted from the title fields because these have an average of between two and three query terms, similar to typical web queries [79]. The TREC-2005 collection has a much large size compared with previous collections (Table 1.1) and therefore more in-line with current and future web search applications. However, in order to demonstrate the robustness of our retrieval methods across collections, our experiments are also performed using the other collections (i.e. TREC-6, 7 and 8). The reason for choosing these collections is because of their different characteristics. While these earlier collections have similar sizes, TREC-6 contains some very long documents, such as congressional records. TREC-7 and TREC-8 use the same document collections without congressional records, but TREC-7 contains more difficult queries for which it is hard to achieve good retrieval results [89].

Table 1.1: Some statistics of the TREC collections used in our experiments.

	TREC-2	TREC-6	TREC-7	TREC-8	TREC-2005
Av. # of title query terms	3.8	2.5	2.4	2.4	2.6
Av. # of rel docs per query	232.56	92.22	93.48	94.56	131.22
Number of documents	741,857	556,077	528,155	528,155	1,033,461

1.3.2 Retrieval evaluation measures

As mentioned earlier, *precision* and *recall* are two important evaluation measures in IR. While *precision* indicates the accuracy of the retrieval result, *recall* indicates its completeness. These two measures do not take into account the ordering of the documents in the retrieval result. For systems that return a ranked list of documents, it is useful to have a measure which emphasizes placing relevant documents higher in the list. A measure that serves this purpose is the Average Precision (AP), which is defined by:

$$\mathrm{AP} = \frac{1}{R} \sum_{r=1}^{N} P(r) \times rel(r), \tag{1.1}$$

where r is the rank, N is the number of the documents in the ranked list, R is the number of relevant documents in the corpus, $rel(r)$ is a binary function indicating the relevance of the document at rank r (i.e. $rel(r)$ is 1 or 0 if the document at rank r is relevant or irrelevant respectively), and $P(r)$ is the precision of the list at the cut-off rank r. Eq.1.1 shows that the AP measure encompasses both precision and recall. It is typical that the value $N = 1000$ (Eq.1.1) is used in evaluations. For a given set of queries, averaging the AP values for all the queries yields the MAP (Mean Average Precision) measure. MAP is now the predominant evaluation measure in the IR literature, such as used in TREC. It is clear that MAP, as for precision and recall, ranges between 0 and 1, with a value of 1.0 indicating perfect retrieval.

In some applications, a high recall value may be important, such as in the searching of patent or legal documents. However, some users may desire a small number of truly relevant documents rather than getting the complete set of relevant documents.

In this case, a common appropriate measure is P@n, i.e. the precision value of the top n retrieved documents, with P@10 typically being used. However, Buckley and Voorhees [3] had found P@10 to have a sensitivity issue, so that a large number of queries (more than 50) may be required to distinguish the performance effectiveness of two different methods. Hence, in this thesis we will not report evaluations with P@10, but focus on the MAP measure only.

In Chapter 3 and Chapter 5, our experiments are performed in a RF setting, in which relevance judgments are made on N_{RF} documents, and the information fed back to the system for a second retrieval. One established methodology to evaluate the RF performance is based on the *residue collection*, from which the N_{RF} judged documents are removed (e.g. Ruthven et al. [66]). The evaluation measures, such as MAP, are calculated based on the remaining relevant documents in the residue collection. We also adopt this practice. Thus, our MAP values reported in this thesis are residue measures.

1.4 Outline

The remainder of this thesis is as follows.

Chapter 2 Literature review and background: In this chapter we first review some methods that have been used in past research to tackle the polysemy and word-mismatch problems. In particularly, several topics involved with our new methods are reviewed in greater detail, including (1) relevance feedback; (2) term weighting; and (3) application of clustering techniques in IR.

Chapter 3 Context-dependent term weights: We describe context-dependent term weights that are computed by our Boost and Discount (B&D) procedure in a relevance feedback (RF) setting. Extensive experimental results are presented to demonstrate that the new term weights can produce enhanced retrieval effectiveness compared with the baseline which uses context-independent BM25 weights.

Chapter 4 Clustering evaluation: We introduce the new clustering evaluation measure CS. Experiments are presented to demonstrate the subclass CS1, which applies to non-overlapping (i.e. disjoint) clusters. For the case of overlapping clusters, we also show how the estimates of the optimal measure may be obtained by greedy approximation algorithms. For the case where clusters overlap only by nesting, we present a proof that the 'cost effectiveness' greedy algorithm in fact yields the global optimal measure.

Chapter 5 Relevance feedback with document-context clustering: We describe a method to apply clustering techniques as an extension to the B&D procedure to generate context-dependent term weights. Experimental results show that this is a promising direction for further research.

Chapter 6 Conclusion and future work: The main results and contributions of the thesis are summarized. Some items for possible future work are proposed.

Chapter 2

Literature Review and Background

This thesis tackles the polysemy and word mismatch problems in information retrieval (IR). In this chapter we first discuss these problems and review the various methods that have been used in past research to solve them (Section 2.1). In particularly, among the successful solutions to both the polysemy and word mismatch problems is 'query expansion' via *relevance feedback* (RF). Building on past research, we seek new solutions to further improve the performance of query expansion in relevance feedback (Chapter 3). In Section 2.2, we will present a review of RF in detail. Furthermore, our new method presented in Chapter 3 is based on the novel *context-dependent term weights*. Hence, Section 2.3 is included to discuss 'Term weighting', which is a crucial element not only in IR, but also in various text mining tasks, such as text categorization. As for the word mismatch problem, we investigate a new clustering method (Chapter 5) to tackle it. In Section 2.4, cluster analysis is reviewed, including common clustering algorithms, the use of clustering methods in IR, and cluster evaluation measures.

2.1 The polysemy and word mismatch problems in IR

Even at quite early stages of IR research, it was realized that a factor which limits retrieval effectiveness is actually related to the difficulty for a person to formulate search requests. These requests, or queries, generally consist of one or more query terms. Intrinsic lexical problems with the query terms arise which affect retrieval effectiveness. Two of the problems that we address in this thesis are *polysemy* and *word mismatch*.

Polysemy refers to the existence of multiple meanings of a word (e.g. [79]). The problem caused by polysemy in retrieval is that a query term by itself may not indicate which specific meaning of the word is intended. There may be many documents containing the query term, albeit with a meaning different from the user's intension and hence not of interest to the user.

Some solutions to the polysemy problem that have been studied in the past include:

1. Query modification Various ways of query modification (or *query expansion*) have been studied. Some of these methods are fully automatic, without any need of user interaction. Others require some input from users. Salton and Lesk [69] tested the effect of automatic query expansion by either broader or narrower terms selected from a hierarchical thesaurus. They found the effect to be inconsistent and hence the method was not generally useful [69]. Voorhees [86] implemented query expansion based on lexical-semantic relations encoded in WordNet [55], a large and general-purpose lexical system built at Princeton University. Their experiments performed on TREC collections found little benefit of the method both in the case of long queries or short queries. In fact, fully automated methods suffer the same polysemy problem, as

the added terms may also have meanings different from that intended.

A method of query expansion that requires user input is relevance feedback (RF). In RF, the retrieval system first returns a ranked list of documents based on the original search query. The list is presented to the user, who then reads a number of the top ranked documents and make relevance judgments on them. The judged relevant and irrelevant documents are then fed back to the retrieval system, which extract words from the judged documents to modify the search query. RF has been studied extensively in the past and found to be an effective scheme to enhance retrieval performance (e.g. Rocchio [65], Harman [26] and Buckley [2]).

2. Latent Semantic Indexing (LSI) LSI was introduced by Deerwester et al. to tackle the polysemy and word mismatch (or synonym) problems [18]. In LSI, the individual terms that describe a document are replaced by 'artificial concepts' that can be specified by one or combinations of several terms. While LSI has been tested and found to be effective in small text collections [18], its effectiveness does not seem to be scalable to larger collections, such as those used in TREC.

3. Document-contexts The document-context (or simply 'context') of a term t is defined as a text window of a fixed-size (i.e. fixed number of words) centred on t (Wu et al. [92]). It was shown in [92] and [90] that using document-contexts gave promising results in retrospective experiments (i.e. given full relevance information). Hence, it is worth studying whether document-context methods are effective in a predictive setting as well (i.e. given no or partial relevance information).

Overall, query expansion in a relevance feedback setting has shown to be a successful method to deal with the polysemy problem, while the document-context approach

is also a promising direction. Hence, in our work we attempt to further improve the retrieval performance by augmenting document-context methods to traditional query expansion in relevance feedback. A more detailed review of RF will be given in Section 2.2.

The second problem that we address is word mismatch, which arises because in writings, different people may use different words to express the same concept. In retrieval this may be problematic because a person's query terms may not match those of a relevant document. For example, instead of the query term 'automobile', a relevant document may use the word 'car'. Some solutions to the word mismatch problems include:

1. Query modification As for the polysemy problem, query expansion via relevance feedback is also an effective solution for word mismatch. By query expansion, extra query terms are selected from the judged relevant documents for a new retrieval. A relevant document that does not match the original query term may be identified if it contains the new terms in the expanded query.

2. Latent Semantic Indexing (LSI) The LSI method was introduced to tackle the word mismatch as well as polysemy problems (Deerwester et al. [18]). However, as mentioned above, the scalability of LSI methods to large text collections is questionable.

3. Clustering Clustering methods automatically classify objects into groups according to their similarity. By the *Cluster Hypothesis* (Jardine and van Rijsbergen [35]), relevant documents tend to be similar to one another. Clustering methods will thus group relevant documents together. This will enable a relevant document that

does not contain the exact query term to be identified, through its similarity with other relevant documents. A more detailed review of the use of clustering in IR is presented in Section 2.4.

Apart from the proven success of query expansion, the work of Tombros and van Rijsbergen [81] suggests that 'query-specific clustering' (see Section 2.4 below) is a promising direction. Hence, we have applied query-specific clustering to our new 'document-context' method . This will be reported in Chapter 5.

2.2 Relevance Feedback

In relevance feedback (RF), a user scans through a number of top ranked documents returned by a retrieval system and makes relevance judgments on them. Information extracted from the judged relevant or irrelevant documents is then fed back to the retrieval system to perform a second retrieval. Typically the relevance information is used to modify the original query, either by adding terms to the original query (i.e. query expansion), or to modify the weight of the query terms (i.e. query re-weighting). In principle, RF can be an iterative process, with the retrieval results based on the modified query being shown to the user, who again makes another relevance judgment on the top ranked documents in the new result. Query modification via RF has been found to be more effective in tackling the polysemy and word mismatch problems than fully automatic methods that rely on a thesaurus. The remaining of this section presents a survey of the past research in RF.

Query expansion

Rocchio [65] first formulated the query expansion (QE) method by means of relevance feedback, implemented in the Vector Space model (VSM). In VSM, both documents and queries are represented by n-dimensional vectors, where n is the number of distinct terms contained in the corpus. In Rocchio's formulation, terms belonging to the known relevant and irrelevant documents are added to the initial query vector $\vec{Q}$ with positive and negative weights respectively. Denoting the set of judged relevant and judged irrelevant documents by R and I respectively, Rocchio's formula of the modified query vector is:

$$\vec{Q}_{RF} = \vec{Q} + \frac{1}{card(R)} \sum_{\vec{D} \in R} \frac{\vec{D}}{|\vec{D}|_1} - \frac{1}{card(I)} \sum_{\vec{D} \in I} \frac{\vec{D}}{|\vec{D}|_1}, \tag{2.1}$$

where $|\vec{D}|_1$ is the city-block length of the document vector $\vec{D}$. If the weight of a query term drops below zero, it is removed from the query. Various modifications to the QE method of Rocchio have been studied, such as the early work of Ide [32]. While the works of Rocchio [65] and Ide [32] include all terms of the judged documents in expanding the query, Harman [26] showed that it was more effective to select terms from the judged documents for QE, according to an appropriate term ranking function. Query expansion in a probabilistic model was also studied by Robertson et al. [63] and Spärck-Jones et al. [78]. They used a term selection function called offer weight [78], which we will employ in our experiments as described in Section 3.3.2.

Pseudo-relevance feedback

One of the often cited problems of relevance feedback is actually the unwillingness of users to make relevance judgments in a real application (e.g. Ruthven et al. [66]). Hence, it is of interest to find ways to perform relevance judgments without direct user involvement. Among the various methods, pseudo-relevance feedback (PRF), or 'blind feedback' is a possible approach. PRF was first proposed by Croft and Harper [10] to estimate probabilities in a probabilistic model for an initial search. Subsequently, it is found to be effective for improving document rankings (e.g. Buckley et al. [1]). The assumption of PRF is that the top-ranked documents in the first retrieval are mostly relevant and contain useful terms that can help to discriminate relevant documents from irrelevant ones. However, one problem of PRF is the possibility of *query drift*, which occurs when the top ranked documents used for blind feedback actually contain few or no relevant documents. In this case, the terms added by PRF will be poor for detecting relevance, and hence degrade the retrieval performance.

Implicit feedback

Implicit feedback is another way to obtain relevance information without directly requesting a user to make relevance judgments. This approach is based on analysing click logs, especially in the web search setting. Some recent studies have shown that clickthrough data can be interpreted as implicit feedback (e.g. Joachims et al. [37]). Each 'click' on a link to a page is regarded as an endorsement (i.e. judged to be relevant), while negative inferences may be drawn from pages that are bypassed (e.g Das Sarma et al. [16]). Huge amounts of such data may be collected from web search

engines and thus available in bulk. Hence, by utilizing clickthrough data, relevance feedback may be a feasible technique in web search applications.

2.3 Term Weights

In Chapter 3, we describe our new method to solve the polysemy problem. The method involves calculating 'context-dependent term weights' of the query terms. This section provides the background knowledge of the important topic of 'term weighting'.

In IR and text mining, a document is represented by a set of weights which are assigned to all the terms that occur in the document. For example, in the simplest Boolean vector representation of a document, the presence or absence of a term is indicated by a unit or zero weight respectively. In general, the *term weight* indicates the importance of each term in the document.

In the various IR models, term weighting is an important component that enters the *ranking functions*, which assign scores to the retrieved documents in order to produce a ranked list. For example, consider the Vector Space Model (VSM) of IR (e.g. Salton et al. [70]). In this model, a document d is represented by the vector $\vec{D}$, whose elements are the term weights $w(t_i, d)$ assigned to each term t_i in document. Suppose a query q consists of terms $\{q_i\}$, so that its vector representation $\vec{Q}$ contain the weights $w(q_i, q)$. In VSM, the ranking function is given by the 'cosine similarity', $Sim(\vec{D}, \vec{Q})$, which is equal to the dot product of the document vector and the query vector, normalized by

the vector lengths:

$$Sim(\vec{D}, \vec{Q}) = \frac{\sum_{t_i \in q \cap d} w(t_i, d) w(t_i, q)}{|\vec{D}||\vec{Q}|}. \tag{2.2}$$

The term weights in Eq.2.2 are not given by the VSM model itself, but may be defined by various weighting schemes. Clearly, the value of the ranking function depends on the particular weighting scheme that is used, and different schemes may yield different ranking results. In fact, it is well established that the performance of an IR system depends crucially on an effective term weighting scheme (e.g. Salton and Buckley [68]).

In the following, we describe some developments of term weighting research.

2.3.1 The TF-IDF weighting

It can be expected that weighting a term by more detailed statistics of its occurrence in a document than the simple Boolean representation can help to better discriminate one document from another and thus aid document retrievals. Based on earlier works, Salton and Buckley [68] identified three important components that constitute an effective term weight:

- **term frequency (TF) in the document** The pioneering work of Luhn [53] in 1958 first considered using the 'occurrence frequency' (or 'term frequency' as it is now generally called in the literature), as an indicator of the significance of a term. The more times a term appears in a document, the more it is regarded being important. The term frequency of the term t in document d is denoted by

$f(t, d)$.

- **prevalence of the term in the collection** Words that appear in many documents in the whole collection are regarded as too common and are not good discriminators of a document's content. In 1972, based on this intuition , Spärck Jones [76] introduced the well known weighting that is now called 'inverse document frequency' (idf). The idf gives less weighting to terms that occur in many documents. A typical and simple definition of the idf is as follows [76]. For a corpus consisting of N documents, if a term t appears in n documents (with n denoting the 'document frequency' of the term), idf weighting is given by $idf(t) = \log(N/n)$.

- **length of the document** In a long and verbose document, there may be repeated use of words. Hence, long documents may contain more query terms than short documents. To avoid biasing long documents over short ones, [68] find that including a length normalization factor in the term weights will improve retrieval performance.

Salton and Buckley [68] performed retrieval experiments with the Vector Space Model using combinations of variants of TF and IDF components in the query and document vectors. Some of the variants of the term frequency, $f(t, d)$ that were tested in [68] include: (1) binary weight; and (2) 0.5 + 0.5 $f(t, d)/\max f(t, d)$. For idf, they also considered the variant $\log((N - n)/n)$. They found the product $f(t, d) \times \log(N/n(t))$ to be the most effective. In the literature, terms weights that have a generalized TF component multiplied by an IDF component are generally referred to as

'TF-IDF'.

Over the past years, various IR models have been developed, including the Vector Space Model described above and probabilistic models. The various IR models generally produce ranking functions that have forms which can be interpreted as a summation over some function of the weighting of query terms (e.g. including factors such as the term frequencies in the document and the query itself, $f(q_i, d)$ and $f(q_i, q)$). It is remarkable that many of the term weights derived by these IR models have the TF-IDF form. While the original introduction of the TF and IDF weights was heuristic in nature, the later derivations by the models may be regarded as theoretical justification of the weights.

One important variation of the TF-IDF form is the BM25 weight introduced in the Okapi system (Robertson et al. [63]). This is generally regarded as the state-of-the-art term weight. This weighting was derived within a probabilistic retrieval model, and a more detailed description of this weighting will be presented in Section 2.3.2. Apart from BM25, another important variant of the TF-IDF term weight that is used by various researchers (e.g. Wong et al. [89]) is the pivoted normalization weight of Singhal et al. [74]:

$$W_{PN}(d,q) = \sum_{t \in d \cap q} \frac{1 + \ln(f(t,d))}{1 + \ln(avgf(d))} \times \frac{1}{(1+s) + s\frac{|d|_1}{\Delta}} \times f(t,q) \cdot \ln \frac{N+1}{df}, \quad (2.3)$$

where s is a constant, $avgf(d)$ is the average term frequency in document d, $|d|_1$ is the City-block length of d, and Δ is the average document length in the collection.

2.3.2 Probabilistic retrieval model and the BM25 weighting

In IR, probabilistic retrieval models are generally based on ranking documents according to the Probability Ranking Principle, which states that retrieval effectiveness is greatest if documents are ranked in the order of decreasing probability of relevance to the query (e.g. Robertson [60]). Probabilistic retrieval was first suggested by Maron and Kuhn in 1960 [54]. Subsequently, various probabilistic models have been proposed, differing in the way the probability of relevance is estimated. Notable examples of probabilistic retrieval models include: the Binary Independence model (Robertson and Spärck Jones [62]), the logistic regression model (Cooper et al. [9]), the 2-Poisson model (Harter [28], Robertson and Walker [59]), and the language model (Ponte and Croft [57], Lafferty and Zhai [47]).

One of the earliest probabilistic retrieval models was developed by Robertson and Spärck Jones [62]. They assigned weightings to query terms according to the probabilities of term occurrence in relevant and irrelevant documents. The term weights derived by their model are generally called RSJ weights. In practice, the RSJ weights can be estimated if there are some known relevant documents, while in the absence of such information, the estimate of the RSJ weight reduces to an idf form [59].

In [62] , two probabilities are defined for the term t_i:

$$p_i = P(\text{document contains } t_i | \text{document is relevant}), \tag{2.4}$$

$$q_i = P(\text{document contains } t_i | \text{document is not relevant}). \tag{2.5}$$

The RSJ term weight is then given by:

$$w_i = \log \frac{p_i(1-q_i)}{(1-p_i)q_i}. \tag{2.6}$$

Suppose relevance information is available – R out of N documents are relevant, and r_i relevant documents contain the term t_i. Then, the estimates of p_i and q_i are:

$$p_i \approx \frac{r_i}{R}, \qquad q_i \approx \frac{n_i - r_i}{N - R}. \tag{2.7}$$

The RSJ weight is then approximated by [62]:

$$w_i = \log \frac{(r_i + 0.5)(N - R - n_i + r_i + 0.5)}{(R - r_i + 0.5)(n_i - r_i + 0.5)}, \tag{2.8}$$

where the value 0.5 is added to each of the components as a smoothing correction.

In the absence of relevance information, R and r_i are both set to zero in Eq.2.8, which becomes:

$$w_i = \log \frac{N - n_i + 0.5}{n_i + 0.5}, \tag{2.9}$$

thus yielding a weight with the IDF form.

Subsequently, the 2-Poisson model of Robertson and Walker [59] was introduced to model term frequencies by a mixture of two Poisson distributions. A series of best match term weighting functions were developed. In particular, this work led to the BM25 term weight in which the effect of document length is taken into account [61], [78]. The BM25 weight is essentially a special form of TF-IDF.

The BM25 ranking formula for a document d and query $q = \{q_1, \ldots, q_n\}$ is:

$$BM25(d,q) = \sum_{i=1}^{n} \frac{f(q_i,d) \cdot (k_1+1)}{f(q_i,d) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{\Delta}\right)} \cdot w_i(q_i) \cdot \frac{f(q_i,q)}{k_3 + f(q_i,q)} \quad (2.10)$$

where $|D|$ is the length of the document, Δ is the average document length in the text collection, and w_i is an IDF factor given by Eq.2.9.

2.3.3 Context-dependent term weights

The novel concept of context-dependent term weights was suggested by Wu et al. ([90] and [91]) via their nonrelevance decision model of IR. As discussed in Section 2.3.1, the traditional TF-IDF term weight of a term t depends on the occurrence statistics of t but not on the other terms in the document or in the collection. On the other hand, a context-dependent term weight of the term t is reweighted based on the document-contexts of t. The comparison of retrieval effectiveness using these weights and the performance using the traditional context-independent TF-IDF weights is the main focus of Chapter 3.

2.4 Clustering

Cluster analysis is applied in a wide range of disciplines such as pattern recognition [34], data mining [38], machine learning [6] and information retrieval [35]. In IR, as mentioned above, clustering is a possible solution to the word mismatch problem. In our work, we have studied the application of clustering techniques to our new method

of context-dependent term weights for RF. This will be reported in Chapter 5. In this section we give some background of cluster analysis, with a focus on IR.

We first provide describe several commonly used clustering algorithms (Section 2.4.1). Then we present a review of the application of clustering techniques in IR (Section 2.4.3), and various commonly used clustering evaluation measures (Section 2.4.4).

2.4.1 Clustering algorithm

Over the past few decades, a large number of clustering algorithms have been developed. An extensive review of these algorithms is given by Xu and Wunsch [94]. Here, we briefly describe some clustering algorithms that various researchers have used in IR.

Hierarchical clustering

Hierarchical clustering methods produce tree-like structures of objects, such that objects strongly similar to each other are grouped into small clusters, which are in turn nested within larger clusters containing less similar objects. Hierarchical clustering algorithms may be broadly divided into two categories: agglomerative and divisive. Agglomerative clustering starts from grouping the two most similar objects, and proceed to build the tree-like structure from the bottom to the top by adding less and less similar objects to the group. In divisive clustering, the single whole grouping of all the items is progressively subdivided into smaller clusters.

In IR, agglomerative clustering is preferred (van Rijsbergen [83], Willett [88]).

Hence, we will focus on agglomerative methods. There are several common agglomerative hierarchical clustering algorithms that are widely used. These differ in the determination of which documents or clusters are merged at each stage in the building of the hierarchical structure:

- **Single Linkage** In single linkage, the similarity between two clusters is the maximum of the similarities between pairs of items, with the members of a pair being taken from each of the two clusters. i.e. for two clusters C_a and C_b, the single linkage similarity is:

$$S(C_a, C_b) = \max_{i \in C_a, j \in C_b} (S(i, j)). \tag{2.11}$$

  The method is called single linkage because clusters are joined at each stage by the single strongest link between them. Single linkage tends to produce 'chain-like' structures.

- **Complete Linkage** In complete linkage, the similarity between two clusters is defined as:

$$S(C_a, C_b) = \min_{i \in C_a, j \in C_b} (S(i, j)). \tag{2.12}$$

  This method tends to produce small clusters that are tightly bound.

- **Group average** The group average method is intermediate between single linkage and complete linkage, with the similarity between two clusters being the average of the similarities between all pairs of items, with members of a pair being taken from each of the two clusters.

- **Ward's method** In this method, the clusters that are merged at each stage are chosen to minimize a certain objective function. For example, Ward's implementation used an error sum of squares objective function.

Partitional clustering

Partitional methods are the main techniques of non-hierarchical clustering. The one most commonly used is **K-means**. Partitional methods have the advantage of low computation costs, typically in the order of $O(N)$ for time complexity for clustering N objects, compared with $O(N^2)$ for hierarchical techniques. However, they generally require a number of experimental parameters, such as the number of partitions or clusters required. Also, there is an arbitrary aspect in having to select some documents as the initial seeds for clustering. These may be the reasons why partitional methods are not commonly used for document clustering in IR.

Fuzzy

Unlike hard partitional clustering where each item only belongs to one cluster, in fuzzy clustering an item is allowed to belong to all clusters with a degree of membership, $u \in [0, 1]$. In document clustering, if each cluster corresponds to a different subtopic, fuzzy clustering allows a document whose content includes several topics to belong to more than one cluster. An example of fuzzy clustering is the fuzzy c-means (e.g. Hoppner et al. [30] and Kummamuru et al. [44]).

2.4.2 Similarity score

Generally a clustering algorithm requires the input of the similarity score between pairs of items to be clustered, such as the scores $S(i,j)$ in Eq.2.11 and Eq.2.12. A common similarity score is the cosine similarity:

$$S(i,j) = \frac{\vec{v_i} \cdot \vec{v_j}}{|\vec{v_i}||\vec{v_j}|}, \tag{2.13}$$

where $\vec{v_i}$ and $\vec{v_j}$ are the vector representations of the items i and j. In the case of document clustering, $\vec{v_i}$ is the document vector $\vec{D_i}$. Typically, TF-IDF term weights (Section 2.3.1) are adopted for the vector representation. The 'cosine' of two documents has the geometric interpretation of being the 'angle' between the two document vectors in N-dimensional space, where N is the number of distinct terms in the corpus. Other measures that have been used by many researchers include the Dice coefficient and Jaccard coefficient (e.g. Jardine and van Rijsbergen [35]).

Dang et al. [13] distinguished between the traditional 'document based' similarity score defined by Eq.2.13 where the vector $\vec{v_i}$ is the document vector $\vec{D_i}$, and a new 'document-context based similarity'. The new similarity score $sim_c(D_i, D_j)$ is calculated as a function of the cosine similarity of pairs of contexts belonging to the two documents. The reason for using this context-based similarity is to reduce the effect of noise terms existing outside of the document-contexts. They found that using the new similarities could produce better clusters, as measured by the MK1 measure. The MK1 measure will be described in Section 2.4.4 below.

2.4.3 Clustering in IR

In this section we review past research in IR where clustering techniques have been applied.

Optimal Search

Cluster analysis was introduced in IR to improve retrieval efficiency [71] and effectiveness [83]. The *Cluster Hypothesis* introduced by Jardine and van Rijsbergen [35] has been the basis behind the effort of various researchers to apply clustering techniques in IR. The hypothesis states that documents relevant to the same query are more similar to one another than to irrelevant documents. In other words, if clustering algorithms are applied to a set of documents, then the relevant documents will be grouped together based on their similarity. In the ideal case, clusters containing relevant documents are well separated from the clusters of irrelevant documents. According to the cluster hypothesis, one would expect the similarity scores between relevant document pairs (Rel-Rel) to be on average larger than those between relevant and irrelevant (Rel-Irr) pairs. Hence, a distribution of similarity scores of Relevant-Relevant and Relevant-Irrelevant pairs should look like the forms shown in Fig.2.1.

For document clustering in IR, two broad classes of clustering methods that have been studied by various researchers are partitional clustering and hierarchical clustering (e.g. van Rijsbergen [83], Willett [88], Steinbach et al. [80]). Partitional methods such as K-means have the attraction of better time complexity than hierarchical methods. Early studies found that the effectiveness of searches using partitional methods was poorer than searches without clustering (Salton [71]), though recently some

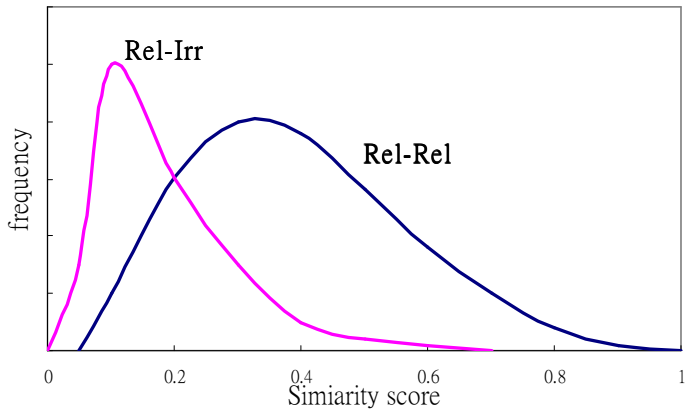

Figure 2.1: Distribution of similarity scores between Relevant-Relevant and Relevant-Irrelevant pairs.

promising results were reported by Steinbach et al. [80] with a 'bisecting K-means' algorithm. Some authors have demonstrated the potential effectiveness of retrieval based on hierarchical clustering (Griffiths et al. [22]; Tombros and van Rijsbergen [81]), whereas other work found retrieval using unclustered collections to be more effective (El-Hamdouchi and Willett [20]). For hierarchical clustering, various algorithms have been applied, including single linkage, complete linkage, group average and Ward's method. Numerous comparisons of these algorithms have been made (Voorhees [85]; El-Hamdouchi and Willett [20]; Griffiths et al. [22]). While K-means groups documents into disjoint partitions, hierarchical methods generate a tree structure in which documents that are more similar are nested within larger clusters containing less similar documents. In the context of web document clustering, Zamir and Etzioni [97] introduced the Suffix Tree Clustering algorithm which is based on identifying phrases common to groups of documents. They defined a base cluster to be a set of documents that share a common phrase. These base clusters may be overlapping in the sense that a document may appear in more than one of them. Overlapping clusters are also

produced by the Spherical Fuzzy c-Means algorithm of Kummamuru et al. [44].

Various retrieval strategies have been studied for searching a hierarchical system of documents (van Rijsbergen and Croft [84], Croft [11], Voorhees [85], El-Hamdouchi and Willett [20] and Willett [88]). With a top-down search, the query is first matched against the two child clusters of the root, and the sub-tree is chosen for which the query-cluster similarity is greater. The search then continues down the tree until some retrieval criterion is satisfied. A bottom-up search starts at the base of the tree and moves upwards until the retrieval criterion is satisfied. There are several approaches of selecting the starting point of this type of search. It may be a relevant document if any one is known. Otherwise, a nonclustered best match search can be performed, and the document that is most similar to the query is chosen to be the start of the bottom-up search. Another approach uses the bottom level clusters (Croft[11]). A bottom level cluster is the smallest cluster through which a document joins the hierarchy. Thus, for N documents, there are N bottom level clusters, up to $N/2$ of which can be duplicates. A scan of all the bottom level clusters is performed, and the one that best matches the query is chosen as the starting point of the bottom-up search. Some early studies of top-down or bottom-up searches involved the retrieval a single cluster (van Rijsbergen and Croft[84], Croft [11]). Griffiths et al. [22] found that searches which retrieved a single bottom level cluster often returned only two or three documents. Hence, they also considered either retrieving the 5 top-ranking bottom level clusters, or retrieving sufficient top-ranking clusters to cover 10 distinct documents. Another retrieval strategy that utilizes document clusters is to identify the clusters that are likely to contain relevant documents and then rank each of the documents in these clusters against the

query (Voorhees [85]). Some of the more recent work that seeks to identify 'high precision' clusters include that of Kurland et al. [46].

Query-specific clustering

In the early stages of research of clustering in IR, document clustering was performed on the entire corpus. This is called static clustering and the clusters thus formed are independent of the search queries. More recently, query-specific clustering has been studied by various researchers (Hearst and Pedersen [29]; Zamir and Etzioni [97], Iwayama [33], Leuski [51], Tombros et al. [81], Liu and Croft [52]). Instead of the entire corpus, query-specific clustering is performed on the retrieval results for each query. Assuming that the initial retrieved list, say of 1000 documents, are fairly well matched to the given query, clustering of these documents may be expected to have a larger chance of grouping together document relevant to the query. Tombros et al. [81] showed that query-specific clustering had the potential to increase the retrieval effectiveness compared to both static clustering and conventional document-based retrieval. Since query-specific clustering involves clustering only a small subset of the whole document collection, the time required for clustering is much less than static clustering. This is another advantage of query-specific clustering.

Query-specific clustering was also employed in the selection of relevant documents in a relevance feedback environment (see next subsection).

Clustering and relevance feedback

Various researchers have studied using clustering methods to improve the performance of PRF or RF. When clustering methods are applied in RF, the relevance judgments may be made either before clustering is applied ('pre-clustering relevance judgments') or after clustering is applied ('post-clustering relevance judgments'). Works where 'post-clustering relevance judgments' is adopted include those of Sakai et al. [67] and Lee et al. [50]. Sakai et al. [67] studied a selective sampling method which skips some top-retrieved documents based on a clustering criterion. The purpose of the sampling is to select a more varied set of documents for feedback and is based on the assumption that top-ranked documents may be too similar and redundant. However, they did not find significant improvements on NTCIR collections. Lee et al. [50] tested a cluster-based resampling method for PRF and found the method to be effective for PRF.

Buckley et al. [2] had found that in relevance feedback, the retrieval effectiveness increased with the number of known relevant documents. While a user may be expected to make relevance judgments on only a small number of documents, clustering may be a plausible method to increase the amount of relevance information for feedback to the retrieval system. This is the rationale of our approach to use clustering methods for RF, as presented in Chapter 5.

Clustering and presentation of retrieval results

While it is common for retrieval systems to return the retrieval results to the user as a ranked list, alternative ways of presentation have been studied by various researchers. In particularly, clustering has been utilized as a way to organize the search results

for presentation to the user. Hearst and Pedersen [29] introduced the Scatter/Gather cluster-based browsing method. In user-studies, their method allowed the users to easily locate clusters with the largest number of relevant documents. Leuski [51] also found that organizing search results by clustering was an effective way for the user to locate relevant material as quickly as possible. In this way, the clustering method can assist a user to select relevant documents for RF.

Clustering methods are actually implemented in some real commercial applications for presentation. For example, the web search engine Vivisimo (Koshman et al. [43]) returns search results in clusters which correspond to different sub-topics.

2.4.4 Clustering evaluation

With the abundance in clustering algorithms, it is important to evaluate the goodness of the clusters that are formed. In the literature, there are several methods of cluster evaluation. The appropriate method to use depends on the specific problem on hand. Some important methods are described in the following. In particularly, for IR a traditional measure is MK1. This is a natural measure to use in IR because unlike other schemes, it is actually based on precision and recall, which are the standard measures in IR.

Cluster validity

The general task of evaluating cluster goodness comes under *cluster validity* (see Halkidi et al. [23] for a review).

MK1

The E-measure was introduced by Jardine and van Rijsbergen [35] in IR to provide a measure of the quality of a document cluster. This measure is a composite of the recall and precision of a cluster and is given by Eq.(2.14)

$$E = 1 - \frac{(\beta^2 + 1)PR}{\beta^2 P + R} \tag{2.14}$$

where P and R are the standard precision and recall values, and β is a constant that specifies the relative weighting of precision and recall. The value $\beta = 1$ signifies equal importance of P and R. In the limit $\beta \to 0$, we have $E \to 1 - P$. Hence, values of $\beta < 1$ correspond to a 'precision-oriented' measure. On the other hand, in the limit $\beta \to \infty$, we have $E \to 1 - R$. Hence $\beta > 1$ correspond to a 'recall-oriented' measure. It can be seen from Eq.2.14 that the range of values of E is between 0 and 1. For the ideal case with $P = R = 1$, the equation gives $E = 0$, while the worst case $P = R = 0$ yields $E = 1$. Hence, a smaller numerical value of E corresponds to better cluster goodness. In the literature, it is also common to use the related F-measure, which is defined as $F = 1 - E$.

Jardine and van Rijsbergen [35] also defined the MK1 measure which is equal to the best E-measure attainable by retrieving a single cluster based on the clustering of a set of documents. This is a natural benchmark measure appropriate to applications where a single retrieved cluster is desired. Various authors have used this measure to evaluate clustering effectiveness. For example, Tombros et al. [81] compared the effectiveness of the various types of hierarchical clustering algorithms based on the

MK1 values obtained with the different algorithms. Dang et al. [13] used MK1 to examine the clustering effectiveness using the traditional ‘document-based’ similarity scores and new ‘context-based’ similarity scores.

In Chapter 4, we present a new class of measure of clustering effectiveness, called CS. Instead of a single ‘optimal cluster’, the CS measure is based on a combination of subclusters. This measure is appropriate for applications where it is desirable for objects of the same class be grouped in tight ‘subclusters’ corresponding to the ‘sub-classes’ or sub-topics. We will describe this measure in detail in Chapter 4.

Rand index

In fields outside of IR, the adjusted Rand index is commonly used in the statistics literature [31]. This index measures the agreement between the appearance of pairs of objects in each cluster, with their appearance in the assigned classes or categories.

Chapter 3

Context-Dependent Term Weights

Among the various methods that have been investigated in past research to tackle the polysemy problem in IR, query expansion (QE) via relevance feedback (RF) is a well established approach. More recently, Wu et al. ([92] and [91]) introduced 'document-contexts' (or simply contexts) as another solution to the problem. Contexts are basically fixed-sized text windows within a document. Suppose a document contains an occurrence of a particular query term q_i. The idea is that the specific usage of the term q_i in the document may be inferred by reading what other words appear in the neighbourhood of q_i. It is of interest to investigate whether document-contexts may be utilized in a RF setting, to further enhance the performance of the traditional QE. This chapter presents such an investigation.

Motivated by the work of Wu et al. [91], our new method uses a new 'context-dependent term weight'. Term weights have been an area of IR research that has attracted much attention in the past (Section 2.3). Many of the commonly used term weights are of the TF-IDF (term frequency $\times$ inverse document frequency) form. It

was shown in [91] that assigning TF-IDF weights to terms in a document can be interpreted as making relevance decisions in IR. They introduced a probabilistic nonrelevance decision retrieval model. Their model mimics a human making a series of 'local relevance decisions' by reading texts in the vicinity of all the query terms that occur in a document and deciding whether these portions of the document are individually relevant to the query. As such, the model is based on document-contexts centred on query terms. By adjusting the term frequency weightings of the query terms according to the local evidence of relevance extracted from the text windows, they derived a ranking formula that has a form similar to the BM25 weight (e.g. Robertson and Walker [59], Robertson [61]), which is essentially a special form of TF-IDF. The derivation of [91] involves making several assumptions, in particular the Minimal Context assumption, which states that for any query, the local relevance at a location k in a document d is determined only by the single term occurring at location k. As pointed out in [91], this assumption is not realistic because it is expected that the local relevance decision should depend not only on the occurrence of a query term, but also on the words around it in the text. They commented that such an unrealistic assumption may cause performance limitations of TF-IDF term weights. Hence, we are motivated to relax this assumption and investigate whether there is any advantage in term weights that depend on contexts with a size larger than unity. In comparison, the traditional TF-IDF value of a term t in a document d depends on the occurrence statistics of t in d or in the corpus, but does not depend on the other terms found in the document. In other words, the traditional TF-IDF is independent of the contexts of t.

We introduce a 'Boost and Discount' (B&D) procedure to compute the new context-

dependent term weights. B&D makes use of the any 'relevance information' that is available. In our current study, where B&D is applied in the RF setting, the relevance information is provided by the user's relevance judgments on some of the retrieved documents. The approach can be applied to other applications, such as text categorization ([72], [95]), where there are given training samples of documents belonging to different predefined categories. The B&D procedure adjusts the term-frequency component of the BM25 term weight of an initial query term q_i (i.e. not an expansion term) in an unseen document, according to the probability of relevance of the document-contexts centered on q_i. The probability of relevance of a context is estimated by a logistic regression model. In effect, the term-frequency weighting, $f(q_i), d$, of a query term q_i is promoted ('boosted') if it is surrounded by terms that are also observed in the surrounding of q_i in known relevant documents. Likewise, $f(q_i, d)$ is demoted ('discounted') if q_i is surrounded by terms that are also observed in its surrounding in known irrelevant documents. We define the 'surrounding' of q_i to be its document-context.

An overview of the rest of this chapter is as follows. In Section 3.2, we review the document-context retrieval model of [91], which provides the theoretical basis of the B&D procedure for calculating the new context-dependent term weights. The B&D procedure is introduced in Section 3.2. Section 3.3 contains our experimental results, with a comparison of the performance of RF using the new term weights and the baseline which uses the traditional context-independent BM25 weights.

3.1 Document-context based probabilistic non-relevance decision model

In this section, we first describe the probabilistic nonrelevance decision model of Wu et al. [91]. Various assumptions made in [91] will be discussed. We will adopt the same assumptions in our new context-dependent term weight method for RF.

Following [91], a document-context of a term t is defined as a fixed-size text-window centred on t:

> **Definition and Notation (Document-context).** The document-context (or simply context) of a term t is the text window consisting of n words centred at t. A context in the document d, centred at location k and having a window size of n terms is denoted by $c(d, k, n)$.

The above notation means that the context $c(d, k, n)$ consists of the $(n-1)/2$ terms on each side of the centre, as well as the central term, $d[k]$, itself.

In the model of [91], a human arrives at a 'document-wide relevance decision' by making a series of 'local relevance decisions', i.e. whether specific portions of the documents are individually relevant to the query. Such local judgments are made based on the following assumption:

> *Assumption (Context-Based Local Relevance Decision).* A local relevance decision at any location k in any document d for any query q is made on the basis of the information in the context that is centered at k in d for some maximal context size n.

The above assumption means that the user decides whether the portion of the document centred at location k is relevant solely by considering the words that occur in the vicinity of k, within a context of size n. The decision is not affected by what lies in other parts of the documents outside the context $c(d, k, n)$. Wu et al. [91] tested this assumption and found that given the context is large enough, the performance of the document-context dependent model does not change substantially. Therefore, we will assume that the Context-Based Local Relevance Decision is true provided that the context size parameter is appropriately calibrated.

Wu et al. [91] considered the local relevance $R_{d,k,q}$ and the document-wide relevance $R_{d,q}$ to be binary variables, having the value 0 for non-relevance and 1 for relevance. In this case, following the TREC evaluation policy for ad hoc retrieval tasks, a document is relevant ($R_{d,q} = 1$) if any part of the document is judged to be relevant, i.e. $R_{d,k,q} = 1$ for any k. This is named the Disjunctive Relevance Decision Principle by Kong et al. [42]. Alternatively, the document is irrelevant if $R_{d,k,q} = 0$ for all locations k in the document. Instead of having to scan through all locations k in a document, Wu et al. [91] assume that any relevant information for a query $q = \{q_1, q_2, \ldots, q_{NQ}\}$ can only be found in contexts centred on query terms $\{q_i\}$, i.e. at locations k such that the word at location k, $d[k]$, is one of the terms contained in q. This is called the Query-Centric assumption:

> *Assumption (Query-Centric).* For any query q and any relevant document d, the relevant information for q is located only in the contexts $c(d, k, n)$ for $k \in [1, |d|]$ and $d[k] \in q$. (i.e., the relevant information is located around query terms).

The Query-Centric assumption implies that $R_{d,k,q} = 0$ for all locations k where $d[k]$ is not a query term. The Context-Based Local Relevance Decision and Query-Centric assumptions together mean that the texts in the vicinity of the query terms will provide the information needed to decide whether the particular portion of the document is relevant. Specifically, the information is contained within a fixed-size text window centred on the query term. In practice, the size of the text window can be much smaller than the size of the whole document. Wu et al. [91] have tested this assumption which appears to hold in most cases, so we will simply accept it as true. Furthermore, the above assumptions imply that in performing query expansion in relevance feedback, the extraction of expansion terms can be limited to the contexts of query terms $\{q_i\}$. Another assumption made by [91] is the following:

> *Assumption (Location-Invariant Decision).* For any query q, if $c(d, j, n) = c(e, k, n)$, then the local relevance decisions made on $c(d, j, n)$ and $c(e, k, n)$ are the same.

The above assumption means that the relevance judgements on two identical contexts are the same, irrespective of the remaining content of the documents that contain them, or of the positions where they occur in the documents. This is quite reasonable, once the Context-Based Local Relevance Decision assumption is asserted. It is also in line with the bag-of-words model commonly adopted in IR, whereby only the occurence of a term is taken into account, but not its location in a document. Recently, there are studies which suggest that term location is important, e.g. chornological term ranking introduced by Troy et al. [82]. In [82], it is noted that the most important content

of a document often appears near the beginning of the document, such as in a news article. Hence, the Location-Invariant Decision assumption may not be always true, as it excludes position consideration. Nonetheless, in our current study we assume it is true for simplicity. This assumption may be relaxed in our future research (Chapter 6).

An important result of [91] is the demonstration that it is possible to derive the probability of a document d being relevant to query q, $p_{\nabla}(R_{d,q} = 1)$,in terms of the TF and IDF weights, where ∇ indicates that the relevance decision is document-wide. To achieve this derivation, it is necessary to make the following assumption:

> *Assumption (Minimal Context).* For any query, the local relevance at a location k in a document d is determined only by the single term $d[k]$.

As pointed out by [91], this is a rather unrealistic assumption. Basically, it means that the words close to a query term do not affect the local relevance judgment. As discussed in Section 3.2, this assumption is one that we will relax in our algorithm that calculates context-dependent term weights. In [91], there are other assumptions (e.g., query independent non-relevance probability assumption) related to the derivation of the inverse document frequency (IDF). The term weights that we use in our current study have an IDF component, and we will assume that these assumptions are true.

Having made the above assumptions, Wu et al. obtained the following:

$$p_{\nabla}(R_{d,q} = 1) \propto \sum_{t \in (V(q) \cap V(d))} \frac{f(t,d) \times idf(t)}{f(t,d) - \alpha + \alpha \frac{|D|_p}{|\Delta(d)|_p}} \tag{3.1}$$

where $\propto$ denotes the rank-equivalence relation, $|\cdot|_p$ denotes the p-norm length of its argument, $\Delta(d)$ is the normalized document d, $f(t,d)$ is the occurrence frequency of

t in d, $idf(t)$ is the inverse document frequency, $V(\cdot)$ denotes the set of distinct terms of its argument, while α is a constant parameter. To derive Eq.(3.1), Wu et al. replaced the term frequency $f(t,d)$ by a weighted version, $\omega(t,d) = f(t,d) \times p(f(t,d)|\bar{r})$, where $p(f(t,d)|\bar{r})$ is the probability that all occurrences of the term t in document d are locally non-relevant. Comparing with Eq.2.10, the expression in the summation of Eq.(3.1) is just the BM25 term weight [59]. Because of the query-centric assumption, the probability $p(f(t,d)|\bar{r})$ can be obtained by considering the evidence of relevance in the contexts within the document. Thus, the probabilistic nonrelevance decision model of [91] is equivalent to ranking documents according to a context-dependent version of the BM25 term weights, whereby $f(t,d)$ is replaced by $\omega(t,d)$.

3.2 Computing context-dependent term weights by Boost and Discount

As described in the preceding section (3.1), the probabilistic model of Wu et al. [91] yields context-dependent term weights which are identical to the BM25 weights, except that the term frequency $f(t,d)$ in the BM25 equation (Eq.2.10) is weighted by a factor related to the probability of relevance of the all the contexts of t. In this section we describe how the novel context-dependent term weights may be generated by a Boost and Discount (B&D) procedure. The B&D procedure is applicable when partial relevance information is available, such as in a relevance feedback or text categorization task. The theoretical justification of the method will be presented in the following.

There are two components in B&D: (1) Estimating the probability of relevance of

a context based on the available evidence, and (2) Calculating the context-dependent term weights based on the probability of relevance. These two components will be described in Section 3.2.1 and Section 3.2.2 respectively.

3.2.1 Modeling the probability of relevance of a context

In statistics, a popular method which can be used to model the probability of the occurrence of some event is *logistic regression* (e.g. Kleinbaum [41]). Logistic regression relates a set of independent variables to a dichotomous dependent variable via the logistic function, $f(z) = 1/(1 + e^{-z})$, which has a sigmoidal shape as shown in Fig.3.1. Because the value of $f(z)$ lies between 0 and 1, it is well suited to model probabilities. In general, the variable z is equal to the total contribution of a set of independent variables $\{x_i\}$:

$$z = \gamma_0 + \gamma_1 x_1 + \gamma_2 x_2 + \cdots + \gamma_n x_n. \tag{3.2}$$

The unknown parameters γ_i in Eq.3.2 are called logistic coefficients and indicate how strongly the occurrence of an event D depends on each of the variables x_i. Hence the *logistic model* may be stated as:

$$P(D = 1|x_1, x_2, \ldots, x_n) = f(z) = f(\gamma_0 + \gamma_1 x_1 + \gamma_2 x_2 + \cdots + x_n), \tag{3.3}$$

where $f(z)$ is the logistic function. For example, logistic regression is commonly used in epidemiology, whereby $f(z)$ models the probability of illness (e.g. heart disease) given a set of risk factors, $\{x_i\}$ (e.g. age, blood pressure, cholesterol level, etc.). In

general applications, the coefficients $\{\gamma_i\}$ are estimated by fitting data on the variables x_i to the observance of D in the samples. In our case, as discussed below, the coefficients are determined by calibrating them to yield the highest performance measure, such as MAP. This is because we wish to seek the model parameters that can produce the best retrieval performance.

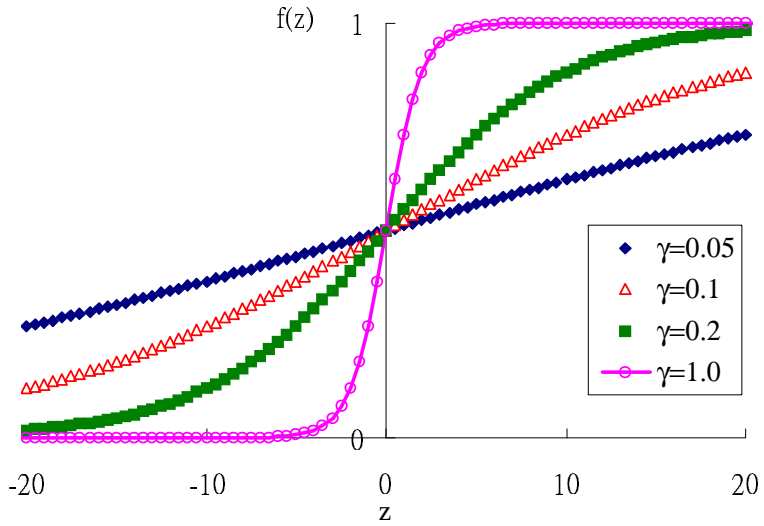

Figure 3.1: The logistic function, $f(z) = 1/(1 + e^{-\gamma z})$.

Logistic regression is well suited for our purpose to model the probability of relevance of a given context, and our B&D model will adopt this method. In the following we describe what the variables $\{x_i\}$ correspond to in our model.

According to the Query-centric assumption of Section 3.1, all evidence of relevance (or non-relevance) only appears within the contexts of query terms, $\{q_i\}$. In B&D, we assume that this evidence is provided by the words that co-occur with each query term q_i within a context centred on the term. Suppose some partial relevance information is available. For example, in relevance feedback, this information comes from a user's

judgment of a (small) number of documents to decide whether each of these is relevant or non-relevant. The query-centric assumption implies that this judgment is made by noticing the occurrence of certain words within the document-contexts centred on the query terms in each document. From each known relevant document, we extract the words that appear in all the contexts of each query term q_i. We call these words 'boost terms' for the query q_i. Note that each query term q_i has its own set of 'boost terms', which are denoted by $S_B(q_i)$. Formally, the boost terms are defined as the following.

> **Definition.** The boost terms, $S_B(q_i)$, are the set of all terms that co-occur with the query term q_i within all document-contexts of size C_B centred on q_i in all the known relevant documents R:
>
> $$S_B(q_i) = \bigcup_{\substack{d \in R \\ k \in Loc(q_i,d) \\ d[l] \in c(d,k,C_B) \wedge (l \neq k)}} d[l],$$
>
> where $d[l]$ is the term at location l in document d, and $Loc(q_i, d)$ denotes the set of all locations in the document d where the query term q_i occurs.

In the preceding definition, the term at the context centres, k, i.e. the query term q_i, is excluded because we consider the evidence of relevance to be based on the co-occurring terms but not the query term itself.

Similarly, from all the known irrelevant documents, we extract the co-occurring words in the contexts centred on q_i, to obtain a set of 'discount terms' denoted by $S_D(q_i)$:

> **Definition.** The discount terms, $S_D(q_i)$, are the set of all terms that co-

occur with the query term q_i within all document-contexts of size C_D centred on q_i in all the known irrelevant documents I:

$$S_D(q_i) = \bigcup_{\substack{d \in I \\ k \in Loc(q_i, d) \\ d[l] \in c(d,k,C_D) \wedge (l \neq k)}} d[l].$$

Note that Definition 1 and Definition 2 for the sets of boost and discount terms, $S_B(q_i)$ and $S_D(q_i)$, contain the constant parameters C_B and C_D, which are the sizes of the contexts from which the co-occurring terms are extracted. Because terms that occur in both known relevant and known irrelevant documents may not provide clear evidence of relevance for an unseen document, in our experiments we remove from the sets $S_B(q_i)$ and $S_D(q_i)$ any term found in their intersection. While the current defintions fo the boost and discount terms are quite strong, relaxations in the term selection may be considered in our future studies. For example, instead of removing from $S_B(q_i)$ and $S_D(q_i)$ all terms found in their intersection, we may retain such terms in $S_B(q_i)$ if the term occurs only once among all judged contexts, and similiarly for terms in $S_D(q_i)$.

Now consider a context $c(d, k, C_m)$, which has a size of C_m words centred on an occurrence of q_i at location k in an unseen document d. Again by the Query-Centric assumption, if the words in the context are similar to the 'boost terms' in the set $S_B(q_i)$, this would support the document as likely to be (locally) relevant too. The more 'boost terms' are found in the context, the higher is the probability that the context is relevant. On the other hand, if 'discount terms' are found in the context, it means there is a reduced probability that the context is relevant. Therefore, in the

logistic model (Eq.3.3), we define two independent variables X_B and X_D which are proportional to counts of boost and discount terms in the context, respectively. Hence, for the probability of the context $c(d, k, C_m)$ to be relevant, $P(R = 1) = f(z)$, where

$$z = \gamma_0 + \gamma_B X_B - \gamma_D X_D. \tag{3.4}$$

In Eq.3.4, the independent variables X_B and X_D may be generally defined as weighted counts:

$$X_B(d, k) = \sum_{d[l] \in c(d,k,C_m) \wedge (l \neq k)} w_B(d[l], q_i) \tag{3.5}$$

where the sum is over all locations within the context at $c(d, k, C_m)$ excluding the centre (location k), and $w_B(d[l], q_i)$ is a weight for the term at location l. In Eq.3.5 the term at location k, i.e. the query term q_i, is not counted because we consider the evidence of relevance to be only based on the co-occurring terms. The simplest choice of the weight w_B in Eq.3.5 is the unit count:

$$w_B(d[l], q_i) = \begin{cases} 1 & \text{if } d[l] \in S_B(q_i) \\ 0 & \text{otherwise} \end{cases}. \tag{3.6}$$

Another possible choice is the 'idf-weighted' count:

$$w_B(d[l], q_i) = \begin{cases} idf(d[l])/idf_0 & \text{if } d[l] \in S_B(q_i) \\ 0 & \text{otherwise} \end{cases}. \tag{3.7}$$

In the above equation, we use a common expression for the IDF: $idf(d[l]) = \log_{10}((N+$

$0.5)/(df(d[l]) + 0.5))$ where N is the total number of documents in the collection, $df(d[l])$ is the document frequency of $d[l]$. The factor $idf_0 = \log_{10}((N+0.5)/0.5)$ normalizes the weight w_B to between 0 and 1. An idf-weighted count gives less weighting to words that are too common, and conforms to the intuition that such words are not good discriminators to provide evidence of relevance. We have tested both types of weighted counts, Eq.3.6 and Eq.3.7 in our experiments. For averaging over the 50 queries of TREC2005, the MAP values obtained using idf-weighted counts and unit counts are 0.2951 and 0.2503 respectively. Hence, we find that the idf-weighted counts give better results. In Section 3.3, we will only report experiments that use the weighting of Eq.3.7. The discount $X_D(d,k)$ are defined similarly by matching the context words with those in the set of discount terms, $S_D(q_i)$:

$$X_D(d,k) = \sum_{d[l] \in c(d,k,C_m) \wedge (l \neq k)} w_D(d[l], q_i) \tag{3.8}$$

with

$$w_D(d[l], q_i) = \begin{cases} idf(d[l])/idf_0 & \text{if } d[l] \in S_D(q_i) \\ 0 & \text{otherwise} \end{cases} . \tag{3.9}$$

The probability $P(R)$ for a context is interpreted as a degree of belief that the context is relevant (e.g. Dang et al. [14]). $P(R) = 1$ and $P(R) = 0$ correspond respectively to a firm belief that the context is relevant and the belief that it is irrelevant. The value $P(R) = 0.5$ means there is total uncertainty regarding its relevance. In the absence of any 'boost' or 'discount' terms in an unseen context, there is no available evidence to indicate its relevance or non-relevance. In this case, the probability $P(R)$

should be 0.5. Therefore we derive the value of the coefficient γ_0 in Eq.3.3 to be $\gamma_0 = 0$, as the value of the logistic function at $z = 0$ is $f(0) = 0.5$. In our experiments, we will obtain the values of the logistic coefficients γ_B and γ_D by calibrating them to yield the highest performance evaluation measure, such as MAP. The calibration results will be presented in Section 3.3.

Note that Eq.3.5 and Eq.3.8, which calculate the weighted count of the words, contain the constant parameter C_m, i.e. the size of the context in an unseen document for matching the boost and discount terms.

In our current study, we have treated all B&D terms uniformly irrespective of their position of occurence within a context. Hence, in Eq.3.5 and Eq.3.8 for the weight for each term, $w_B(d[l], q_i)$ and $w_D(d[l], q_i)$, we have used idf-weighting without any positional factor. In our future work, various positional weightings of the terms may be studied, for example, the distance of a term from the context centre. Another example is term ordering, i.e. whether the term occurs on the left or right side of the context centre. With respect to the Location-Invariance Decision assumption mentioned in the previous section, we may also relax the assumption by giving a larger weighting to terms that occur near the beginning of a document. The vaious positional factors can be easily included in our procedure by adjusting the weights $w_B(d[l], q_i)$ and $w_D(d[l], q_i)$.

3.2.2 Calculating the context-dependent term weights

The retrieval model of Wu et al. [91] derived a ranking function in which the term weights have the BM25 form, with the difference that the term-frequency $f(t, d)$ is adjusted by a factor related to the probability of relevance of the contexts of t. In

other words, the term weights become 'context-dependent'. We now explain how the context-dependent term weights are computed in the B&D procedure, utilizing the probability of relevance of a context as estimated in the method described in Section 3.2.1 above.

According to the Query-centric assumption (Section 3.1), all evidence of relevance only appears within the contexts of *query terms*. Therefore in B& D, we adopt the BM25 term weights (Eq.2.10) for all terms that are not in the original query, while the query terms $\{q_i\}$ are weighted by a BM25-like form with $f(q_i, d)$ in (Eq.2.10) being replaced by a new component, $f_{BD}(q_i, d)$. We write $f_{BD}(q_i, d)$ in the following form:

$$f_{BD}(q_i, d) = f(q_i, d) + \Delta f_{BD}(q_i, d), \tag{3.10}$$

in which $\Delta f_{BD}(q_i, d)$ is the adjustment according to the probability of relevance of the contexts of q_i. In particularly, $\Delta f_{BD}(q_i, d)$ should be proportional to the probability of relevance. Furthermore, $\Delta f_{BD}(q_i, d)$ should be equal to zero in the absence of evidence of relevance (i.e. when $P(R) = 0.5$). Therefore, we define $\Delta f_{BD}(q_i, d)$ as in the following, to satisfy these requirements:

$$\Delta f_{BD}(q_i, d) = \sum_{k \in Loc(q_i, d)} D \times (P(R(c(d, k, C_m))) - 0.5), \tag{3.11}$$

where the multiplicative factor D may be interpreted as a document length which converts the probability $P(R(c(d, k, C_m)))$ to a frequency count. In Eq.3.11, for a context without any evidence of relevance, so that $P(R(c(d, k, C_m))) = 0.5$, and

$P(R(c)) - 0.5 = 0$, thus giving no contribution to $\Delta f_{BD}(q_i, d)$ as desired.

Using the logistic regression model estimate of the probability of relevance as described in the previous section, Eq.3.11 becomes:

$$\Delta f_{BD}(q_i, d) = \sum_{k \in Loc(q_i, d)} D \times (f(\gamma_B X_B(d, k) - \gamma_D X_D(d, k)) - 0.5), \qquad (3.12)$$

where $f(\cdot)$ is the logistic function.

If the contexts of q_i contain many terms which match with the discount terms $S_D(q_i)$, so that the weighted count $X_D(d, k)$ becomes sufficiently large, it is possible for Δf_{BD} in Eq.(3.12), and hence f_{BD} in Eq.(3.10), to become negative. In this case, it is actually desirable that the term weight $f(q_i, d)$ also goes negative, as this will indicate that the document is irrelevant based on the evidence of the contexts. To allow the sign of the term weight to follow that of f_{BD}, we modify the BM25 term-frequency factor to:

$$f_{BM}(q_i, d) = \frac{abs(f_{BD}(q_i, d))}{abs(f_{BD}(q_i, d)) + k \left[1 - b + b\frac{|d|_2}{\Delta}\right]} \cdot \text{sign}(f_{BD}(q_i, d)), \qquad (3.13)$$

where $|d|_2$ is the Euclidean length of d, and Δ is the average Euclidean length of all documents in the collection. Adding the $\text{sign}(f_{BD}(q_i, d))$ factor in Eq.3.13, will let $f_{BM}(q_i, d)$ become negative when $f_{BD}(q_i, d)$ is negative, as desired.

Fig.3.2 depicts the flow of relevance feedback, with context-dependent term weights being obtained by the B&D procedure. In the figure, the left branch corresponds to standard QE, with the term weights of both the initial query terms and expansion

terms being given by the traditional BM25 equation (Eq.2.10). The right branch corresponds to context-dependent term weights being used for the initial query terms. Referring to Fig.3.2, compared with traditional query expansion using standard BM25 weights alone, the additional steps in the B&D procedure include: (1) the extraction of boost/discount terms from the known relevant and non-relevant documents and (2) scanning through each context of every query term $\{q_i\}$ and matching with the boost/discount terms. Our current focus is on investigating the effectiveness of context-dependent term weights. Efficiency consideration is not our primary concern in this study.

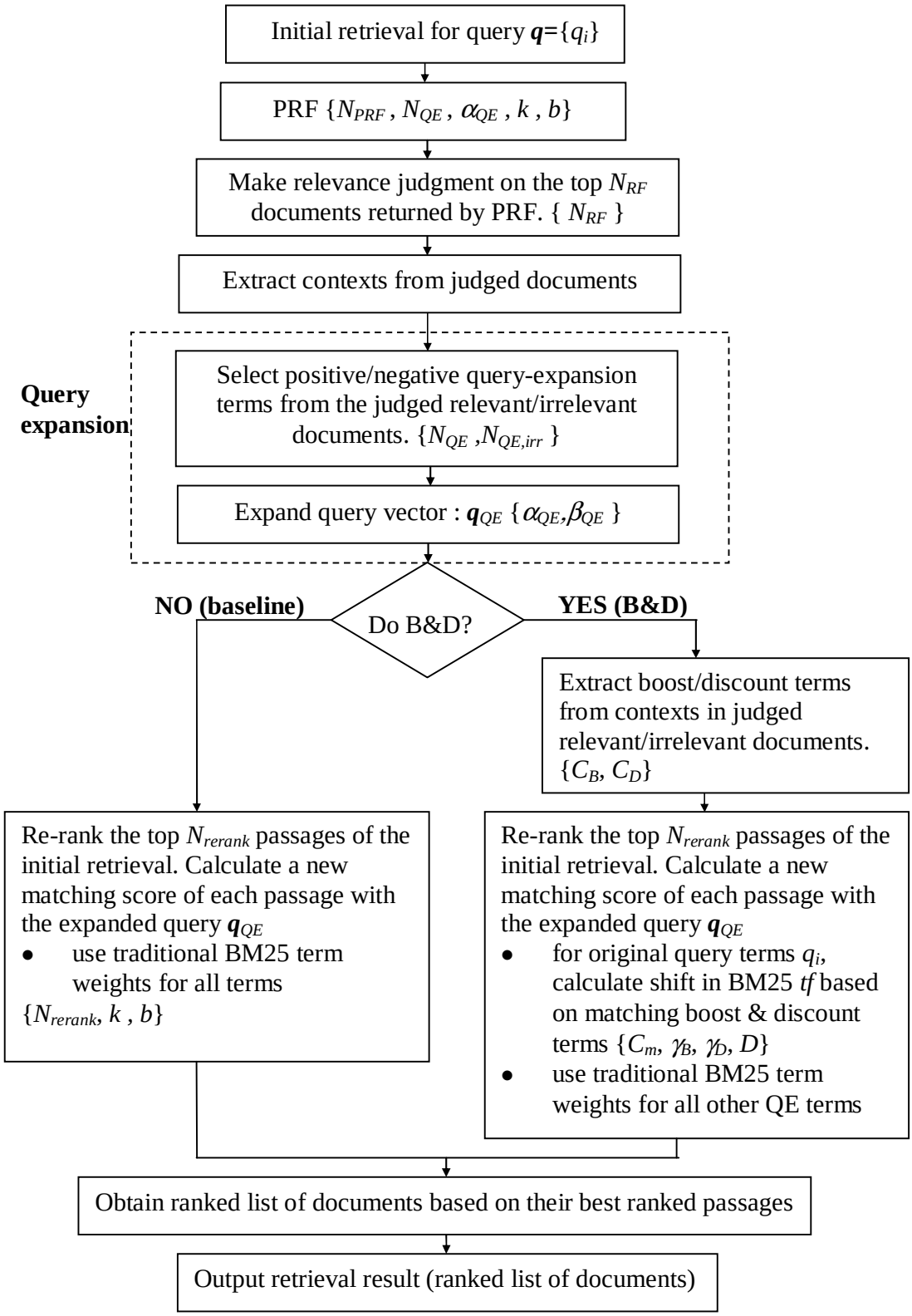

Figure 3.2: Flow diagram of relevance feedback with standard query expansion or with B& D. The parameters used in the various steps are indicated in curly brackets.

3.3 Experiments

The main goal of our current study is to compare the effectiveness of retrieval that uses context-dependent term weights with the traditional TF-IDF weights in the RF setting. As explained above, the context-dependent term weights are only calculated for the initial query terms $\{q_i\}$. We use the state-of-the-art BM25 term weights for all query expansion terms that do not appear in the initial query.

As indicated in Fig.3.2, after an initial retrieval with the query q_i, pseudo-relevance feedback (PRF) is performed. The list of documents retrieved by PRF is then supplied to the RF stage of the experiments, where relevance judgments are made on the top N_{RF} documents in the list. The main reason why we include PRF is to obtain a strong baseline for comparison with the results of using the new term weights. Using PRF will produce a retrieval list that contains more highly ranked relevant documents than the initial retrieval, i.e. there will be more relevant documents in the top N_{RF}. Past research (e.g Buckley et al. [2] showed that the effectiveness of RF increased with the number of known relevant documents. Hence, using the PRF produced list, rather than the initial list, should yield a better performance for the baseline. Any further improvement obtained by using the new term weights would then be a more convincing demonstration of the effectiveness of these term weights.

Section 3.3.1 describes the experimental environment and setup. Section 3.3.2 presents the calibration of various model parameters: (A) pseudo-relevance feedback (PRF),(B) Query expansion using standard BM25 term weights, (C) B&D procedure. Section 3.3.3 contains the comparison of the performance of relevance feedback using

the context-dependent term weights against the baseline.

3.3.1 Experimental environment and setup

We have performed experiments using the 50 title queries of each of the TREC-6, 7, 8 and 2005 test collections. In Section 1.3 we have described the reasons for choosing these various collections in our experiments. Some statistics of the collections are shown in Table 1.1. Here, we have not used TREC-2 which contains some long queries, with an average of 3.8 terms per query. On the other hands, the title queries of TREC-6, 7, 8 and 2005 all contain an average of close to 2.5 terms. The various parameters used by the baseline (QE with BM25 term weights) and the B&D procedure are calibrated using the TREC-2005 collection. The reason for choosing this collection as the basis of parameter calibration is its much large size compared with previous collections and therefore more in-line with current and future web search applications. In order to demonstrate the robustness of the B&D algorithm across collections, we conducted experiments for the other collections (i.e. TREC-6, 7 and 8) using the same set of parameters optimized for TREC-2005. In our experiments, the standard IR techniques of *stemming* and *stop-word removal* are applied. Stemming refers to converting words into their root forms. For example, the words *retrieval, retrieve, retrieved, retrieving* will all be converted to the root representation, *retrieV*. We have used the common Porter stemming algorithm [58] on all documents and queries. *Stop-word removal* means that words which are considered non-informative (e.g. prepositions, *the, a, and, or, of,* etc.), are removed from the documents and queries.

Our retrieval system is 'passage-based' rather than 'document-based', following

previous studies (e.g. Callan [5] and Kaszkiel and Zobel [39]) which found that passage-based retrievals could yield better retrieval results. In passage-based retrieval, each document is divided into blocks, called 'passages'. In our system, each passage contains a fixed number of words. All the passages are ranked according to the values of their ranking function. The final retrieved units are then the original documents ranked according to their highest ranking constituent passages [5]. It was found in [39] that a passage size of between 150 and 300 words gave the best performance. Hence, our experiments have used passages that consist of 250 words.

Regarding the evaluation of our RF retrieval results, the well established residue MAP measure [66] is used, as described in Section 1.3. The residue MAP is calculated based on remaining relevant document in the residue collection, from which the N_{RF} judged documents are removed.

Last, we briefly describe some assumptions that we have used in our experimental environment for RF. These assumptions are mainly adopted following Wong et al. [89]: (1) Identical judgment assumption: For a given query, the relevance judgment made by a user in RF is identical to the relevance judgment made by the evaluator for all documents in the collection; (2) Independent assessment assumption: The relevance judgment for the same document and the same query is the same irrespective of the relevance judgment of other documents and queries; (3) Non-identifying term assumption: Query terms which are not in the initial query formulated by the user should occur in more than one document in the collection, thus preventing these query terms from uniquely identifying a relevant document; (4) Default irrelevance assumption: If the relevance of a document to a query has not been judged by an evaluator,

Table 3.1: List of parameters used in pseudo-relevance feedback (PRF), query expansion and 'Boost & Discount'

Symbol	Description
N_{PRF}	Number of top passages assumed to be relevant in PRF
N_{RF}	Number of relevance judgments made in RF
N_{QE}	Number of query expansion terms selected from judged relevant documents
$N_{QE,irr}$	Number of query expansion terms from judged irrelevant documents
α_{QE}	Weight of original query vector in the expanded query vector
β_{QE}	Weight of positive vs. negative components of the query-expansion vector
k	Scaling in the BM25 term-frequency factor
b	Slope in the BM25 term-frequency factor
N_{rerank}	Number of passages returned by an initial retrieval to be re-ranked in RF
C_B	Size of contexts in relevant documents for "Boost" terms extraction
C_D	Size of contexts in irrelevant documents for "Discount" terms extraction
C_m	Size of contexts in an unseen document for matching B&D terms
γ_B	Logistic coefficient for Boost terms
γ_D	Logistic coefficient for Discount terms
D	Multiplicative factor controlling the strength of B&D (Eq.3.11)

the document is assumed to be irrelevant to the query [87].

3.3.2 Calibration of model parameters

In this subsection, we describe the calibration of the various model parameters using the TREC-2005 collection. A list of the parameters is summarized in Table 3.1.

Pseudo-relevance feedback (PRF)

In passage-based retrieval, PRF makes the assumption that the top N_{PRF} passages returned by an initial retrieval are relevant, where N_{PRF} is a fixed number. The original query vector $\vec{q}$ is thus modified by adding terms selected from these top N_{PRF} passages and a second retrieval is performed. However in our experiments, instead of simply selecting terms from the top N_{PRF} passages, we made a modification in the PRF scheme for the following reason. We have observed that the TREC-2005 collection contains many document duplicates. For example, two different documents in the collection may be essentially the same news article, but differ by having different time tags. Large portions of these two documents are actually identical. It is quite probable that the top N_{PRF} passages also consist of duplicates. Including duplicates in the PRF process would bias toward the duplicated passages. Therefore, instead of simply using the top N_{PRF} passages for term selection, we check for duplicate passages, skipping any duplicates found. The duplicates are detected by having the same matching score with the query, as well as having identical vocabularies above a 95% threshold. The matching score for a passage is obtained by summing the term weights of all the query terms appearing in the passage. Our condition for duplicity means that the passages must have the same occurrences of the query terms *and* have vocabulary overlaps above the 95% threshold.

After skipping any duplicate found, the QE terms are then selected from the top distinct N_{PRF} passages. As in standard RF, the expansion terms are selected according to a ranking score (Harman [26]). For each term t appearing in the top N_{PRF} passages,

we compute a score:

$$Score(t) = \frac{freq(t)}{1 + freq(t)} \times idf(t) \times pf(t) \times \left(1 + \frac{tmprf(t)}{1 + tmprf(t)}\right) \qquad (3.14)$$

where $freq(t)$ = total term frequency of t in the top distinct N_{PRF} passages, $pf(t)$ = number of these passages contain t, $idf(t)$ = inverse document frequency of t in the whole collection, and $tmprf(t) = df(t) - pf(t) + 1$, with $df(t)$= document frequency of t in the collection. The first three factors in Eq.3.14 is similar to one of the ranking scores found to be effective by Harman [26], who used a 'noise' factor similar to IDF. The rationale for including the last bracketed factor in Eq.3.14 is as follows. In this factor, $tmprf(t)$ is proportional to the number of passages in the collection that contain term t, apart from those in the top N_{PRF}. If $tmprf(t)$ is small, the term t would not be very useful as a query expansion term because it occurs rarely. Hence, we add a factor that reduces the score of those terms that have small values of $tmprf(t)$. The last factor in Eq.3.14 serves this purpose because it is monotonically increasing, bound between 1 and 2. Averaging over 50 queries of TREC-2005, we found that the MAP values obtained by including or excluding the $tmprf(t)$ factor in Eq.3.14 are 0.2781 and 0.2776 respectively. The query expansion vector $\vec{q}_{QE\_PRF}$ is made up of the top N_{QE} terms with the highest scores given by Eq.3.14. In our experiments, we set $N_{PRF} = 20$ and $N_{QE} = 80$, since these values were shown to be effective in our early studies.

An expanded query vector $\vec{q}_{PRF}$ is then obtained by mixing the initial query $\vec{q}$ with

the vector $\vec{q}_{QE_PRF}$:

$$\vec{q}_{PRF} = \alpha_{QE}\frac{\vec{q}}{|\vec{q}|} + (1 - \alpha_{QE})\frac{\vec{q}_{QE\_PRF}}{|\vec{q}_{QE\_PRF}|} \tag{3.15}$$

where α_{QE} is a mixing factor with a value between 0 and 1, and $|\vec{q}|$ and $|\vec{q}_{QE\_PRF}|$ are city-block lengths. A second retrieval is performed for the new query $\vec{q}_{PRF}$. In the calibration of PRF, we seek the set of parameters that yields the most relevant documents in the top 20 (i.e. the best P@20), so that the largest amount of relevant information will be available for RF that makes 20 relevance judgments ($N_{RF} = 20$). The set of parameters that we calibrate is $\{\alpha_{QE}, k, b\}$, where k and b are parameters in the BM25 term-frequency factor, Eq.2.10. The standard values of $\{k, b\}$ are $\{1.2, 0.75\}$ [59]. In our calibration, we allowed different values of $\{k, b\}$ for terms contained in the initial query $\vec{q}$ and the QE terms $\vec{q}_{QE_PRF}$. For TREC-2005, the optimal set of parameters that we found is summarized in Table 3.2.

Table 3.2: Summary of parameters used for pseudo relevance feedback (PRF).

			Initial query terms		Query expansion terms	
N_{PRF}	α_{QE}	N_{QE}	k	b	k	b
20	.25	80	5.0	.65	1.2	.75

Calibration of the baseline (BM25 term weights)

In this section, we present calibration results of our RF baseline model which uses traditional BM25 term weights for all terms in an expanded query. One important difference between RF and PRF as described above is that in RF, some of the docu-

ments may be judged as irrelevant, whereas in PRF (for passage-based retrieval) the top N_{PRF} passages are simply assumed to be relevant. Therefore in RF, 'irrelevance' as well as 'relevance' information is available. In query expansion, we include terms extracted from known irrelevant documents as negative components of the expanded query (see Eq.3.20). It should be noted that in RF, a user makes relevance judgements on whole documents, and so information from the whole of the judged documents is used for feedback. However in the passage-based PRF, information contained in the top retrieved *passages* is used.

In our RF experiments, relevance judgments are made on the top N_{RF} documents returned by PRF. Similar to the procedure for PRF (Section 3.3.2), because of the issue of duplicate documents in TREC-2005 we check for duplicates in the PRF retrieval output, skipping any duplicate found. Because of the Context-Based Local Relevance Decision and the Query-Centric assumptions (Section 3.1), we select QE terms from the document-contexts of the top N_{RF} documents, rather from the whole documents. We have tested several context sizes with the 50 queries of TREC-2005. With context sizes of 21, 41, 61 and 81, we obtained MAP values that vary from 0.286 to 0.291, and the differences are not statistically significant. In the experiments reported here, a context size of 41 (i.e. the centre query term plus 20 words on each side of it) is used for QE terms selection. Suppose within the top N_{RF} distinct documents, there are N_{rel} relevant documents, $\{R\}$, and N_{irr} irrelevant documents, $\{I\}$. We define a ranking score that is a variation of the 'offer weight', $OW(t)$, used for expansion term selection in the Okapi system (e.g. Robertson et al. [59] and Spärck Jones et al. [78]).

For a term t that appears in one or more of the documents in $\{R\}$, the score is:

$$S_{rel}(t) = \frac{f_{rel}(t)}{1 + f_{rel}(t)} \times OW(t) \times \left(1 + \frac{d_{rel}(t)}{1 + d_{rel}(t)}\right), \tag{3.16}$$

where $OW(t)$ is the offer weight given by:

$$OW(t) = r(t) \times \log_{10}\left\{\frac{[r(t) + 0.5] \times [N - df(t) - N_{rel} + r(t) + 0.5]}{[df(t) - r(t) + 0.5] \times [N_{rel} - r(t) + 0.5]}\right\}, \tag{3.17}$$

where N is the total number of documents in the collection and $r(t)$ is the number of documents in R containing t. In Eq.3.16, $f_{rel}(t)$ is the total number of occurrences of t in the set $\{R\}$,and $d_{rel}(t) = df(t) - r(t) + 1$. Eq.3.16 has the same form as Eq.3.14, the ranking score used in PRF. The difference is that for Eq.3.16, the scores are calculated based on known relevant documents, as opposed to the blind feedback of the top N_{PRF} passages for Eq.3.14. We have added the $f_{rel}(t)$ factor in Eq.3.16 because Harman [26] showed that a ranking function including a $f_{rel}(t)$ factor enhanced the performance of QE. The purpose of the $d_{rel}(t)$ factor in Eq.3.16 is the same as that of the $tmprf$ factor in Eq.3.14, namely to reduce the score for terms that rarely occur in the unseen documents. In analogy to the Eq.3.16, we define the score for a term t appearing in judged irrelevant documents as follows:

$$S_{irr}(t) = \frac{f_{irr}(t)}{1 + f_{irr}(t)} \times OW_{irr}(t) \times \left(1 + \frac{d_{irr}(t)}{1 + d_{irr}(t)}\right). \tag{3.18}$$

where $f_{irr}(t)$ is the total number of occurrences of t in $\{I\}$, $d_{irr}(t) = df(t) - i(t) + 1$,

with $i(t)$ being the number of known irrelevant documents containing the word t, and

$$OW_{irr}(t) = r(t) \times \log_{10} \left\{ \frac{[i(t) + 0.5] \times [N - df(t) - N_{irr} + r(t) + 0.5]}{[df(t) - i(t) + 0.5] \times [N_{irr} - i(t) + 0.5]} \right\}. \quad (3.19)$$

We construct the query expansion vector $\vec{q}_{rel}$, whose elements correspond to the N_{QE} terms in $\{R\}$ having the highest scores $S_{rel}(t)$. The vector $\vec{q}_{irr}$ is similarly constructed, with $N_{QE,irr}$ terms selected from $\{I\}$ based on the highest term weights $S_{irr}(t)$. The overall query expansion vector $\vec{q}_{QE\_RF}$ is then obtained by mixing $\vec{q}_{rel}$ and $\vec{q}_{irr}$:

$$\vec{q}_{QE\_RF} = \beta_{QE} \frac{\vec{q}_{rel}}{|\vec{q}_{rel}|} - (1 - \beta_{QE}) \frac{\vec{q}_{irr}}{|\vec{q}_{irr}|}, \quad (3.20)$$

where β_{QE} is a mixing constant with a value between 0 and 1. In Eq.(3.20), the terms extracted from the known irrelevant documents are given a negative weight. Finally, the RF query vector $\vec{q}_{RF}$ is a weighted sum of the initial query $\vec{q}$ and the query expansion vector $\vec{q}_{QE\_RF}$:

$$\vec{q}_{RF} = \alpha_{QE} \frac{\vec{q}}{|\vec{q}|} - (1 - \alpha_{QE}) \frac{\vec{q}_{QE\_RF}}{|\vec{q}_{QE\_RF}|}. \quad (3.21)$$

Rather than performing a new retrieval using the query $\vec{q}_{RF}$, we re-rank the passages returned by the initial retrieval for the original query $\vec{q}$. This re-ranking ensures that all the passages and hence the final documents retrieved contain at least one of the original query terms. The re-ranking is done by calculating a new ranking score of each passage for the query $\vec{q}_{RF}$. Suppose N_P passages are returned by the initial retrieval. Rather than re-ranking all of these passages, we only re-rank the top min(N_{rerank}, N_P)

of the passages, where N_{rerank} is a constant parameter. The reason for doing this is that we assume that the lowly ranked passages returned by the initial retrieval are unlikely to be relevant. Re-ranking only the top passages will avoid the bottom passages from being spuriously promoted in the re-ranking. Without loss of generality, N_{rerank} can be set to N_P to cover all passages.

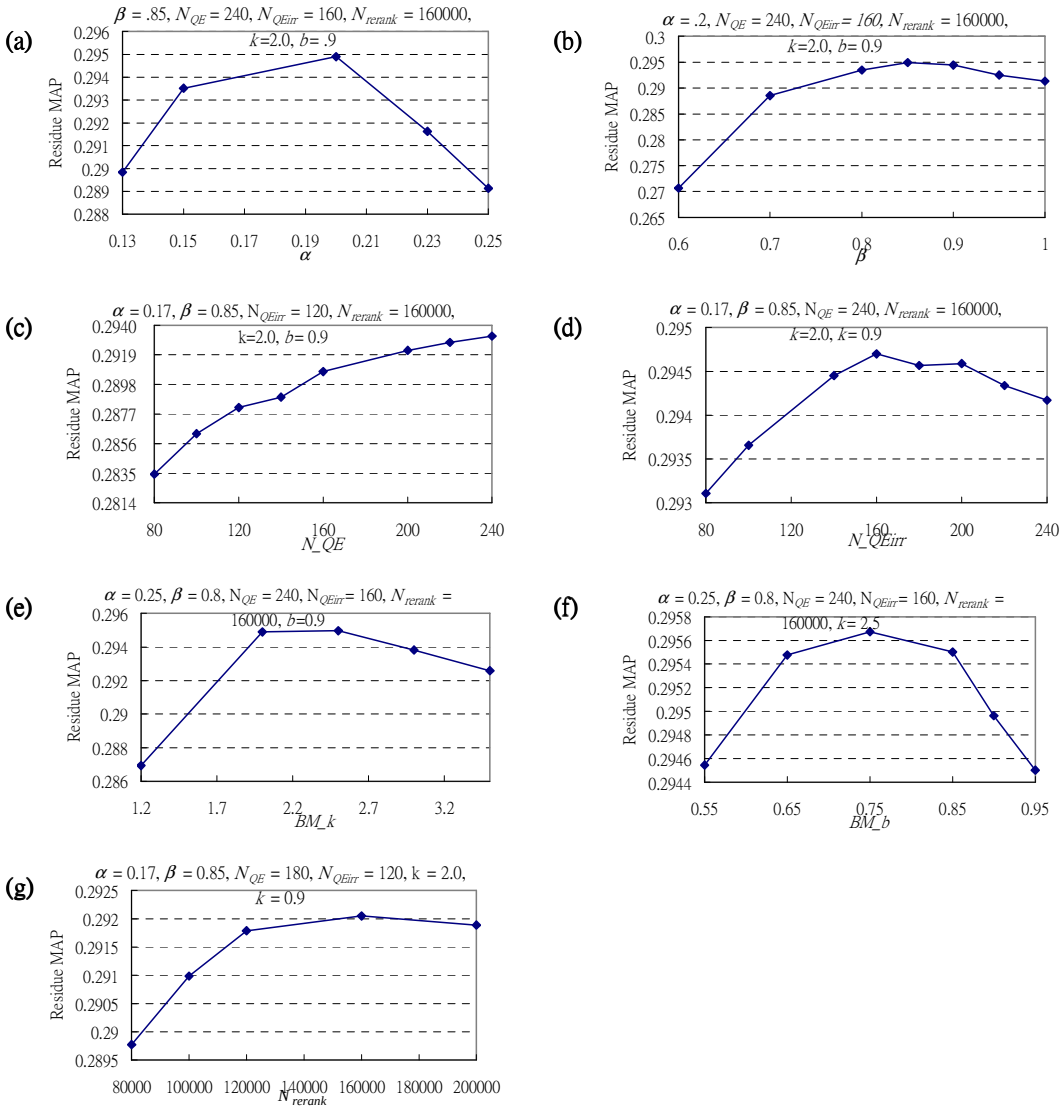

Figure 3.3: Calibration of Query Expansion via RF parameters, based on averaging over 50 title queries of TREC-2005.

We perform the calibration of our system for the case of $N_{RF} = 20$. We seek the set

of parameters $\{\alpha_{QE}, \beta_{QE}, N_{QE}, N_{QE,irr}, N_{rerank}, b, k\}$ that give the best residue MAP. Fig.3.3 represents the calibration of these parameters. In all the plots, the residue MAP values are averages over 50 queries of TREC-2005. In each of the plots Fig.3.3, only the parameters shown in the X-axis are varied, while the remaining parameters are fixed at the values indicated in the figures. Note that because of the large number of parameters (Table 3.1) and calibration time considerations, we do not perform an exhaustive grid search of the globally optimal set of parameters. Rather, as indicated in Fig.3.3, we seek local optimal values of each parameter, within the ranges shown in the figure. From the plots, we obtain the following best set of the parameters $\{$ $\alpha_{QE} = 0.20$, $\beta_{QE} = 0.85$, $N_{QE} = 240$, $N_{QE,irr} = 160$, $N_{rerank} = 160000$, $k = 2.5$, $b = 0.75\}$.

According to Eq.(3.20), when the value of the parameter β_{QE} is less than 1.0, the query expansion vector $\vec{q}_{QE\_RF}$ contains contribution of terms extracted from known irrelevant documents. Fig. 3.3(b) suggests that averaging over 50 queries, the residue MAP obtained with $\beta_{QE} = 0.85$ is slightly better than the value obtained with $\beta_{QE} = 1.0$. The results suggest that there is some benefit to include negatively weighted terms in relevance feedback. However, previous work by Dunlop [19] and Wong et al. [89] found that negative query expansion does not always give good performance. Therefore, we have compared the performance for $\beta_{QE} = 1.0$ and $\beta_{QE} < 1.0$ in more detail.

Fig.3.4 shows the results of a trial experiment in which we analyse the difference between MAP values obtained for QE with $\beta_{QE} = 1.0$ and $\beta_{QE} = 0.8$, with all the remaining parameters being equal. We use the notation $N_R@20$ to denote the number of relevant documents among the top 20 documents returned by PRF, i.e. the number

of known relevant documents for RF. Fig.3.4 shows the values of MAP($\beta_{QE} = 1.0$)-MAP($\beta_{QE} = 0.8$), averaged over queries with the same value of N_R@20 for TREC-2005, plotted against N_R@20. The figure illustrates the following points: (1) While the calibration plot Fig.3.3(b) shows that averaging over 50 queries, $\beta_{QE} = 0.8$ gives slightly better MAP than $\beta_{QE} = 1.0$, there may be some queries for which $\beta_{QE} = 1.0$ gives the better performance. (2) For queries with small values of N_R@20, $\beta_{QE} = 1.0$ tends to be better. In particular, Fig.3.4 shows that for N_R@20=0 and 1, the MAP obtained with $\beta_{QE} = 1.0$ is on average better than that obtained with $\beta_{QE} = 0.8$ by over 0.01. (3) At larger values of N_R@20, $\beta_{QE} = 0.8$ tends to be better. In Fig.3.4, this is particularly apparent for N_R@20=7 and 12.

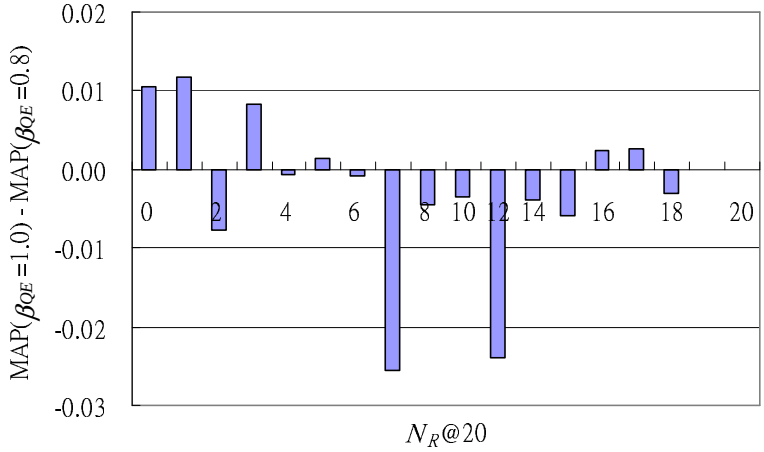

Figure 3.4: Difference between residue MAP values obtained for QE with $\beta_{QE} = 1.0$ and $\beta_{QE} = 0.8$, averaged over queries with the same value of N_R@20 for TREC-2005, plotted against N_R@20.

The above results shown in Fig.3.4 suggest that there may be some benefit in calibrating the set of parameters differently for different queries, depending on the N_R@20 value of the query. Therefore, we have investigated a scheme (called 'Split') in which different sets of parameter values are used for different queries, according to

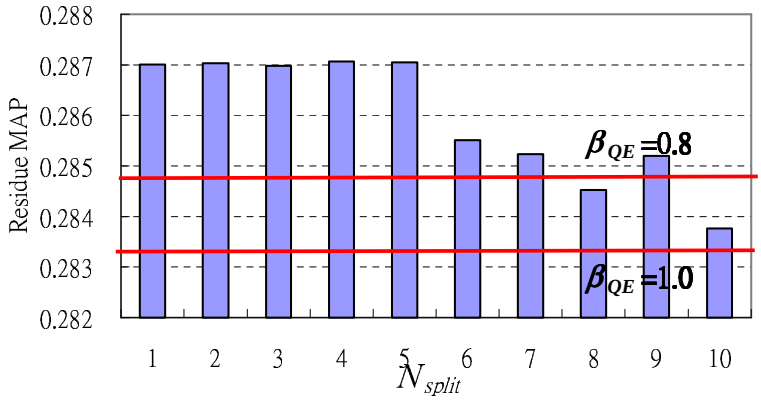

Figure 3.5: Residue MAP averaged over 50 queries of TREC-2005, obtained by setting $\beta_{QE} = 1.0$ for queries with $N_R@20 \leq N_{split}$ and setting $\beta_{QE} = 0.8$ for queries with $N_R@20 > N_{split}$.

the value of $N_R@20$. Specifically, we define a parameter N_{split} such that we use one particular set of parameter values for $N_R@20 \leq N_{split}$, and a different set of parameters when $N_R@20 > N_{split}$. We performed some trial experiments in which we set $\beta_{QE} = 1.0$ for queries with $N_R@20 \leq N_{split}$, but use $\beta_{QE} = 0.8$ for queries with $N_R@20 > N_{split}$. Fig.3.5 shows the residue MAP obtained with this scheme with various values of N_{split}, averaging over the 50 queries of TREC-2005. The MAP values obtained with $\beta_{QE} = 0.8$ and $\beta_{QE} = 1.0$ for all queries are also indicated in Fig.3.5 by two horizontal lines. The figure shows that for some values of N_{split}, the MAP obtained by 'Split' is better than the values obtained with constant β_{QE} for all queries. The result is best for $N_{split} \leq 5$, with the MAP values being similar for N_{split} between 1 and 5. Therefore, we will adopt this scheme in our experiments, setting N_{split} to 3, in order to stay away from the 'cliff' with the rapid decrease in MAP (which occurs at N_{split} = 6 for the case shown in Fig.3.5). Referring to Fig.3.4, setting N_{split} to 3 also seems reasonable, as Fig.3.4 depicts a change in behavior for $N_R@20 > 3$. Hence, we perform new calibrations separately for queries with $N_{split} \leq 3$ and queries with

$N_{split} > 3$. The results are summarized in Table 3.3.

Calibration of B&D parameters

We have performed calibration of B&D on the TREC-2005 collection, for $N_{RF} = 20$. B&D is applied in conjunction with query expansion, as indicated in the flow diagram depicted in Fig.3.2. In addition to the parameters $\{\alpha_{QE}, \beta_{QE}, N_{QE}, N_{QE,irr}, N_{rerank}, b, k\}$ used in the baseline, we also seek the set of parameters $\{\gamma_B, \gamma_D, D, C_B, C_D, C_m\}$ that together give the best MAP. Similar to the baseline calibration, we adopt a 'Split' scheme for B&D. Hence, we perform calibrations separately, for the set of queries with $N_R@20 \leq 3$ and for queries with $N_R@20 > 3$. The calibrations are shown in Fig.3.6 ($N_R \leq 3$) and Fig.3.7 ($N_R \leq 3$). The results are summarized in Table 3.3.

Table 3.3: Summary of the parameters calibrated to obtain the best residue MAP for TREC-2005, using the baseline or B&D models

N_{RF}	Term weight	N_R	α_{QE}	β_{QE}	N_{QE}	$N_{QE,irr}$	N_{rerank} (×1000)	Initial query terms							
								k	b	γ_B	γ_D	D	C_B	C_D	C_m
20	Baseline	≤ 3	.23	1.0	80	200	160	3.5	.45	-		-	-	-	-
		> 3	.17	.85	180	120	120	2.0	.90	-		-	-	-	-
		NS	.20	.85	240	160	160	2.5	.75	-		-	-	-	-
	B&D	≤ 3	.23	1.0	80	120	200	4.0	.65	.125	.06	10.0	41	21	21
		> 3	.25	.9	180	100	120	4.0	.75	.15	.07	12.0	21	11	51
10	Baseline	≤ 3	.25	1.0	120	200	160	2.5	.35	-		-	-	-	-
		> 3	.17	.9	180	140	160	2.0	.9	-		-	-	-	-
		NS	.17	.85	180	200	160	2.0	.65						
	B&D	≤ 3	.275	1.0	120	200	160	2.0	.35	.275	.06	12.0	61	21	31
		> 3	.25	1.0	140	120	120	3.0	.95	.275	.11	12.0	11	11	31

Note: (1) Corresponding to the 'Split' scheme, two different sets of parameters are calibrated for queries with $N_R <= 3$ and $N_R > 3$, where N_R is the number of relevant documents in the top NRF returned by pseudo-relevance feedback. For the Baseline, calibration is also performed without 'Split', as indicated by 'NS' in the third column. (2) The BM25 $\{k,b\}$ values shown in the table are applied to the initial query terms $\{q_i\}$. For the query expansion (QE) terms, the standard values $\{k,b\}=\{1.2,0.75\}$ are used for all cases (both Baseline and B&D).

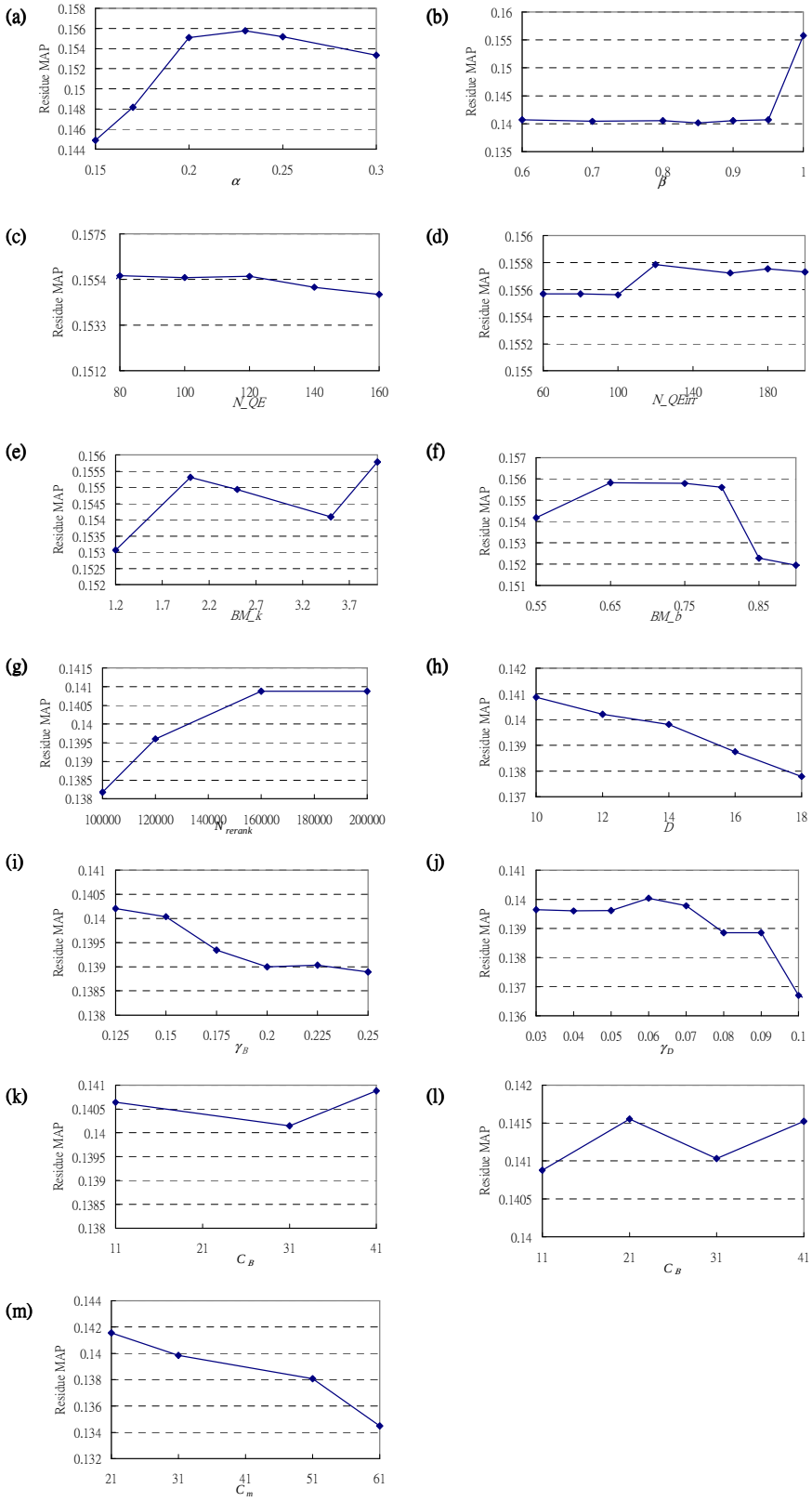

Figure 3.6: Calibration of B&D parameters, for TREC-2005 queries with $N_R@20 \leq 3$.

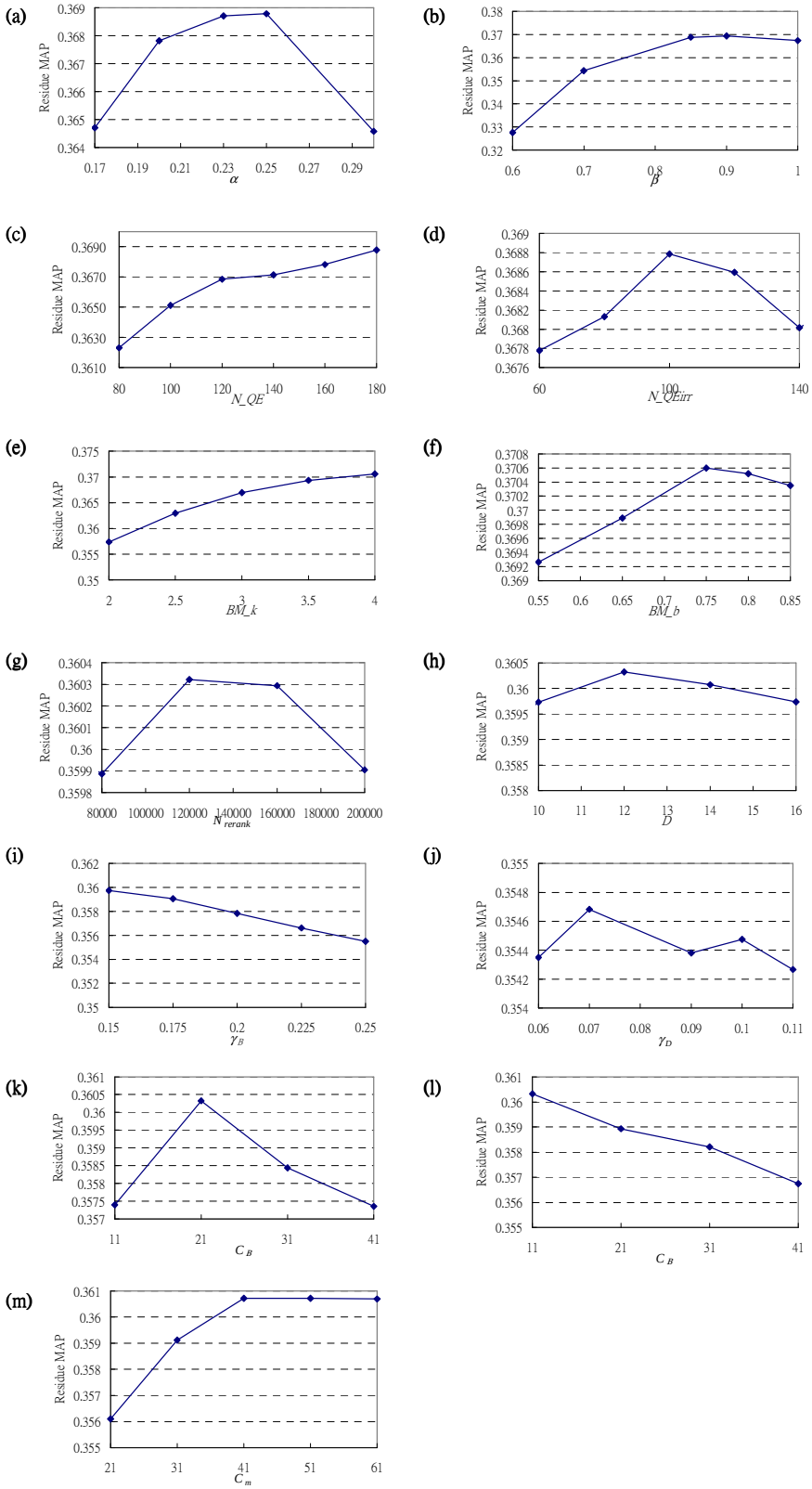

Figure 3.7: Calibration of B&D parameters, for TREC-2005 queries with $N_R@20 > 3$.

3.3.3 Comparison of RF performance using context-dependent term weights vs baseline

Judging top 20 documents of the initial retrieval

We first compare the value of residue MAP obtained using the context-dependent term weights computed by B&D, against the baseline MAP value obtained using the traditional BM25 weights for all terms. Fig. 3.8 shows the difference between the B&D MAP and the baseline (with Split) value, for the 50 title queries of TREC-2005. In the figure, the queries are sorted in increasing order of N_R@20, which is the number of known relevant documents in the relevance feedback with N_{RF} = 20. The figure shows that for most queries, using the context-dependent term weights computed by B&D can yield better MAP values than using the baseline BM25 weights for all terms. The average MAP values are summarized in Table 3.4. For the baseline with standard BM25 term weights, we show in Table 3.4 the MAP values for both with and without Split. As expected, the value obtained using Split is higher than than obtained without Split. For TREC-2005 averaging over 50 queries, the values of residue MAP obtained by the baseline with and without Split are 0.2971 and 0.2957 respectively, while B&D yields a MAP value of 0.3148. Therefore B&D can yield a relative improvement in residue MAP by about 6.0% over the best baseline result. We have checked the statistical significance of the improvement with the Wilcoxon mathched-pairs signed-ranks test. The Wilcoxon p-values indicate that the improvement obtained by B&D is statistically significant at the 99% confidence level (Table 3.4).

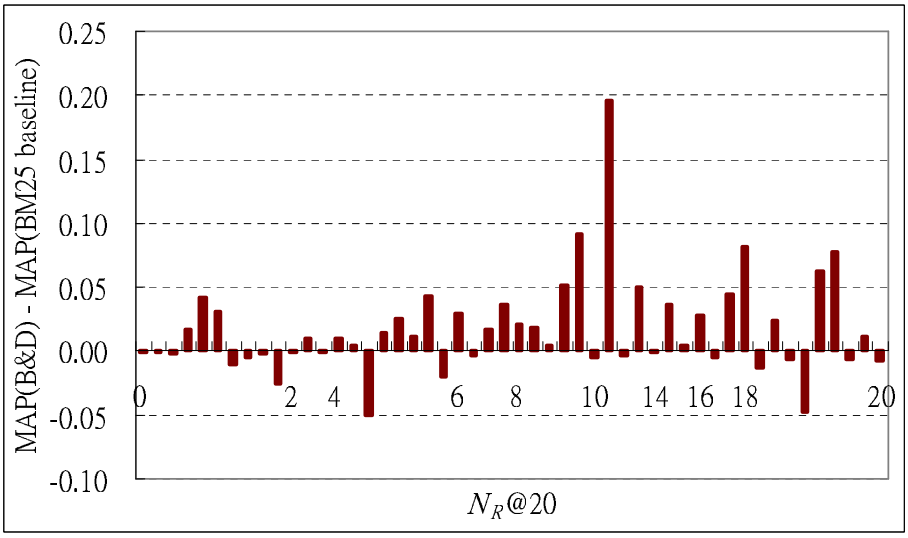

Figure 3.8: Difference of the residue MAP values obtained by B&D and the QE baseline (with Split) for the 50 title queries of TREC 2005. The queries are sorted in increasing order of N_R@20, which is indicated by the X-axis.

Judging top 10 documents of the initial retrieval

The results of our RF experiments presented above show that for making relevance judgments on the top 20 document returned by PRF (i.e. $N_{RF} = 20$), B&D performs better than the QE baseline. It is also of interest to compare the performance based on a smaller number of relevance judgments. This is because it is more realistic for a user to make relevance judgments on the top 10 documents rather than the top 20. This also corresponds to typical web-based retrieval systems that return 10 documents in each page of results. Therefore, we have performed further experiments on relevance feedback with $N_{RF} = 10$ for both our QE baseline system and the B&D algorithm. It is expected that for both the baseline and B&D, the best settings for the case of $N_{RF} = 10$ may be different from those obtained for $N_{RF} = 20$. Therefore, we performed new calibrations for TREC-2005 to find the sets of parameters that give the best residue MAP values. The new sets of parameters for the baseline and for B&D

Table 3.4: Summary of the residue MAP values obtained by the baseline and B&D term weights in RF with 20 or 10 relevance judgments, for various TREC collections

N_{RF}	TREC	B&D (Split)	Baseline (No Split)			Baseline (Split)		
			B&D – Baseline(No Split)			B&D – Baseline(Split)		
		MAP	MAP	Increase	*p*-value	MAP	increase	*p*-value
20	2005	**0.3148**	0.2957	0.0191 (6.5%)	0.0000	0.2971	0.0177 (6.0%)	0.0017
	6	**0.2544**	0.2463	0.0081 (3.3%)	0.0495	0.2349	0.0195 (8.3%)	0.0037
	7	**0.2302**	0.2115	0.0187 (8.8%)	0.0034	0.2134	0.0168 (7.9%)	0.0085
	8	**0.2790**	0.2548	0.0242 (9.5%)	0.0000	0.2616	0.0174 (6.7%)	0.0016
10	2005	**0.3060**	0.2864	0.0196 (6.8%)	0.0003	0.2917	0.0143 (4.9%)	0.0043
	6	**0.2803**	0.2606	0.0197 (7.6%)	0.0091	0.2433	0.037 (15.2%)	0.0016
	7	**0.2359**	0.2281	0.0078 (3.4%)	0.0381	0.2262	0.0097 (4.3%)	0.0305
	8	**0.2892**	0.2607	0.0285 (10.9%)	0.0000	0.2602	0.029 (11.1%)	0.0000

Note: For 'Split', two different sets of parameters are used for queries with $N_R \leq 3$ and $N_R > 3$, as indicated in Table 3.3. The p-values are obtained with the Wilcoxon Matched-pairs signed-ranks test, shown up to 3 digits.

are also summarized in Table 3.3.

The results of the residue MAP are shown in Table 3.4. Following the procedure employed for $N_{RF} = 20$, for B&D we also tested the Split scheme. Separate calibrations are performed for queries with $N_R@10 \leq 3$ and for queries with $N_R@10 > 3$. For the baseline, we again report the values of residue MAP obtained for both with and without Split. Table 3.4 show that for TREC-2005 with $N_{RF} = 10$, the residue MAP obtained by B&D is 0.3060, which has a relative improvement of 4.9% over the best baseline value (0.2917). For this case, the Wilcoxon p-values indicate that the improvement is statistically significant at the 99% confidence level. It should be re-

minded that the residue MAP values as shown in Table 3.4 cannot be directly compared across different values of N_{RF}, because they are calculated using different residue sets of relevant documents in the whole collection.

Other TREC collections

While our results reported above show that B&D is effective in improving the performance of RF with both $N_{RF} = 10$ and $N_{RF} = 20$ for TREC-2005, we wish to confirm whether this improvement is also observed for other TREC collections. Hence we have also performed the experiments using the TREC-6, 7 and 8 collections. In order to claim that the improvement due to B&D is collection-independent, we need to show that the improvement can be obtained using the same set of parameters as found for TREC-2005, without any further calibrations. Therefore, our experiments for TREC-6, 7 and 8 are carried out using the parameters shown in Table 3.3. Furthermore, we have used the Split scheme as described above. Because the calibrations are performed for TREC-2005, the best baseline MAP values obtained using Split is bound to be better than the value obtained without Split. The reason is that the calibration without Split is a subset of the Split scheme. However, as we apply the same sets of calibrated parameters to TREC-6, 7 and 8, it is not guaranteed that the Split sets of parameters will yield better MAP than the 'No Split' set of parameters. Therefore, as a tougher condition to demonstrate the effectiveness of the B&D term weights over the baseline, for the baseline we obtain MAP values for both the Split and 'No Split' schemes. We set the requirement that the B&D MAP value must be better than the higher of the MAP values, with statistical significance at the 95% confidence level, in order to claim

the effectiveness of the B&D term weights.

The results of the RF experiments are included in Table 3.4. The residue MAP values reported for TREC-7 and TREC-8 are averages over the respective 50 title queries of these collections. For TREC-6, there is one query (Q348) that has 5 relevant documents in the whole collection. We found that for this query, all the 5 relevant documents are retrieved within the top 10 documents returned by PRF. Hence, when $N_{RF} = 10$, there is no relevant document in the residue collection. Therefore, in this case (TREC-6, $N_{RF} = 10$) the residue MAP values shown in Table 3.4 are averages over 49 queries. Similarly, another query (Q312) has 11 relevant documents in the whole collection, and all of these are retrieved within the top 20 documents of PRF. Therefore, for $N_{RF} = 20$, the residue MAP values for TREC-6 are averages over 48 queries.

As shown in Table 3.4, in some cases (e.g. TREC-6), the QE baseline obtained without Split is actually better than using Split. However, for all the collections tested and for both 10 and 20 relevance judgments, we found that the residue MAP obtained by B&D is always better than the QE baseline values, whether with or without Split. Overall, the improvement is statistically significant at the 95% confidence level. Therefore, our results have confirmed the effectiveness of context-dependent term weights in RF, both across collections and for different numbers of relevance judgments made.

Chapter 4

Clustering Evaluation

To tackle the word mismatch problem in IR, query expansion in relevance feedback has been found to be an effective solution. Clustering methods are another possible solution. In the next chapter (Chapter 5) we will investigate the use of clustering methods in the determination of context-dependent term weights. Before that, we need to find an effective clustering algorithm for our purpose. This leads to the following research problem – what is an appropriate measure to evaluate the goodness of clustering results? This is the problem that we first consider in this chapter [1].

4.1 Clustering effectiveness measure based on a combination of subclusters

A clustering algorithm separates a collection of objects into groups which are called clusters. When objects are manually assigned to groups, these groups are called classes

[1] This chapter is based on two of our published papers: Dang et al. [12] and [14].

or categories. For example in IR, objects (e.g. documents) may be assigned to one of two classes: relevant or irrelevant. In IR, the ultimate goal is to recover all the relevant documents from a given collection. Ideally, the clustering algorithm groups all the relevant and irrelevant documents into two separate clusters. In this ideal scenario, the IR task would be reduced to identifying any one of the relevant documents, as the cluster that it belongs to contains all the remaining relevant documents.

It was pointed out by Dang et al. [12] that there are applications in IR where it is desirable for objects of the same class to be grouped into multiple subclusters. For example, for better presentation of retrieval results, the search engine Vivisimo [43] returns the retrieval results in the form of clusters, which typically correspond to different subtopics of a search query. In this case, more than one cluster may contain information relevant to a user's need. For such applications, a good clustering algorithm should group objects of the same class together, not necessarily all into a single cluster, but into smaller 'high precision' clusters. Such 'tight' clusters where relevant documents are concentrated are illustrated by the clustering result as shown in Fig.4.1. The figure depicts a dendrogram that represents the group-average clustering of the top-40 retrieved documents of one of the TREC-7 queries. The horizontal axis is a dissimilarity scale. The leaf nodes on the right border of the dendrogram correspond to individual documents. The IDs of only the relevant documents are printed in text form at their leaf nodes, but not the IDs of the irrelevant documents. Visually, the relevant documents seem to be concentrated into several groups. Intuitively, we expect that these groups generally correspond to different sub-topics categories [8].

The MK1 measure was introduced by Jardine and van Rijsbergen [35] for the eval-

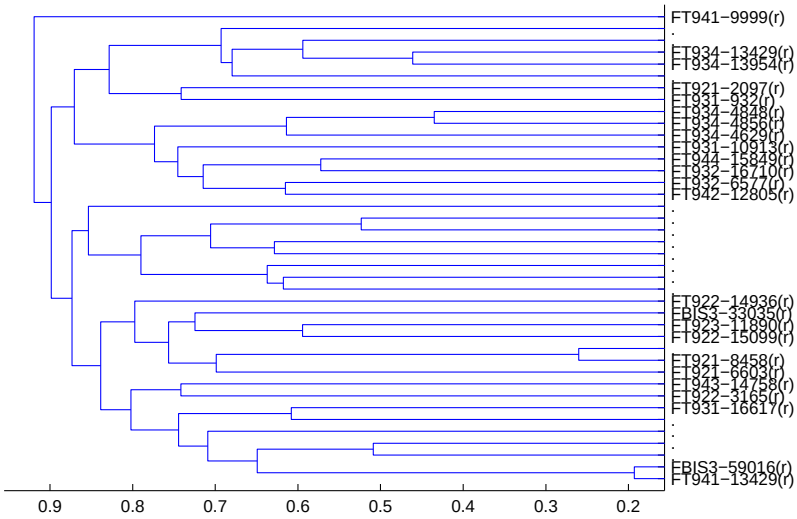

Figure 4.1: Dendrogram representing group-average clustering of the top 40 retrieved documents of TREC-7, query 351 (Falkland petroleum exploration).

uation of clustering results. As MK1 is equal to the E-measure (Eq.2.14) of the single 'optimal cluster' that could be extracted from the clustering results, this measure is appropriate for applications where it is desirable for all relevant documents to be concentrated in a single cluster. For the different type of applications that desire 'tight' high-precision clusters, we have introduced a new class of measures ([12]), called CS (combination of subclusters) that reflects this requirement. This measure is obtained in terms of an optimization problem, whose objective function is the micro-average F-measure, which is introduced in the following (Section 4.1.1).

4.1.1 Micro-average F-measure

The E-measure as defined in Eq.(2.14) applies to binary classes, whereby an object (i.e. document) belongs either to the 'relevant' or the 'irrelevant' class. In order to broaden the applicability of our approach, we first generalize the E-measure to multiple-classes.

This generalization enables our approach to deal with graded relevance [15] in information retrieval, as well as data-mining problems with multiple-classes.

We denote the classes by L_R, where $R = 1, 2, \ldots, c$, with c being the total number of classes. Suppose there are a total of N_R elements of a particular class L_R in a collection of objects. Let $S = \{C_1, C_2, \ldots, C_n\}$ be a given family of clusters of the objects. For a cluster C_i of size $N(C_i)$, let $r(C_i, L_R)$ denote the number of elements in C_i that belong to the class L_R. The precision of the set C_i with respect to the class L_R is $\pi(C_i, L_R) = r(C_i, L_R)/N(C_i)$, while its recall is $\rho(C_i, L_R) = r(C_i, L_R)/N_R$. The E-measure $E(C_i, L_R)$ and F-measure $F(C_i, L_R)$ combine the precision and recall values. The F-measure is defined as:

$$F(C_i, L_R) \stackrel{\text{def}}{=} \frac{(\beta^2 + 1)\pi(C_i, L_R)\rho(C_i, L_R)}{\beta^2\pi(C_i, L_R) + \rho(C_i, L_R)}, \tag{4.1}$$

while $E(C_i, L_R) = 1 - F(C_i, L_R)$. A higher F-measure, and hence lower E-measure, implies a better quality of the cluster [21], with the perfect cluster having $F = 1$. Substituting the expressions of $\pi(C_i, L_R)$ and $\rho(C_i, L_R)$ in Eq.(4.1), we can rewrite the F-measure as:

$$F(C_i, L_R) = \frac{(\beta^2 + 1)r(C_i, L_R)}{\beta^2 N_R + N(C_i)}. \tag{4.2}$$

For each individual cluster in the family S, the F-measure with respect to class L_R may be calculated, and we denote the largest of these values as $F^*(L_R)$. In the context of information retrieval, Jardine and van Rijsbergen [35] defined a performance measure for a clustering algorithm, called MK1, which is equal to 1-F^*, with L_R being

the class of 'relevant documents'. Extending to multiple classes, Larsen and Aone [48] introduced an 'overall F-measure', F_S, which is the sum of the best F-measure for each class weighted according to the class size. Zhao and Karypis [98] called this the FScore measure and it is given by

$$F_S = \sum_{R=1}^{c} \frac{N_R}{N} F^*(L_R), \tag{4.3}$$

where c is the total number of classes. In Eq.(4.3), the sum is over all class labels, and $N = \sum_R N_R$ is the total number of elements in the collection.

A subset of S is specified by a set of indices, $J \subseteq \{1, 2, \ldots, n\}$. For this subset, we can calculate a micro-average F-measure which is defined as the F-measure of the union of all members of the subset. Let C_J denote the union of all the clusters labeled by the indices contained in J: $C_J = \cup_{i \in J} C_i$, $J \subseteq \{1, 2, \ldots, n\}$. In analogy to Eq.(4.2), the micro-average F-measure of C_J is defined as

$$F_\mu(C_i, L_R) \stackrel{\text{def}}{=} \frac{(\beta^2 + 1) r(C_J, L_R)}{\beta^2 N_R + N(C_J)}, \tag{4.4}$$

where $r(C_J, L_R)$ = total number of class L_R elements in C_J, and $N(C_J) = |C_J|$ = number of objects in C_J. The macro-average F-measure may be defined as a simple average of the individual F-measures of the component clusters of C_J:

$$F_M(C_i, L_R) \stackrel{\text{def}}{=} \frac{1}{|J|} \sum_{i \in J} F(C_i, L_R). \tag{4.5}$$

The measure $F^*(L_R)$, or the Fscore measure, Eq.(4.3), are appropriate clustering

effectiveness measures if it is desired that all elements of each individual class L_R are grouped into a single cluster. However, if it is desirable for elements belonging to the same class to be grouped together in 'high precision' clusters, an appropriate measure is the micro-average F-measure $F_\mu(L_R)$. One reason why the macro-average F-measure is not appropriate is that if there are overlapping clusters, it would double count the overlapping items. Also, if an algorithm returns a large number of high precision, but small clusters, $F_M(L_R)$ would be poor because it is limited by the small recall value of each of the clusters. On the other hand, $F_\mu(C_J, L_R)$ does not have this problem. The maximum micro-average F-measure with respect to class L_R is:

$$F^*_\mu(L_R) = \max_{J \subseteq \{1,2,\ldots,n\}} F_\mu(C_J, L_R) = \max_{J \subseteq \{1,2,\ldots,n\}} \frac{(\beta^2+1) r(C_J, L_R)}{\beta^2 N_R + N(C_J)}. \qquad (4.6)$$

Note that while it is obvious that pooling together clusters will yield a better recall than selecting a single cluster, whether a better micro-average E-measure (or alternatively F-measure) can be obtained depends on the presence of multiple high-precision clusters. This is because the E-measure and F-measures are composites of both recall and precision.

Before discussing how to define a new effectiveness measure based on merged clusters, we first review the algorithm for obtaining MK1, which is the E-measure of the single 'optimal cluster'. Such an algorithm (Algorithm 0) is shown in Fig.4.2. Starting from the root, the algorithm steps through all splitting-levels, which are the (dis)similarity levels at which a cluster splits into two children clusters. The E-measure of every cluster in the hierarchy is computed, and the smallest value is returned as

MK1.

Algorithm 0 (MK1)

1 Do hierarchical clustering of N documents
2 $E \leftarrow$ E-measure of the single cluster consisting of all the N documents
3 **for** each of the $(N-1)$ splitting-levels of the hierarchy **do**
4 Compute E-measure, E_i, of each cluster
5 $E \leftarrow \min(E, \min_i E_i)$
6 **endfor**
7 MK1 $\leftarrow E$

Figure 4.2: Algorithm to calculate MK1

In analogy to MK1, we define a class of measure called CS, which equals to the best micro-average E-measure that is attainable by a combination of clusters. For a hierarchical system consisting of N documents, it is expected that the total number of possible combinations is of the order of 2^N and the question arises whether it is possible to find the optimal combination by an efficient algorithm with polynomial time complexity. Otherwise, the practicality of such a measure is questionable. In fact, in designing a new effectiveness measure, we impose a requirement that the time-complexity to compute the new measure should be comparable to computing MK1.

In Section 4.2, we will demonstrate that it is possible to obtain a linear time complexity algorithm for CS if we restrict to seeking the optimal combination of disjoint clusters. As for the general case where clusters are non-disjoint, at the moment we are unable to provide a polynomial time algorithm to solve the optimization problem. Rather, we will present several greedy algorithms to yield estimates of the optimal E-measure, as described in Section 4.3.

4.2 Optimal combination of disjoint clusters – CS1

Instead of seeking the best out of all possible combinations of clusters, we study a sub-class of the problem that restricts the clusters to be disjoint. In particular, we consider the following scheme. First, we form disjoint clusters by cutting a hierarchical structure at one of the splitting-levels (Fig.4.3). Out of all possible subsets of these disjoint clusters, we seek the one that yields the smallest micro-average E-measure. We then step through all the splitting-levels, and for each level we find the cluster combination that gives the smallest micro-average E-measure. We define a measure, called CS1, to be the smallest value among these locally optimal E-measures, stepping through all levels of the hierarchy. The algorithm to obtain CS1 is summarized in Fig.4.4 (Algorithm 1).

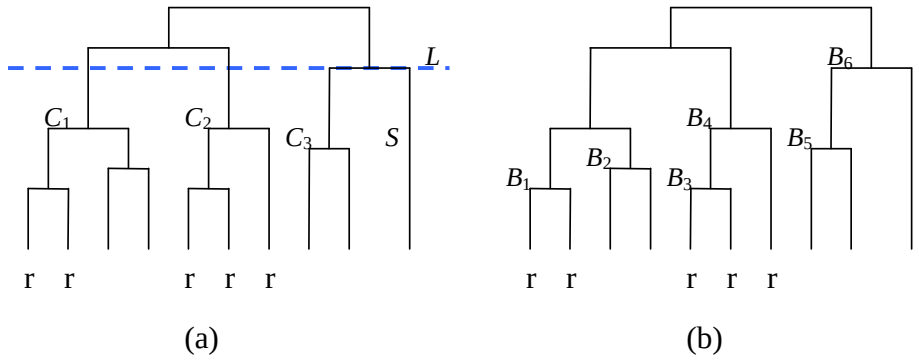

Figure 4.3: Illustration of clusters used in (a) Algorithm 1 and (b) Algorithms 2A, 2B, 2C as applied to the same hierarchical system. Each leaf node is a document, and relevant documents are denoted by ‘r’.

Algorithm 1 (CS1) differs from Algorithm 0 (MK1) in two major aspects. First, in Algorithm 1 (line 4) we discard all the ‘singleton’ clusters. This is necessary because

Algorithm 1 (CS1)

1 Do hierarchical clustering of N documents
2 $E \leftarrow$ E-measure of the single cluster consisting of all the N documents
3 **for** each of the $(N-1)$ splitting-levels of the hierarchy **do**
4 Discard all singleton clusters
5 $E_\mu \leftarrow$ smallest micro-average E-measure among all combinations of the remaining clusters
6 $E \leftarrow \min(E, E_\mu)$
7 **endfor**
8 MK1 $\leftarrow E$

Figure 4.4: Algorithm to calculate CS1

otherwise the globally smallest value of E would be obtained by simply merging all the singleton relevant documents. This would then give the largest possible value of $r(C_J, L_R)$ (equals R_T, the total number of relevant documents among the N documents) in the numerator of Eq.(4.4) and the smallest possible size of a merged cluster containing R_T relevant documents ($N(C_J) = R_T$). Secondly at each level of the hierarchy in Algorithm 1, we look for the optimal combination of clusters that gives the smallest micro-average E-measure, instead of picking out the single cluster C_i that has the smallest value E_i as in Algorithm 0. This step in Algorithm 0 has linear time-complexity, so it is desirable that the corresponding steps in Algorithm 1 also have linear time-complexity. We will demonstrate in Section 4.2.1 how such an algorithm that solves the optimization problem exactly can be found. Fig.4.3(a) shows an example of a hierarchical system of clusters that illustrates Algorithm 1. In this figure, cutting the hierarchical tree at level L yields the disjoint clusters C_1, C_2 and C_3, as well as a singleton S. The singleton S is discarded, and line 5 of the algorithm seeks

among the combinations of C_1, C_2 and C_3 the one that has the smallest micro-average E-measure.

In Algorithm 1, by seeking the smallest micro-average E-measure among all combinations of the clusters and stepping through the hierarchy, we necessarily include among the candidates the single 'optimal cluster' of MK1. The only exception is when MK1 corresponds to the E-measure of a singleton relevant cluster, which is a candidate being excluded by step 4 of Algorithm 1. Hence we can make the following remark:

> **Remark 1.** The value of CS1 is always smaller than or equal to MK1, except when MK1 is attained by selecting a singleton relevant cluster.

4.2.1 Reformulation of the optimization problem

A crucial step in computing CS1 according to Algorithm 1 is finding the combination of clusters that yields the smallest value of the micro-average E-measure. This problem may be solved by a reformulation to a well-known optimization problem, as shown below.

Writing $E = 1 - F$, we may restate our problem as the following. Given a set of clusters $\{C_1, C_2, \ldots, C_m\}$, we seek a subset of these clusters

$$C_J = \bigcup_{i \in J} C_i, \qquad \text{with} J \subseteq U, U \equiv \{1, 2, \ldots, m\} \tag{4.7}$$

that maximizes the objective function

$$F = \frac{(1 + \beta^2) r_J}{\beta^2 card(R) + N_J} \tag{4.8}$$

where $r_J = card(\cup_{i \in J} R_i)$ and $N_J = card(\cup_{i \in J} C_i)$.

We first make the following observation regarding the optimal solution:

> **Observation 1.** The set J which maximizes the objective function F necessarily does not include any cluster C_i which contains only irrelevant document.

This observation may be proved by contradiction. Suppose the cluster C_J is the optimal solution and that it has a component cluster C_i that contains only irrelevant documents. Then, by dropping the cluster C_i from C_J, there would be no decrease in r_J, while N_J is reduced in Eq.(4.8), leading to an increase in F. Hence, the optimization problem may be simplified by discarding all clusters which do not contain any relevant documents. Without loss of generality, we relabel the remaining clusters such that $U = \{1, 2, \ldots, m\}$ is the set of indices of clusters containing at least one relevant document. Note that m in the relabeled set is different from the original value if some clusters have been discarded. Then, the number of relevant documents in cluster C_i must be greater than zero:

$$r_i > 0 \qquad \text{for all } i \in U.$$

Up to now, the statement of the optimization problem given above is quite general and does not say whether the clusters contain any common elements. On the other hand, our current problem stated by Line 6 in Algorithm 1 is a special case whereby all the clusters $\{C_i\}$ are disjoint. This property arises because all the clusters are obtained by cutting the hierarchical tree at a certain similarity level. In this case for the merged

cluster C_J, we have $r_J = \sum_{i \in J} r_i$ and $N_J = \sum_{i \in J} N_i$, where N_i is the total number of documents in each of the component cluster C_i. Introducing the variables $\{x_i\}$, where $1 \leq i \leq m$, given by

$$x_i = \begin{cases} 1 & \text{if } i \in J \\ 0 & \text{if } i \notin J \end{cases} \tag{4.9}$$

the objective function, Eq.(4.8), can be rewritten as

$$F = \frac{(1+\beta^2)\sum_{i=1}^{m} r_i x_i}{\beta^2 card(R) + \sum_{i=1}^{m} N_i x_i}. \tag{4.10}$$

The original optimization can therefore be reformulated as maximizing F in Eq.(4.10), for $x_i \in \{0, 1\}$, which is precisely a unconstrained 0-1 linear hyperbolic (or fractional) programming problem (Hammer and Rudeanu [24], Nagih and Plateau [56], Robillard [64], Hansen et al. [25]). While the general 0-1 linear hyperbolic programming problem with arbitrary coefficients is NP-hard (Hansen et al. [25]), in the instances where the denominator is always positive, exact algorithms have been proposed by various authors. In particular, quadratic time algorithms were given by Hammer and Rudeanu [24] and Robillard [64], while Hansen et al. [25] and Nagih and Plateau [56] provided linear time algorithms. For our present case, all the coefficients in Eq.(4.10) are positive definite, hence the optimization can be solved exactly in linear time.

4.2.2 Experiments on CS1

We have performed experiments to compare the CS1 and MK1 measures. In our experiments, clustering is performed using the hierarchical clustering routines provided

by the open source C Clustering Library of de Hoon et al. [17]. We first describe the experimental environment before presenting the results in detail.

Experimental environment

Our new evaluation measure CS1, just like the traditional MK1, can be applied both to the clustering of an entire document collection and to query-specific clustering. In this thesis, we compare CS1 with MK1 with extensive experiments of query-specific clustering on the TREC-2, -6 and -7 ad hoc test collections. One reason why we have chosen to use query-specific clustering is computational time consideration. This is a concern because of the large size of the test collections (Table 1.1). Another reason is that there are applications of query-specific clustering which involve locating relevant documents in multiple clusters (e.g. Iwayama [33], Leuski [51]), and CS1 would be an appropriate evaluation measure.

Each of the TREC collections comes with 50 topics for which relevance judgment is available. Table 1.1 shows some statistics for these collections. We have chosen the three collections because of their different characteristics. First, the title queries for both TREC-6 and TREC-7 generally consist of three query terms or fewer, which are typical in real-life web-search requests. It is useful to study both TREC-6 and TREC-7 because TREC-6 contains many longer documents. On the other hand, TREC-2 is quite different from the others in that it contains some title queries with more query terms, and there are on average many more relevant documents per query in the collection.

In query-specific clustering, we first perform an initial retrieval for each title query with our search engine. We then apply single-linkage, complete-linkage and group

average clustering algorithms to the top-100 and top-1000 retrieved documents for each query. Based on the cluster hierarchies thus obtained, we compute both MK1 and CS1 according to the algorithms (Algorithm 0 and Algorithm 1) described above.

As mentioned in the previous section, the linear time algorithm of Hansen et al. [25] is applicable to the optimization problem in computing CS1. However, for ease of implementation, we have used the algorithm of Robillard [64] in our study. As shown below, this later algorithm also provides an interpretation of the component clusters that constitute the optimal combination which yields the smallest E-measure. Suppose the clusters are labeled such that the m fractions r_i/N_i are ordered in an non-decreasing order:

$$\frac{r_1}{N_1} \leq \frac{r_1}{N_1} \leq \cdots \leq \frac{r_m}{N_m}. \tag{4.11}$$

Let k be an integer in $\{1, 2, \ldots, m\}$ such that

$$\frac{\sum_{j=k}^{m} r_j}{\beta^2 card(R) + \sum_{j=k}^{m} N_j} < \frac{r_i}{N_i} \qquad \text{for all } i \geq k \tag{4.12}$$

and

$$\frac{\sum_{j=k}^{m} r_j}{\beta^2 card(R) + \sum_{j=k}^{m} N_j} \geq \frac{r_i}{N_i} \qquad \text{for all } i < k. \tag{4.13}$$

Robillard's optimal solution, $x^* \in \{0, 1\}^m$ is given by:

$$x^* = \begin{cases} 1 & \text{if } j \geq k \\ 0 & \text{if } j < k \end{cases} \tag{4.14}$$

and the maximal value is

$$F^* = \frac{(1+\beta^2)\sum_{j=k}^{m} r_j}{\beta^2 card(R) + \sum_{j=k}^{m} N_j}. \tag{4.15}$$

The fraction r_i/N_i is actually the precision of the cluster C_i. Therefore, Robillard's algorithm means that the component clusters in the combination that gives the least E-measure are those that have the highest precision, with Eq.(4.12) and (4.13) providing the 'stopping criterion'. The fraction that appears on the left-hand side of Eq.(4.12) and (4.13) is actually $(1+\beta^2)^{-1}$ times the F-measure of the merged cluster composed of $\{C_k, C_{k+1}, \ldots, C_m\}$. To arrive at the optimal solution, one successively picks the highest precision clusters in a ranked list until $(1+\beta^2)^{-1}$ times the cumulative F-measure of the merged pool is larger than the precision of the next remaining cluster. Hansen et al. [25] made a similar observation in their work on a different optimization problem, and they called it a 'precision-driving optimality'. In our case, the 'precision-driven' solution is somewhat surprising in that it applies to all values of the parameter β, which specifies the relative importance of precision and recall. The effect of β only enters through the stopping criterion Eq.(4.12) and (4.13).

Obviously to calculate MK1 and CS1 as described above, it is necessary to know whether each document in a cluster is relevant or not. This information is provided by TREC's relevance judgments. However, due to the large size of the TREC collections (Table 1.1), it is infeasible to assess the relevance of every document in the corpus. Rather, a method of pooling is used to select documents for relevance judgment (Voorhees [87], Soboroff [75]. The top-ranked documents of each TREC par-

ticipant, up to a specific 'pool depth' (generally set at 100) are picked and merged to form a set of documents that are judged. Consequently it is typical that some of the documents returned by any retrieval system are not judged. The general practice in TREC evaluations is that any document that is not judged is assumed to be irrelevant (Voorhees [87]). This assumption has been under scrutiny in various studies (Keenan et al. [40], Zobel [96]). The work of Keenan et al. [96] indicated that the pool depth (100 documents) used in TREC adequately identifies the relevant documents in the entire collection, at least in the gigabyte regime. Zobel [96] concluded that the results of TREC retrieval experiments were reliable. In the different setting of NTCIR, an evaluation workshop of Japanese text retrieval similar to TREC, Kuriyama et al. [45] also investigated the method of pooling. They verified the effectiveness of pooling in finding relevant documents and also confirmed the reliability of evaluations using the test collection based on pooling. In our experiments, we have also adopted the assumption that all non-judged documents are irrelevant. However, in order to confirm that the assumption does not have a significant impact on our experiments which use the TREC-2, -6 and -7 collections, we have performed some additional experiments in which we discard any retrieved documents that do not have relevance judgment information. The results using the filtered document sets, as summarized in Table 4.2, demonstrate that the assumption does not affect the qualitative conclusions in our current work. Hence this gives us confidence in the calculations in our experiments using the concerned assumption.

Results for $\beta = 1.0$

The results for our experiments, for the case of $\beta = 1.0$, are summarized in Table 4.1. In the table, the values of MK1 and CS1 are averages over 50 queries for each of the TREC collections. In the third column of the data in Table 4.1, we have also included a third quantity, called CS1s. This is the value of CS1 that we would have obtained if 'singleton' clusters were not discarded (line 4 in Algorithm 1). As mentioned before, this value corresponds to a merged cluster containing all the relevant documents alone and no irrelevant documents. Without line 4 in Algorithm 1, no matter what clustering algorithm is used, the ideal cluster would be achieved by picking all the relevant singleton documents at the leaf-nodes level in the algorithm. As this is obviously an over-optimistic scenario, CS1s would not serve as a meaningful evaluation measure, and it is included here only to indicate the lower bound for CS1.

The fourth data column in Table 4.1 is the difference between the measures, Δ=MK1-CS1. We find Δ to be positive for all of the TREC collections used, and for all clustering algorithms, i.e. the average CS1 is always smaller than the corresponding MK1 numerically. To test for the statistical significance of the difference between MK1 and CS1, we performed the Wilcoxon matched-pairs signed-ranks test. The p-values as presented on the last column of Table 4.1 indicate that there is statistical significance at the 99.9% level for almost all cases. In fact, we find that for every individual query, CS1 is generally equal to or smaller than MK1. This can be illustrated by plotting the scatter diagram of CS1 against MK1 for every query. Figures 4.5(a) to (c) show the scatter diagrams for the top-1000 retrieved documents of TREC-7, using three cluster-

ing algorithms. Each point in the figures corresponds to an individual query. As the points generally lie below the 45 degree line, the smaller value of CS1 is confirmed. The plots for the other collections all show this behaviour.

Table 4.1: Evaluation measures for $\beta = 1.0$, averaged over 50 queries for each collection.

		MK1	CS1	CS1s	Δ=MK1-CS1	Δ/MK1 (%)	MK1-CS1 *p*-value
				Trec-2, β=1.0			
	Single linkage	0.753	0.749	0.674	0.004	0.6%	0.0007
top-100	Group average	**0.751**	0.730	0.674	0.021	2.7%	<0.0001
	Complete linkage	0.754	**0.720**	0.674	0.034	4.6%	<0.0001
	Single linkage	0.720	0.623	0.274	0.097	13.4%	<0.0001
top-1000	Group average	**0.677**	0.520	0.274	0.157	23.2%	<0.0001
	Complete linkage	0.707	**0.484**	0.274	0.223	31.5%	<0.0001
				Trec-6, β=1.0			
	Single linkage	0.655	0.634	0.474	0.021	3.2%	0.0001
top-100	Group average	**0.648**	0.599	0.474	0.049	7.6%	<0.0001
	Complete linkage	0.652	**0.586**	0.474	0.066	10.1%	<0.0001
	Single linkage	0.659	0.578	0.238	0.081	12.3%	<0.0001
top-1000	Group average	**0.597**	0.468	0.238	0.129	21.7%	<0.0001
	Complete linkage	0.611	**0.436**	0.238	0.175	28.6%	<0.0001
				Trec-7, β=1.0			
	Single linkage	0.711	0.693	0.556	0.018	2.5%	0.0015
top-100	Group average	**0.707**	0.652	0.556	0.055	7.8%	<0.0001
	Complete linkage	0.719	**0.638**	0.556	0.081	11.2%	<0.0001
	Single linkage	0.728	0.609	0.272	0.119	16.3%	<0.0001
top-1000	Group average	**0.697**	0.504	0.272	0.193	27.7%	<0.0001
	Complete linkage	0.714	**0.473**	0.272	0.241	33.8%	<0.0001

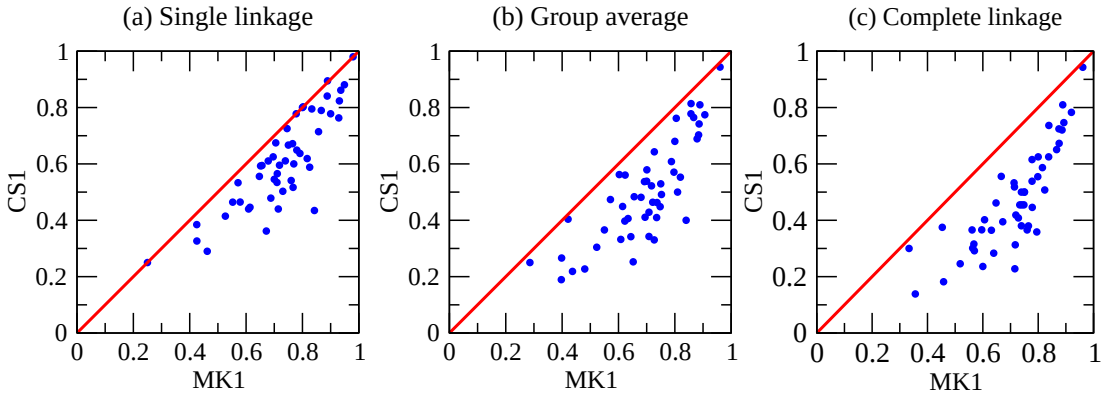

Figure 4.5: Plots of CS1 vs. MK1 for top 1000 retrieved documents for the queries of TREC-7, with $\beta = 1.0$.

Results for filtered sets of retrieved documents that have relevance judgment information

As mentioned in the Experimental environment section, only a subset of documents in each TREC test collection are judged for relevance, and the common assumption is that all non-judged documents are irrelevant. In this section we investigate the impact of this assumption and present some additional experiments in which we discard all retrieved documents that do not have relevance judgment information. The data as summarized in Table 4.2 shows that for the worst case (TREC-6, top-1000 retrieved documents), on average only about 329 documents among the retrieved sets are judged for each query. However, comparing our previous results in Table 4.1 and the results in Table 4.2, we find that the assumption in concern does not have a significant impact. From Table 4.2, after filtering the documents that are not judged, both MK1 and CS1 are found to be improved. This is expected because all the relevant documents are retained in the filtering, while a lot of irrelevant 'noise' documents are removed. However, while the numerical difference between CS1 and MK1 in Table 4.2 is re-

duced compared with Table 4.1, we still observe the average CS1 to be always smaller than MK1. The percentage difference Δ/MK1 is in the range of 0.5% to 10.2% for the top-100 retrieved set, and 7.8% to 31.3% for the top-1000 retrieved set. Furthermore, we have confirmed the statistical significance in all cases by the Wilcoxon test. At $\beta = 1$, group average clustering still gave the best MK1, while complete linkage gave the best CS1, the same as the findings in the Results for $\beta = 1.0$ section. In the rest of this section, our experiments are performed without filtering away the non-judged documents.

Table 4.2: Averaged evaluation measures for $\beta = 1.0$. Retrieved documents that do not have relevance judgment information are discarded.

		Number of documents after filtering	MK1	CS1	Δ=MK1-CS1	Δ/MK1 (%)	MK1-CS1 *p*-value
			Trec-2, β=1.0				
top-100	Single linkage	79.8	0.744	0.741	0.003	0.5%	0.0005
	Group average		**0.743**	0.727	0.016	2.2%	<0.0001
	Complete linkage		0.744	**0.717**	0.028	3.7%	<0.0001
top-1000	Single linkage	396.3	0.593	0.547	0.046	7.8%	<0.0001
	Group average		**0.573**	0.473	0.100	17.5%	<0.0001
	Complete linkage		0.583	**0.445**	0.138	23.7%	<0.0001
			Trec-6, β=1.0				
top-100	Single linkage	77.9	0.649	0.631	0.019	2.9%	0.0002
	Group average		**0.644**	0.596	0.047	7.4%	<0.0001
	Complete linkage		0.645	**0.583**	0.062	9.6%	<0.0001
top-1000	Single linkage	329.4	0.594	0.535	0.058	9.8%	<0.0001
	Group average		**0.547**	0.447	0.100	18.4%	<0.0001
	Complete linkage		0.568	**0.422**	0.145	25.6%	<0.0001
			Trec-7, β=1.0				
top-100	Single linkage	91.2	0.708	0.691	0.017	2.4%	0.0022
	Group average		**0.704**	0.650	0.054	7.7%	<0.0001
	Complete linkage		0.709	**0.637**	0.072	10.2%	<0.0001
top-1000	Single linkage	430.6	0.677	0.591	0.086	12.7%	<0.0001
	Group average		**0.656**	0.488	0.168	25.6%	<0.0001
	Complete linkage		0.673	**0.462**	0.211	31.3%	<0.0001

Results for $\beta = 0.5$ and 2.0

In addition to computing the cluster effectiveness measure MK1 with $\beta = 1.0$, it is common in the literature [35] to use the values $\beta = 0.5$ and 2.0, which represent precision-oriented and recall-oriented retrieval, respectively. We have also studied these regimes, and the experimental results are presented in Table 4.3. For $\beta = 0.5$ and 2.0, we again confirm that the values of CS1 are categorically smaller than the corresponding MK1 values. For clarity, in Fig.4.6 we have plotted the MK1 and CS1 values against β for the top-1000 retrieved documents for TREC-7.

Table 4.3: Evaluation measures for different values of β, averaged over 50 queries.

		Trec-2								
		β=0.5			β=1.0			β=2.0		
		MK1	CS1	Δ/MK1	MK1	CS1	Δ/MK1	MK1	CS1	Δ/MK1
top-100	Single	0.677	0.645	4.7%	0.753	0.749	0.6%	0.782	0.782	0.1%
	Group av.	0.662	0.609	7.9%	0.751	0.730	2.7%	0.782	0.776	0.8%
	Complete	0.668	0.593	11.3%	0.754	0.720	4.6%	0.783	0.771	1.6%
top-1000	Single	0.696	0.520	25.2%	0.720	0.623	13.4%	0.653	0.610	6.4%
	Group av.	0.613	0.432	29.5%	0.677	0.520	23.2%	0.635	0.516	18.8%
	Complete	0.654	0.405	38.1%	0.707	0.484	31.5%	0.650	0.475	26.9%

		Trec-6								
		β=0.5			β=1.0			β=2.0		
		MK1	CS1	Δ/MK1	MK1	CS1	Δ/MK1	MK1	CS1	Δ/MK1
top-100	Single	0.571	0.550	3.8%	0.655	0.634	3.2%	0.668	0.650	2.6%
	Group av.	0.549	0.506	7.9%	0.648	0.599	7.6%	0.667	0.622	6.7%
	Complete	0.554	0.490	11.5%	0.652	0.586	10.1%	0.672	0.614	8.6%
top-1000	Single	0.568	0.471	17.1%	0.659	0.578	12.3%	0.654	0.600	8.3%
	Group av.	0.505	0.383	24.2%	0.597	0.468	21.7%	0.612	0.461	24.6%
	Complete	0.496	0.360	27.3%	0.611	0.436	28.6%	0.620	0.423	31.8%

		Trec-7								
		β=0.5			β=1.0			β=2.0		
		MK1	CS1	Δ/MK1	MK1	CS1	Δ/MK1	MK1	CS1	Δ/MK1
top-100	Single	0.647	0.609	6.0%	0.711	0.693	2.5%	0.709	0.702	1.0%
	Group av.	0.630	0.561	10.9%	0.707	0.652	7.8%	0.709	0.682	3.9%
	Complete	0.639	0.539	15.7%	0.719	0.638	11.2%	0.713	0.669	6.1%
top-1000	Single	0.639	0.482	24.5%	0.728	0.609	16.3%	0.705	0.625	11.4%
	Group av.	0.582	0.406	30.2%	0.697	0.504	27.7%	0.679	0.509	25.0%
	Complete	0.612	0.382	37.6%	0.714	0.473	33.8%	0.698	0.467	33.1%

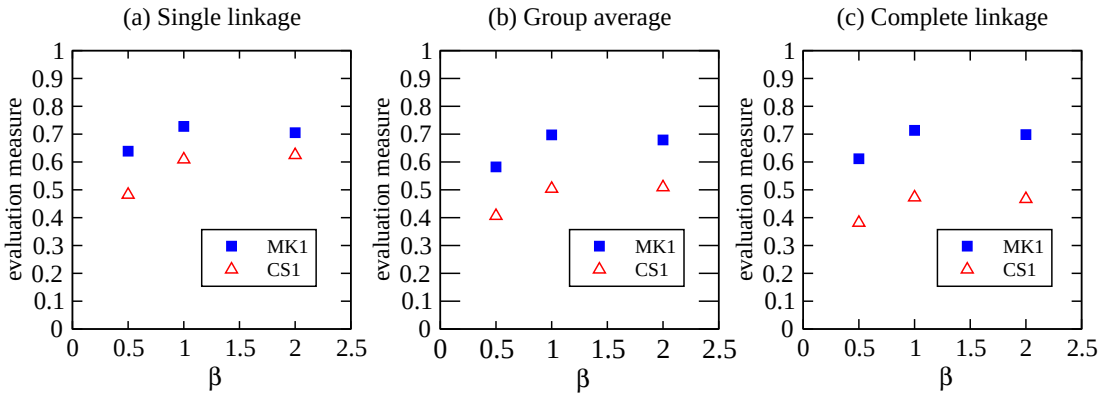

Figure 4.6: Plots of the average values of CS1 and MK1 vs β, for the top 1000 retrieved documents for the queries of TREC-7, corresponding to three types of hierarchical clustering algorithms as indicated

Some statistics of the optimal combinations of clusters

The important implication of smaller values of CS1 compared with MK1 is that a retrieval strategy that combines multiple clusters has some potential of out-performing a strategy that returns a single cluster. Of course, the difficulty is how to identify the appropriate clusters to combine. Therefore we further analyze the characteristics of the optimal solutions which might help in this respect. Some examples of the statistics of the optimal solutions are shown in Table 4.4.

Table 4.4: Statistics of the optimal clusters for TREC-7.

		Trec-7, $\beta = 1$				
		MK1		CS1		
		number of docs in optimal solution	number of relevant docs in optimal solution	number of docs in optimal solution	number of relevant docs in optimal solution	Number of subclusters in the optimal solution
top-100	Single linkage	54.9	22.3	53.0	21.5	1.9
	Group average	54.4	22.6	39.5	22.3	4.6
	Complete linkage	54.4	21.8	34.4	22.5	7.1
top-1000	Single linkage	181.7	32.9	50.2	28.6	7.8
	Group average	79.2	25.4	64.6	40.1	12.8
	Complete linkage	120.2	30.3	67.2	43.3	18.4

Note: All values are averaged over 50 queries. Here, CS1 corresponds to the optimal combination of clusters obtained by cutting a hierarchical tree at a single similarity level.

The data in the table shows that for clustering of the top-100 documents, MK1 and CS1 correspond to roughly the same number of relevant documents, but the total number of documents in the CS1 optimal solution is generally smaller. This means there are fewer irrelevant documents in the CS1 solution, thus giving a higher precision value. Since recall is roughly the same, a higher precision leads to a higher F-measure for the combination of clusters. For clustering of the top-1000 documents, the optimal solution for CS1 generally contains much fewer documents than the MK1 solution. With both complete linkage and group average clustering, the CS1 optimal solution returns a lot more relevant documents than the single cluster of MK1. The last column in Table 4.4 shows the average number of clusters being combined to yield the optimal solution for CS1. For complete linkage clustering of 1000 documents, the optimal solution is made up of about 18 clusters on average. It appears that on average each of these clusters consist of 3 to 4 documents only, indicating that the method picks up small clusters where relevant documents are concentrated, and excludes irrelevant documents as much as possible.

4.3 Optimal combination of overlapping clusters – CS2

As discussed in Section 4.2, our proposed new clustering effectiveness measure, CS1, corresponds to an optimal combination of clusters that are disjoint. For completeness, we will consider the case where some clusters have common elements. This will show how our cluster evaluation approach can be applied to more general problems and clustering algorithms. The corresponding clustering effectiveness measure, applicable for

general overlapping clusters, will be called CS2. Examples where overlapping clusters occur include Suffix Tree Clustering (Zamir and Etzioni [97]) and fuzzy c-Means clustering (e.g. Kummamuru et al. [44]). With hierarchical clustering, members of the family of bottom level clusters may also overlap in a nested fashion (El-Hamdouchi and Willett [20]).

In order to demonstrate our approach for overlapping clusters, we will study the problem of the optimal combination of the bottom level clusters of a hierarchical system as an example (e.g. Fig.4.3(b)). Based on Observation 1 stated in Section 4.2, we can discard any bottom level clusters that do not contain any relevant documents. Suppose there are m bottom level clusters that remain. Our optimization problem is to find the subset of these m clusters that yield the maximum micro-average F-measure, Eq.(4.8). This maximal value is the value of the CS2 measure.

Algorithms

In general for overlapping clusters, we can re-formulate the optimization in a similar manner as for the disjoint case. The difference is that for overlapping clusters, we cannot write r_J and N_J in Eq.(4.8) as simple sums of r_i and N_i over all the component clusters C_i in the merged clusters. As before, we introduce the variables which indicate the presence or absence of cluster C_i in the merged cluster C_J, Eq.(4.9). However, instead of being a fraction of linear functions as in Eq.(4.8), the objective function F in the present case is a non-linear function in $\{x_i\}$. In the most general case, the exact form of F will involve an exponential number of coefficients that specify the number of documents in all the possible intersections of the clusters. Specifically,

these coefficients include:

$$
\begin{aligned}
r_i &= card(R_i) \text{ = number of relevant documents in cluster } C_i \\
r_{ij} &= card(R_i \cap R_j) \\
&\quad \text{= number of common relevant documents in clusters } C_i \text{ and } C_j \\
r_{ijk} &= card(R_i \cap R_j \cap R_k) \\
&\quad \text{= number of common relevant documents in clusters } C_i \text{ , } C_j \text{ and } C_k \\
&\vdots \\
r_{12\ldots m} &= card(R_1 \cap R_2 \cap \ldots \cap R_m)
\end{aligned} \tag{4.16}
$$

and similarly

$$
\begin{aligned}
N_i &= card(C_i) \text{ = number of relevant documents in cluster } C_i \\
N_{ij} &= card(C_i \cap C_j) \\
&\quad \text{= number of common relevant documents in clusters } C_i \text{ and } C_j \\
&\vdots \\
N_{12\ldots m} &= card(C_1 \cap C_2 \cap \ldots \cap C_m)
\end{aligned} \tag{4.17}
$$

Before we discuss the algorithms for solving the optimization problem, we first note a similarity between this problem and the Red-Blue Set Cover (RBSC) problem introduced by Carr et al. [7], which is itself a natural generalization of the well-known set cover problem. In RBSC, there are finite sets of 'red' elements R, 'blue' elements B, and a family S which is a subset of the superset, i.e. $S \subseteq 2^{R \cup B}$. The problem is to find a subfamily which covers all blue elements, but which covers a minimum possible number of red elements. For our case, intuitively the optimal solution that maximizes the objective function F, Eq.(4.8), would contain as many relevant documents as possible and as few irrelevant documents as possible. The similarity with

RBSC is realized by replacing our relevant and irrelevant documents by 'blue' and 'red' elements respectively. However, our problem differs from RBSC in the following. First, RBSC requires complete cover of the blue elements (relevant documents), but our optimization problem does not impose the requirement to completely cover all the relevant documents. Second, the objective function that RBSC seeks to minimize is the number of red elements (irrelevant documents) in the cover. However, in our case the objective function has the more complex fractional form of Eq.(4.8).

In each of Eq.(4.16) and (4.17), there are of the order of 2^m coefficients, where m is the total number of clusters. Note that in the special case of completely disjoint clusters, the only non-zero coefficients are r_i and N_i, and the optimization reduces to a 0-1 linear fractional programming problem as discussed in Section 4.2.1. However in the most general case, because of the exponential number of coefficients, we cannot find a polynomial time algorithm. Instead, we take the standard approach to estimate the optimal solution by greedy algorithms.

Generally, a greedy algorithm is an iterative process in which every iteration takes the step that maximizes some given heuristics. There are several choices of greedy algorithms that are apparent for our problem:

Algorithm 2A. First, because of the similarity of our problem with the set cover, our first algorithm adopts an approach commonly used in the later problem. Specifically, at each iteration we select the cluster C_k that has the largest 'cost effectiveness'. In our case, the cost effectiveness is r_c/N_c, where r_c is the number of relevant documents in the cluster C_k that are not yet in the pool, and N_c is the total number of new documents that would be added to the pool by merging with C_k.

Algorithm 2B. We follow the 'precision-driven optimality' approach of the disjoint cluster problem. At each iteration, the cluster that has the largest precision r_k/N_k is selected. It should be noted that if all clusters are disjoint, Algorithms 2A and 2B are actually equivalent.

Algorithm 2C. Lastly, we consider the natural 'true greedy' algorithm. At each iteration, we select from the set of remaining clusters the one that gives the largest value of the objective function F on merging with C, the current pool.

The reason why we have described several greedy algorithms is that at the moment, we are unable to obtain a performance guarantee for any of them and so we do not have the theoretical knowledge of which one will give the best estimate to the true optimal solution. Therefore, we will compare by experiments the optimal estimates obtained by each of them. One might have thought the 'true greedy' algorithm (2C) would be the best, but our results presented in the next section found that in some cases it actually gives the worst estimate. This shows that for our problem, experiments to try out several algorithms are needed to find a good estimate of the optimal solution.

The three algorithms 2A, 2B and 2C, which estimate the maximum micro-average F-measure attainable by combination of a subset of m bottom level clusters, are summarized in Fig.4.7. It should be noted that these algorithms do not step through the hierarchical levels as in Algorithm 1 (line 3) because we are seeking the optimal combinations of the bottom level clusters. In Fig.4.7, the bottom level clusters have indices $1, \ldots, m$. As an illustration, in Fig.4(b) the bottom level clusters are labeled by $B_1, B_2, \ldots, B_6$. In this example, B_3 is nested within B_4, and B_5 is nested within B_6.

Algorithm 2A (Heuristics: Largest 'cost effectiveness')

1 $C \leftarrow \oslash, J \leftarrow \{1, 2, , \ldots m\}, F \leftarrow 0$

2 **do**

3 find $k \in J$ that maximizes $\frac{\Delta r_c}{\Delta N_c}$, where

$\Delta r_c = card((C \cup C_k) \cap R) - card(C \cap R)$ and

$\Delta N_c = card(C \cup C_k) - card(C \cap R)$

4 $C \leftarrow C \cup C_k, \quad J \leftarrow J - \{k\}$

5 $F \leftarrow \max\left(F, \frac{(1+\beta^2)card(C \cap R)}{\beta^2 card(R) + card(C)}\right)$

6 **while** $J \neq \oslash$

7 $E \leftarrow 1 - F$

Algorithm 2B (Heuristics: Largest precision)

1 $C \leftarrow \oslash, J \leftarrow \{1, 2, , \ldots m\}, F \leftarrow 0$

2 **do**

3 $k \leftarrow \underset{k \in J}{\operatorname{argmax}} \frac{r_k}{N_k}$

4 $C \leftarrow C \cup C_k, \quad J \leftarrow J - \{k\}$

5 $F \leftarrow \max\left(F, \frac{(1+\beta^2)card(C \cap R)}{\beta^2 card(R) + card(C)}\right)$

6 **while** $J \neq \oslash$

7 $E \leftarrow 1 - F$

Algorithm 2C ('True greedy')

1 $C \leftarrow \oslash, J \leftarrow \{1, 2, , \ldots m\}, F \leftarrow 0$

2 **do**

3 find $k \in J$ that maximizes $\frac{\Delta r_c}{\Delta N_c}$, where

$F_C = \frac{card(C \cup C_k) \cap R}{\beta^2 card(R) + card(C \cup C_k)}$

4 $C \leftarrow C \cup C_k, \quad J \leftarrow J - \{k\}$

5 $F \leftarrow \max\left(F, \frac{(1+\beta^2)card(C \cap R)}{\beta^2 card(R) + card(C)}\right)$

6 **while** $J \neq \oslash$

7 $E \leftarrow 1 - F$

Figure 4.7: Greedy approximation algorithms to estimate optimal effectiveness measure for overlapping clusters

4.3.1 Experiments – overlapping clusters

We have applied each of the greedy algorithms (2A, 2B and 2C) to estimate the smallest E-measure obtainable by combining bottom level clusters for the top-100 and top-1000 retrieved documents of TREC-2, -6 and -7 title queries. The families of bottom level clusters are specified by the hierarchical systems studied in the experiments of Section 4.2.2.

We have performed our experiments for $\beta = 1.0$, which means equal importance of recall and precision. The results are summarized in Table 4.5, where we have also included the MK1 and CS1 measures (taken from Table 4.1) as references.

Table 4.5: Estimates of optimal combination of bottom level clusters, using several greedy algorithms (Algorithms 2A, 2B and 2C), averaged over the queries of each TREC collection

		MK1	CS1	Algorithm 2A (Maximize $\Delta n_c/\Delta N_c$)	Algorithm 2B (Maximize n_k/N_k)	Algorithm 2C ('True greedy')	CS1-2A p-value
				Trec-2, β=1.0			
top-100	Single linkage	0.753	0.749	0.739	0.742	0.751	0.0002
	Group average	0.751	0.730	0.706	0.715	0.743	< 0.0001
	Complete linkage	0.754	0.720	0.701	0.708	0.708	< 0.0001
top-1000	Single linkage	0.720	0.623	0.524	0.542	0.643	< 0.0001
	Group average	0.677	0.520	0.418	0.446	0.590	< 0.0001
	Complete linkage	0.707	0.484	0.408	0.431	0.429	< 0.0001
				Trec-6, β=1.0			
top-100	Single linkage	0.655	0.634	0.615	0.621	0.636	< 0.0001
	Group average	0.648	0.599	0.562	0.575	0.601	< 0.0001
	Complete linkage	0.652	0.586	0.553	0.562	0.560	< 0.0001
top-1000	Single linkage	0.659	0.578	0.506	0.519	0.567	< 0.0001
	Group average	0.597	0.468	0.388	0.415	0.466	< 0.0001
	Complete linkage	0.611	0.436	0.377	0.396	0.392	< 0.0001
				Trec-7, β=1.0			
top-100	Single linkage	0.711	0.693	0.673	0.677	0.694	< 0.0001
	Group average	0.707	0.652	0.621	0.631	0.686	< 0.0001
	Complete linkage	0.719	0.638	0.612	0.621	0.617	< 0.0001
top-1000	Single linkage	0.728	0.609	0.523	0.538	0.612	< 0.0001
	Group average	0.697	0.504	0.417	0.441	0.548	< 0.0001
	Complete linkage	0.714	0.473	0.407	0.431	0.420	< 0.0001

Comparing the experimental optimal estimates obtained by Algorithms 2A,2B and

2C, it is found that for all TREC collections and for all clustering methods, Algorithm 2A, which selects clusters based on 'cost effectiveness' at each iteration, always produces the smallest estimate of the E-measure for the optimal combination of bottom level clusters. This can be seen by the average values shown in Table 4.5. Furthermore, we have carried out the Wilcoxon matched-pairs signed-ranks test and confirmed the statistical significance of this finding, with p-values generally far below 0.0001. On the other hand, for both single linkage and group average clustering, the data shows a somewhat surprising result that the 'true greedy' algorithm (2C) actually gives the largest, hence worst, estimate. Only for complete linkage clustering does the 'true greedy' algorithm give comparable estimates as the 'precision-driven' algorithm (2B). An explanation for the poor performance of the 'true greedy' algorithm is that at the early iterations, it tends to select the larger low-precision clusters containing more relevant documents and hence a larger recall. Such clusters may well yield a larger F-measure than some of the other high-precision but small clusters containing only two or three documents. In this way, the algorithm locks many irrelevant documents in the pool, preventing a low eventual E-measure. However, Algorithms 2A and 2B favour the tight high-precision clusters at each iteration, thus keeping the irrelevant documents out of the pool.

We emphasize that it is only for our present optimization problem that we observe the best greedy algorithm to be the one based on selecting the best 'cost effectiveness' cluster at each iteration. In particular, our problem is a special example of overlapping clusters where the bottom level clusters overlap in a nested fashion. Since other greedy algorithms may prevail for other families of overlapping clusters, we have to perform

experiments to obtain the estimates for all the algorithms discussed here. Furthermore, it is interesting to study how far our best estimate is from the true optimal value. Future research could include seeking a bound of the estimates for the algorithms, to indicate how far the estimates are from the true optimal value. Another action may be to seek the true optimal solution by enumeration and compare with our greedy estimates. Of course, the number of combinations of clusters is exponential, so this could only be attempted for queries that do not have too many, say within 20 or 30, relevant documents.

From Table 4.5, it is observed that our estimates of the best E-measure obtainable by combining bottom level clusters as obtained by Algorithm 2A are generally smaller than CS1. We have also confirmed the statistical significance of this difference, as indicated by the Wilcoxon p-values given on the last column of Table 4.5. The reason for the better results of Algorithm 2A may be understood by referring to the example of Fig.4.3. Algorithm 2A allows combination of clusters corresponding to different similarity levels. In Fig.4.3(b), the optimal combination is clearly B_1 and B_4, which cover all the relevant documents in the system, without including any irrelevant documents. In comparison, Algorithm 1 may either select C_2 in Fig.4.3(a) which yields a smaller recall, or the combined C_1 and C_2 which yields a smaller precision.

4.4 The MMF Problem and Optimality of GAA for Nested Clusters

In this section, we discuss in greater detail the the optimization problem defined by Eq.(4.6), which we will call the Maximum Micro-average F-measure (MMF) problem.

4.4.1 MMF problem and related work

It is clear that for a finite family of clusters, S, a maximal value $F_\mu^*(L_R)$ must exist, corresponding to the maximum value of $F(C_J, L_R)$ among all possible subsets C_J of S. However, it is possible for more than one subset of S to yield the same maximal value $F_\mu^*(L_R)$. In this case, we impose an additional condition:

> **Largest Recall Condition.** If more than one subset of S yield the same maximal value $F_\mu^*(L_R)$, the global optimal solution G of the MMF problem is defined to be the one that contains the largest number of class L_R elements, $r(G, L_R)$, i.e. the one that has the largest recall.

As mentioned in Section 2.4.4, values of β larger than 1 in the E-measure Eq.(2.14) correspond to a recall-oriented regime. For $\beta \to \infty$ in Eq.(4.6), $F_\mu^* \approx \max_J r(C_J, L_R)/N_R$, which means that the objective function is dominated by recall. The optimal value F_μ^* will then be attained by a subset that covers all the class L_R elements in the collection, so that $r(C_J), L_R$ reaches the largest possible value N_R. Therefore when β is

sufficiently large, Eq.(4.6) becomes

$$F_{\mu}^{*}(L_R) = \max_{C_J} \frac{(\beta^2+1)N_R}{\beta^2 N_R + N(C_J)} \tag{4.18}$$

where C_J is constrained to cover all the class L_R elements in the collection. This is the 'precision at fixed recall of 1' problem considered by Gao and Ester [21]. Since N_R is a constant, the maximization, Eq.(4.18) is equivalent to minimizing $N(C_J)$. Our MMF problem is then reduced to finding the subfamily C_J that covers all L_R elements in the collection but includes the minimum possible number of elements belonging to other classes. This is a multiple-class generalization of the Red-Blue Set Cover (RBSC) problem introduced by Carr et al. [7] who considered binary class ('Red'/'Blue') elements. RBSC is in turn an extension of the classical set cover problem.

In Section 4.3, we described several greedy algorithms to provide estimates to the MMF problem, Eq.(4.6). Our experiments for various hierarchical clustering methods showed that Algorithm 2A, which is based on a 'cost-effectiveness' heuristics, yields numerically better estimates than the other greedy algorithms. Hierarchical clustering algorithms generate clusters that overlap only by nesting (e.g. Fig.4.3). This means any pair of the clusters may either be disjoint, or in case they intersect, one of them is a proper subset of the other. In this section, we study further the MMF optimization problem for the sub-class of clustering algorithms which generate clusters that overlap only by nesting. First, we make a slight modification to Algorithm 2A by including explicitly a stopping criterion in the iterative process. The modified algorithm (GAA) is shown in Fig.4.8. We make the important claim that for clusters that overlap only

by nesting, GAA yields the true global optimal value. We will provide a mathematical proof of this claim. Our result has practical significance because hierarchical clustering is commonly used [35],[11],[88],[81]. In fact, our result also applies to a family of disjoint clusters, as these may be regarded as a special case of 'overlap only by nesting' whereby none of the clusters actually intersect. This means the results are applicable to partitional clustering algorithms, e.g. K-means, as well.

For a given family of clusters $S = \{C_1, C_2, \ldots, C_n\}$, we seek an optimal subset of S that yield the maximal micro-average F-measure, Eq.4.6. In the rest of this section, we will drop the factor $(\beta^2 + 1)$ in the numerator of the objective function which does not affect the maximization. For further simplification of the notation, we will also hide the label L_R, with the understanding that the derivations presented below can be applied to each individual class label L_R. Accordingly, we denote the constant factor $\beta^2 N_R$ in the denominator by a constant A. Hence, our problem becomes seeking the following maximal value:

$$F_\mu^* = \max_{J \subseteq \{1,2,\ldots,n\}} \frac{r(C_J)}{A + N(C_J)} = \max_{J \subseteq \{1,2,\ldots,n\}} \frac{r\left(\bigcup_{i \in J} C_i\right)}{A + N\left(\bigcup_{i \in J} C_i\right)}. \tag{4.19}$$

In Line 5 of GAA (Fig.4.8), the value $\Delta_H r(C_i)$ is equal to the number of *new* class L_R elements that the cluster C_i would add to the current pool H, while $\Delta_H N(C_i)$ is equal to the total number of *new* elements that would be added. Thus, the ratio $\Delta_H r(C_i)/\Delta_H r(C_i)$ can be regarded as a 'cost effectiveness' measure of adding the cluster C_i to the pool. Line 5 means that at each iteration, among the remaining clusters the one with the largest cost effectiveness is selected as a candidate to be added to the

Algorithm GAA

Input: Clusters C_i, $i = 1, 2, \ldots, n$

Output: Optimal value F_μ^*; Indices of component sets of the optimal solution J^*.

1 $H \leftarrow \oslash, F \leftarrow 0, J^* \leftarrow \oslash$

2 $J \leftarrow \{1, 2, \ldots, n\}$

3 *stop* $\leftarrow$ *false*

4 **do**

5 $k \leftarrow \underset{i \in J}{\operatorname{argmax}} \left[\dfrac{\Delta_H r(C_i)}{\Delta_H N(C_i)} \right]$, where
$r(C_i)$ = number of class L_R elements in C_i
$N(C_i)$ = number of elements in C_i
$\Delta_H r(C_i) = r(H \cup C_i) - r(H)$
$\Delta_H N(C_i) = N(H \cup C_i) - N(H)$

6 **if** $\frac{\Delta_H r(C_i)}{\Delta_H N(C_i)} \geq F(H)$

7 **then** $H \leftarrow H \cup C_k$

8 $F \leftarrow \frac{r(H)}{A + N(H)}$

9 $J \leftarrow J - k$, $J^* \leftarrow J^* + k$

10 **else** *stop* $\leftarrow$ *true*

11 **while** $J \neq \oslash$ **AND** *stop* = *false*

12 $F_\mu^* \leftarrow F$

Figure 4.8: Greedy approximation algorithm (GAA) for MMF problem

current pool. If the cost effectiveness of this candidate is numerically larger than or equal to the F-measure of the current pool, then it is added to the pool. Otherwise, the iteration stops. This criterion is stated in Line 6 of the algorithm. In the rest of this section, we discuss the application of GAA to families of clusters in which any overlapping of the clusters occur only by nesting (e.g. hierarchical clustering in

Fig.(4.3)). Formally, we define the instance in which the clusters overlap only by nesting as follows.

Definition 1. *The clusters* $\{C_1, C_2, \ldots, C_n\}$ *'overlap only by nesting' iff* $\forall C_i \forall C_j (C_i \cap C_j \neq \emptyset) \Rightarrow (C_i \subseteq C_j) \vee (C_j \subseteq C_i)$.

The above definition means that if two clusters intersect, then one of them must be a subset of the other. It is obvious that the following proposition holds:

Proposition 1. For a family of clusters that overlap by nesting, the union of any subset of the clusters, C_J, can be written as a union of disjoint clusters.

The main goal of this section is to prove the following theorem.

Theorem 1. *For the instance of the MMF problem where the clusters in the given family overlap only by nesting, the greedy approximation algorithm GAA yields the global maximal value* F_μ^*. *If more than one subfamily of clusters yield the same global maximal value* F_μ^*, *then GAA will return the one that satisfies the Largest Recall Condition.*

Theorem 1 can be proved by mathematical induction. The proof includes: (1) a basis step; (2) an inductive step; and (3) a termination condition corresponding to Line 6 of the GAA algorithm. We will make extensive use of the following Lemma [64],[25]:

Lemma 1. *For two fractions* a_1/b_1 *and* a_2/b_2, *where* $a_i \geq 0$, $b_i > 0$, $i = 1, 2$, *we have* $\min\left[\frac{a_1}{b_1}, \frac{a_2}{b_2}\right] \leq \frac{a_1+a_2}{b_1+b_2} \leq \max\left[\frac{a_1}{b_1}, \frac{a_2}{b_2}\right]$. *Here, the inequalities*

are strict if $(a_1b_2 - a_2b_1) \neq 0$*, while the equalities are strict if* $(a_1b_2 - a_2b_1) = 0$.

The proof of the basic step and inductive step will utilize Lemma 2 as stated below. This lemma involves a numerical comparison of the F-measure of a known subset of the global optimal solution and the largest cost effectiveness among the remaining clusters, which corresponds to the cluster selected in Line 5 of GAA.

Lemma 2. *Given an instance* S *in which the clusters overlap only by nesting, and a subset* H *known to be a proper subset of the global optimal* G *of MMF. If the cluster* $C \in S\backslash H$ *has the largest cost effectiveness* $\Delta_H r(C)/\Delta_H N(C)$ *among the clusters in* $S\backslash H$*, and* $\frac{\Delta_H r(C)}{\Delta_H N(C)} \geq \frac{r(H)}{A+N(H)}$ *, then* C *must also be a subset of* G.

Proof. Since H is known to be a proper subset of G, G must be equal to the union of H and some subset of $S\backslash H$. Let $G = H \cup K$. Then, by Proposition 1, we can write K as a union of a set of disjoint clusters $\{C_i\}$ where $C_i \in S\backslash H$, and $K = C_1 \cup C_2 \cup \cdots \cup C_m$. We will now prove Lemma 2 by contradiction: If we assume that the cluster C is *not* a subset of the optimal solution G, then a contradiction arises. The possible scenarios in which C is not a subset of G are: (i) C and G are disjoint, or (ii) G contains a proper subset of C. In case (ii) we also need to differentiate between three sub-cases: (iia)C and H are disjoint, and (iib)C and H overlap, C and K also overlap, and (iic) C and H overlap, while C and K are disjoint.

Case (i). Since $G = H \cup K$, we can write $r(G)$ and $N(G)$ as $r(G) = r(H) + \Delta_H r(K)$ and $N(G) = N(H) + \Delta_H N(K)$, where $K = C_1 \cup C_2 \cup \cdots \cup C_m$ and $C_i \in S\backslash H$.

Since the clusters C_i are disjoint, $r(K) = r(C_1) + r(C_2) + \cdots + r(C_m)$ and $N(K) = N(C_1) + N(C_2) + \cdots + N(C_m)$. Hence

$$\frac{\Delta_H r(K)}{\Delta_H N(K)} = \frac{\Delta_H r(K) = \Delta_H r(C_1) + \Delta_H r(C_2) + \cdots + \Delta_H r(C_m)}{\Delta_H N(K) = \Delta_H N(C_1) + \Delta_H N(C_2) + \cdots + \Delta_H N(C_m)}. \quad (4.20)$$

Repeated application of Lemma 1 to Eq.(4.20) leads to

$$\frac{\Delta_H r(K)}{\Delta_H N(K)} \leq \max_{i=1,\ldots,m} \frac{\Delta_H r(C_i)}{\Delta_H N(C_i)}. \quad (4.21)$$

Since the cluster C has the largest $\Delta_H r/\Delta_H N$ among all clusters in $S\backslash H$, by Eq.(4.21) we have

$$\frac{\Delta_H r(C)}{\Delta_H N(C)} \geq \frac{\Delta_H r(K)}{\Delta_H N(K)}. \quad (4.22)$$

Since we assume that $G = H \cup K$ is the optimal solution, it is necessary that

$$F(H) \leq F(H \cup K)$$

$$\frac{r(H)}{A + N(H)} \leq \frac{r(H) + \Delta_H r(K)}{A + N(H) + \Delta_H N(K)}. \quad (4.23)$$

By Lemma 1,

$$\min\left[\frac{r(H)}{A + N(H)}, \frac{\Delta_H r(K)}{\Delta_H N(K)}\right] \leq \frac{r(H) + \Delta_H r(K)}{A + N(H) + \Delta_H N(K)} \leq \max\left[\frac{r(H)}{A + N(H)}, \frac{\Delta_H r(K)}{\Delta_H N(K)}\right].$$

Hence, Eq.(4.22) and Eq.(4.23) lead to

$$\frac{r(H)+\Delta_H r(K)}{A+N(H)+\Delta_H N(K)} \leq \frac{\Delta_H r(K)}{\Delta_H N(K)} \leq \frac{\Delta_H r(C)}{\Delta_H N(C)}.$$

Again by Lemma 1,

$$\frac{r(H)+\Delta_H r(K)}{A+N(H)+\Delta_H N(K)} \leq \frac{r(H)+\Delta_H r(K)+\Delta_H r(C)}{A+N(H)+\Delta_H N(K)+\Delta_H N(C)}. \tag{4.24}$$

Since C and G are disjoint, $r(G\cup C) = r(H)+\Delta_H r(K)+\Delta_H r(C)$ and $N(G\cup C) = N(H)+\Delta_H N(K)+\Delta_H N(C)$. Therefore, Eq.(4.24) yields

$$\frac{r(G)}{A+N(G)} \leq \frac{r(G\cup C)}{A+N(G\cup C)},$$

that is $F(G) \leq F(G\cup C)$. If $F(G) < F(G\cup C)$, G cannot be the optimal solution. If $F(G) = F(G\cup C)$, since $G\cup C$ contains more class L_R elements than G, it also means G cannot be the optimal solution that we seek under the Largest Recall Condition, as stated in Section 4.4.1. Therefore in either case a contradiction arises.

The proof that a contradiction occurs also in the cases (iia), (iib) and (iic) is quite similar to the above, and the detail will be omitted here. In all cases, the assumption that C is not a subset of the global optimal solution cannot be true. Therefore, the cluster C with the largest $\Delta_H r/\Delta_H N$ must be a subset of the optimal solution G. □

Lemma 3. *Consider an arbitrary family of clusters S. Given that a subset H is a subset of the global optimal G of MMF, if the cluster $C \in S\backslash H$ has the largest cost effectiveness $\Delta_H r(C)/\Delta_H N(C)$ among the remaining*

clusters in $S\backslash H$, and $\frac{\Delta_H r(C)}{\Delta_H N(C)} < \frac{r(H)}{A+N(H)}$, then none of the clusters in $S\backslash H$ is a subset of G.

Proof. It follows immediately from Lemma 1 that $F(H \cup C) < F(H)$. Therefore C cannot be a subset of the global optimal solution. Since C has the largest cost effectiveness $\Delta_H r(C)/\Delta_H N(C)$ among the remaining clusters in $S\backslash H$, it follows by the same argument that none of the remaining clusters in $S\backslash H$ can be a subset of the global optimal solution. □

We now present the proof of Theorem 1 by mathematical induction.

Proof [Theorem 1: Basis step]. The greedy approximation algorithm (GAA) selects the cluster with the largest $\Delta_H r(C)/\Delta_H N(C)$ in each iteration, where H is the current pool. At the first step, the current pool is the empty set, which must be a subset of the global optimal solution. Furthermore, $F_\mu(\emptyset) = 0$, so for any cluster containing at least one class L_R element, $\Delta_H r(C)/\Delta_H N(C) = r(C)/N(C) > F_\mu(\emptyset)$. Hence, by Lemma 2 the cluster selected by GAA at the first iteration must be a subset of the global optimal.

Proof [Theorem 1: Inductive step]. Assume that at the i^{th} iteration of GAA, all the clusters included in the current pool H are subsets of the optimal solution. Then it readily follows from Lemma 2 that as long as the current pool H is not the optimal solution itself, the cluster selected according to GAA at the $(i+1)^{\text{th}}$ iteration must also be a subset of the optimal solution.

Proof [Theorem 1: Termination condition (Lines 6 – 9 of GAA)]. The termination condition follows immediately from Lemma 3.

We now prove the last statement of Theorem 1. This statement concerns the situation where more than one subset of clusters yield the same maximal value F_μ^*. The statement means that the GAA iterations will continue until the solution with the most L_R elements is obtained.

Proof. By induction, the global maximal value F_μ^* must be reachable by GAA. Suppose there are more than one subset of clusters that yield the same value F_μ^*, and suppose G is the first of these subsets that is reached after some iterations of GAA. Among the remaining clusters in $S \backslash G$, let C be the one that has the largest cost effectiveness $\Delta_G r(C)/\Delta_G N(C)$. If $\Delta_G r(C)/\Delta_G N(C) = F(G) = F_\mu^*$, then $F(G \cup C) = F_\mu^*$ by Lemma 1. Hence, $G \cup C$ also yields the optimal value F_μ^*. Since C satisfies the condition at Line 6 of GAA, it is added to the pool. Because $\Delta_G r(C)$ must be positive, $G \cup C$ must contain more class L_R elements than G. Repeated application of the arguments shows that the final solution reached by GAA is the subset yielding the optimal value F_μ^* that contains the most class L_R elements. □

Finally, we prove the uniqueness of the global optimal solution found, as stated in Theorem 2.

> **Theorem 2.** *For the MMF problem instances where the given clusters overlap only by nesting, there is a unique global optimal solution G under the Largest Recall Condition.*

Proof. This theorem means that there cannot be two distinct subsets G and G' such that they both yield the maximal value, i.e. $F(G) = F(G') = F_\mu^*$, and both have the largest recall. We will prove by contradiction. Assume two such distinct subsets G and

G' exist. Let $G \cap G' = H$. H may be the empty set or a set of clusters. Note that we cannot have $H = G$, which corresponds to G being a proper subset of G'. Otherwise, since G and G' have the same recall, all elements in $G' \backslash G$ must not belong to the class L_R. Then G' must have a poorer precision than G, and so they cannot both yield the maximal F_μ^*. Let $G = H \cup K$ and $G' = H \cup K'$. By Lemma 1,

$$\min\left[\frac{r(H)}{A+N(H)}, \frac{\Delta_H r(K')}{\Delta_H N(K')}\right] \leq \frac{r(H)+\Delta_H r(K')}{A+N(H)+\Delta_H N(K')} \leq \max\left[\frac{r(H)}{A+N(H)}, \frac{\Delta_H r(K')}{\Delta_H N(K')}\right]$$

$$\min\left[\frac{r(H)}{A+N(H)}, \frac{\Delta_H r(K')}{\Delta_H N(K')}\right] \leq F((G') \leq \max\left[\frac{r(H)}{A+N(H)}, \frac{\Delta_H r(K')}{\Delta_H N(K')}\right].$$

Since $F(H) \leq F(G')$, this implies $F(G') \leq \Delta_H r(K')/\Delta_H N(K')$. Because $F(G) = F(G')$, and $\Delta_H r(K')/\Delta_H N(K') = \Delta_G r(K')/\Delta_G N(K')$, it leads to $F(G)$ $\leq \Delta_G r(K')/\ \Delta_G N(K')$. Then again by Lemma 1, $F(G) \leq F(G \cup K')$. Similarly, $F(G') \leq F(G' \cup K)$. Therefore both G and G' cannot be the global optimal solution under the Largest Recall Condition because $\Delta_G r(K')$ and $\Delta_G r(K)$ must be non-zero. The contradiction means that the global optimal solution is unique. □

4.4.2 Worst case time-space complexity of GAA

As discussed in Section 4.4.1, the maximization of micro-average F-measure (MMF) is a generalization of the Red-Blue Set Cover (RBSC) problem [7]. In the limit of sufficiently large β and in the instances in which the clusters contain binary-class ('Red'/ 'Blue') elements, MMF is equivalent to RBSC. RBSC in turn contains as a special case the classical Set Cover (SC) problem [7]. Therefore, MMF is at least as hard as

SC, which is known to be NP-hard. For a family of n sets, there are $2^n - 1$ possible combinations of the sets. Hence solving MMF, Eq.(4.19), by enumeration requires exponential time.

In every iteration of the GAA algorithm for MMF (Fig.4.8), Line 5 selects from among the remaining clusters not yet added to the pool, the one that has the largest 'cost effectiveness' $\Delta_H r(C_i)/\Delta_H N(C_i)$. At the first step, there are n candidate clusters C_i, where $i \in J$ and $J = \{1, 2, \ldots, n\}$ that need to be scanned. At each subsequent iteration, there is one fewer candidate cluster in J. However, the cost effectiveness of all remaining candidate clusters need to be recalculated at Line 5 of every iteration because of the reference to the updated pool H. A maximum of n iterations need to be performed in the algorithm. Therefore, the worst-case time complexity of the algorithm is $O(n^2)$.

For a collection of N objects, the algorithm needs to store the indices of all the class L_R objects, whether or not these have been included in the current pool H. Hence, the space complexity of the algorithm is $O(N)$.

Chapter 5

Clustering for Relevance Feedback

One of the IR problems that we tackle in this thesis is word mismatch. As mentioned in Chapter 2 (Literature Review and Background), query expansion via relevance feedback (RF) is a well established method and is shown to be an effective solution to this problem. Alternatively, clustering methods have been investigated in the past as another possible solution (see Chapter 2 for a review). However, the effectiveness of clustering methods to enhance IR performance has been inconclusive [81]. While some researchers showed that utilizing hierarchical clustering in retrieval was a promising approach (e.g. Griffiths et al. [22]), the results of some others were negative (e.g. El-Hamdouchi and Willett [20]). In the early studies, document clustering was generally performed on the entire corpus, which is an approach called static clustering. More recently, query-specific clustering has been studied by various researchers (e.g. Hearst and Pedersen [29]). Instead of the entire corpus, query-specific clustering is performed on the retrieval results for each query. Tombros and van Rijsbergen [81] showed that query-specific clustering had the potential to increase the retrieval effectiveness com-

pared to both static clustering and conventional document-based retrieval.

In Chapter 3 we studied context-dependent term weights and introduced a Boost and Discount (B&D) procedure which computes these term weights in the setting of relevance feedback (RF). We showed that using the context-dependent term weights enhanced retrieval effectiveness compared with using the traditional BM25, which are a form of context-independent TF-IDF weights. We are motivated to investigate whether augmenting clustering methods to our B&D procedure can further enhance the retrieval effectiveness attained by using context-dependent term weights. Following past research, like that of Tombros et al. [81], we use query-specific clustering in our methods. This chapter presents the findings of our pilot study in this approach.

Our general approach is to use clustering methods to discover more boost and discount terms. A trial experiment (described in Section 5.1) shows that if more boost and discount terms are found (from relevant and irrelevant documents respectively), the retrieval performance can be enhanced. We have thus performed an extensive set of experiments in the RF setting, comprised of various ways of applying clustering to find more relevant contexts apart from those contained in the judged relevant documents. In our experiments, we find that it is generally very hard to achieve further performance improvement over the context-dependent term weight approach of Chapter 3. In particularly, the use of clustering techniques alone is unable to yield performance enhancement compared with not using clustering. One problem is that our clustering methods tend to introduce too much noise to the set of boost and discount terms used by B& D. On the other hand, another well known classification method in machine learning is the support vector machine (SVM). We tested using SVM with B&D, but

our experiments found that it also does not yield performance improvement. Further tests were performed by combining clustering techniques with SVM. We find that a scheme which applies SVM to clustered contexts is able to produce small but statistically significant improvement in MAP, compared with the standard B&D result, in the TREC-2005 test collection.

An outline of the remaining of this chapter is as follows. In Section 5.1 we discuss our general approach in applying clustering techniques in RF. Section 5.2 discusses some aspects of our experimental settings. In Section 5.3 we describe some of the methods that we have tested and the experimental results.

5.1 Clustering approach

As discussed in Chapter 3, in a RF task where N_{RF} documents are judged, from the known relevant and irrelevant documents we extract the sets of boost and discount terms, $S_B(q_i)$ and $S_D(q_i)$. The premise of our approach is that clustering techniques can help to discover more boost and discount terms. The assumption is that if more of these terms are known then the performance of RF using our context-dependent term weights will be enhanced. By using clustering, we try to discover more relevant and irrelevant contexts from which to extract the boost and discount terms. First, we perform a retrieval by RF, with the standard B&D procedure. Then, query-specific clustering is carried out on the combination of contexts [1] of the top ranked N documents of standard B&D retrieved list, together with the contexts of the judged N_{RF}

[1]The reason for performing context-clustering rather than document-clustering is discussed in Section 5.2.

documents. The reason for clustering the top ranked documents of a B&D retrieved list, rather than those of a PRF or baseline QE list, is because the B&D list is expected to contain more relevant documents at the top. This would be helpful since our aim is to discover more unseen relevant contexts [2] [3]. By the cluster hypothesis, the relevant contexts in the unseen documents will form clusters with the contexts of the judged relevant documents, thus allowing them to be identified.

To test the assumption that discovering more boost and discount terms can enhance retrieval effectiveness, we first perform a 'retrospective' experiment, where the set of all the relevant documents for each query are known. Then, from the relevant documents found in the top N (with $N = 80$) of the previously retrieved list, we extract the set of 'retrospective boost terms', $S_{B,retro}(q_i)$. Similarly, from all the irrelevant documents in the top N, we extract the set of 'retrospective discount terms', $S_{D,retro}(q_i)$. These terms are combined with the sets previously extracted from the RF judged documents, i.e. $S_{B,RF}(q_i) \cup S_{B,retro}(q_i)$ and $S_{D,RF}(q_i) \cup S_{D,retro}(q_i)$. We then perform a retrieval with B&D, using the new sets of terms to estimate the probability of relevance of a context (see Section 3.2). In this case, we obtained the MAP to be 0.4124. In comparison, using the sets of terms $S_{B,RF}(q_i)$ and $S_{D,RF}(q_i)$ extracted from the 20 judged documents yields a MAP value 0.3148 (Table 3.4). Hence, if all the relevant and irrelevant documents in just the top 80 documents are identified, this can in principle lead to a relative improvement in MAP by over 30%. Our objective is to test whether

[2] We define 'relevant contexts' to be the contexts contained in relevant documents. Similarly, 'irrelevant contexts' are those found in irrelevant documents.

[3] In this chapter, we use the phrase 'unseen documents' to refer to the documents that are not among the N_{RF} documents judged for relevance in RF. Similarly, 'unseen contexts' are contexts found in the 'unseen documents'.

clustering methods can help to identify these relevant and irrelevant documents.

5.2 Experimental settings

We first discuss several aspects of the basic experiment environment that we have adopted.

A. Context clustering.

We perform clustering of contexts rather than clustering of documents. First, using contexts follows the Query-centric assumption that evidence of relevance or irrelevance is found only within the contexts in a document. Furthermore, clustering contexts, not the whole documents, will avoid the clustering being affected by too many noise terms that appear outside of the contexts. This approach is supported by the work reported by Dang et al. [13], who showed that better clustering results were obtained using context-based similarity scores rather than document-based similarity scores.

B. Clustering algorithm.

Based on the results of Chapter 4, among the various hierarchical clustering algorithms tested, complete linkage gave the best clustering result, in terms of the CS measure (Table 4.3 and Table 4.5). This means that among the algorithms, complete linkage is most effective one in forming tight high-precision clusters. This suits our purpose as we wish to minimize any noise being introduced into the sets of boost and discount terms. Therefore, we will use complete linkage in our experiments.

C. Context similarity.

A necessary input to the clustering algorithm is a measure of similarity between a pair of contexts. Following Dang et al. [13], we use a cosine similarity between the context vectors, $\vec{c_1}(d_1, k_1, m)$ and $\vec{c_2}(d_2, k_2, m)$, in which the term weights are given by a modified BM expression:

$$\mathrm{sim}(\vec{c_1}, \vec{c_2}) = \frac{\vec{c_1} \cdot \vec{c_2}}{|\vec{c_1}| \times |\vec{c_2}|}, \tag{5.1}$$

with the term weights in the context vectors being

$$w(t, c_1(d_1, k_1, m)) = \frac{f(t, d_1)}{f(t, d_1) + \frac{|\vec{d_1}|}{\Delta}} \sqrt{\log \frac{N - df(t) + 0.5}{df(t) + 0.5}}, \tag{5.2}$$

where $|\cdot|$ is the Euclidean length, and Δ is the average Euclidean document length. The IDF factor in Eq.5.2 is a square root of the logarithmic term. This is unlike the standard BM expression [59], which uses the logarithmic factor itself rather than the square root. It was found that using the square root gave better clustering results [13], as the term is multiplied with itself in the cosine calculation. Hence, we will also adopt this form.

D. Pre-clustering relevance judgments

When clustering methods are applied in RF, a consideration is whether the relevance judgments are made before clustering is applied ('pre-clustering relevance judgments') or after clustering is applied ('post-clustering relevance judgments').

Works where 'post-clustering relevance judgments' is adopted include those of Sakai et al. [67] and Lee et al. [50] (see 2.4.3). While the selective sampling method studied by Sakai et al. [67] studied did not find significant improvements on NTCIR collections. However, Lee et al. [50] tested a cluster-based resampling method for PRF and reported that the method was effective for PRF.

In our RF experiments, we have adopted the alternative 'pre-clustering relevance judgments' procedure. Relevant judgments are made on the top N_{RF} documents of the PRF retrieved list as in the standard RF approach (Chapter 3, Fig. 3.2). Subsequently, clustering techniques are applied to the contexts of top-ranked documents in the retrieved list in order to discover more relevant contexts from the unseen documents (see Fig.5.1). The main reason why we use this procedure is to enable the comparison of our retrieval results with the standard B&D results obtained previously (Table 3.4), noting that to have a fair comparison, the residue collections should be the same in two sets of RF experiments. This in turn means that the sets of judged documents in the two sets of RF experiments should be identical. Using the 'post-clustering relevance judgments' procedure will results in a set of judged documents different from those used in Chapter 3, so that the retrieval results cannot be directly compared. Of course, it is also of interest to test the effectiveness of 'post-clustering relevance judgments', but a new baseline needs to be established. We will leave this study for our future research.

Fig.5.1 shows the flow of our clustering approach for RF. In the figure, Step 1 to Step 3 are the standard RF procedures by query expansion and using B&D-computed term weights. These are the steps used in the experiments described in Chapter 3. Step

4 to Step 8 represent the additional application of clustering techniques. In the next Section, we will describe several methods that we have tested for the specific ways in which clustering techniques are used in Step 6 and Step 7.

Clustering for RF

---

1. Make relevance judgments on the top N_{RF} documents returned by PRF
2. Extract contexts, $\{X_J\}$, and boost and discount terms from judged documents
3. Retrieval by QE, with B&D-computed term weights
4. Extract contexts, $\{X_U\}$, from the top N documents of the B&D retrieved list
5. Apply clustering method to the sets of contexts, X_J and X_U
6. Assign an unseen context as relevant (R_{clus}) or irrelevant (I_{clus}) according to the clustering result
7. Extract extra boost terms $S_{B,clus}$ and discount terms $S_{D,clus}$ from the assigned relevant and irrelevant contexts
8. Retrieval by B&D using the expanded sets of boost and discount terms

---

Figure 5.1: Flow of our general approach of applying clustering in RF

5.3 Experimental results

In this section we present the results of various methods that we have tested. All of these methods follow the general procedure according to the steps shown in Fig.5.1. The methods differ in the way clustering techniques are applied to try to discover more boost and discount terms (Step 6 and Step 7 of Fig.5.1). In the current pilot study, the results we present in this section are obtained for the TREC-2005 collection, as our

B&D calibration (Chapter 3) was performed based on this collection.

Method A: Clustering only

As discussed in Section 5.2, we apply complete linkage clustering to the combined set $X_J \cup X_U$ which are respectively the contexts found in the judged N_{RF} documents and those in the top $N = 80$ documents in the standard B&D retrieved list. After forming a hierarchical structure of the contexts, the question is how to select clusters from this structure. If the cluster hypothesis holds, we expect the contexts in the unseen relevant documents to form clusters with those in the judged relevant clusters. Because these relevant contexts may fall into different subtopics, we seek the optimal combination of subclusters as discussed in Chapter 4. In Chapter 4, it was shown that the optimal combination of subclusters corresponds to the set which has the best CS2 value, i.e. the maximal micro-average F-measure. However, one difference between our current problem and the calculations of Chapter 4 should be noted. While the maximal micro-average F-measure F^*_μ (i.e. 1.0-CS2), Eq.4.6 is calculated based on all the known relevant documents in the collection, in our current case only the relevant documents within the N_{RF} judged documents are known.

Let R_J denote the set of judged relevant contexts and $r_J(S)$ denote the number of contexts in the set of combined subclusters, S, that belong to R_J. In analogy to Eq.4.6, we seek the optimal combination of subclusters of contexts that yields the maximal value of the following objective function:

$$F^*_{\mu,R} = \max_S \frac{(\beta^2+1) r_J(S)}{\beta^2 card(R_J) + card(S)}. \tag{5.3}$$

In Eq.5.3, the micro-average F-measure is defined based on the set of judged relevant contexts, R_J, only. The optimal solution of Eq.5.3 can be easily obtained by the GAA algorithm (Fig.4.8). Our expectation is that *unseen* relevant contexts will be found in the optimal solution, S^*, by association with the *judged* relevant contexts. Thus, all unseen contexts found in S^* are assigned as 'relevant' (Step 6 in Fig.5.1). Then, a set of extra boost terms is obtained by extracting from all the unseen contexts that are included in the optimal solution (Step 7 in Fig.5.1).

Likewise, we assume that *unseen* irrelevant contexts are similar to the *judged* irrelevant contexts. Thus, we seek the unseen irrelevant contexts by means of optimizing an analogous function as Eq.5.3, but defined according to the judged *irrelevant* contexts:

$$F^*_{\mu,I} = \max_S \frac{(\beta^2+1)i_J(S)}{\beta^2 card(I_J) + card(S)}, \tag{5.4}$$

where I_J is the set of judged irrelevant contexts and $i_J(S)$ denotes the number of contexts in the set S that belong to I_J. The optimal solution of Eq.5.4 can also be obtained by GAA.

We have performed RF experiments, with 20 relevance judgments (i.e. $N_{RF} = 20$), on the TREC-2005 collection according to the flow shown in Fig.5.1. For the context clustering in Step 4 and Step 5 of Fig.5.1, we set $N = 80$, i.e. we apply clustering to the contexts of the top 80 unseen documents, together with the contexts in the 20 judged documents. In our first experiment (labeled A1), the size of the contexts used in the clustering is set to 61 words. As shown below (see Table 5.1), we find the cluster size does not affect the retrieval results very much. The unseen contexts found in the opti-

mal solutions of Eq.5.3 and Eq.5.4 are assigned as relevant and irrelevant respectively. Then extra boost terms $S_{B,clus}$ and discount terms $S_{D,clus}$ are extracted from these contexts (Step 7 of Fig.5.1). As shown in Table 5.1, we find the RF performance to be quite poor (MAP=0.2417) compared with the standard B&D result (MAP=0.3148). Analysing the sets of unseen contexts being assigned as relevant (R_{clus}) or irrelevant (I_{clus}), we find that these have on average a precision of 0.61 and 0.54 respectively. The poor precision of the (I_{clus}) contexts indicate that the clustering method is not effective in picking out irrelevant contexts among the unseen contexts. This means there is much noise being introduced in the discount terms, which may be the cause for the poor performance of Method A1.

Since a higher precision for the contexts R_{clus} is observed, we then tried a different setting (Method A2), where we only add the extra boost terms $S_{B,clus}$ to our B&D procedure, but uses only the discount terms S_D extracted from the judged irrelevant documents. With Method A2, we obtain MAP=0.3064. Although some extra boost terms are discovered by the clustering method, some of the these are actually noise terms, because the precision of the assigned contexts R_{clus} is less than 1. It appears that detrimental effect of the noise is more than the benefit of finding some extra terms.

We have also tested various context sizes for clustering, but no significant difference is found. For example with a context size of 101 word (Method A3 in Table 5.1), the MAP is 0.3066, close to the value for the context size of 41 word.

Therefore, we find that applying clustering only fails to bring performance improvement in RF, compared to not using clustering.

Because the use of clustering alone (Method A) does not produce performance

Table 5.1: RF (N_{RF}=20) performance with various settings of Method A, averaged over 50 queries of TREC-2005.

	Standard B&D	Method A1 $S_{B,clus}$ and $S_{D,clus}$ Context size = 61	Method A2 $S_{B,clus}$ only Context size = 61	Method A3 $S_{B,clus}$ only Context size = 101
MAP	0.3148	0.2417	0.3064	0.3066
Wilcoxon *p*	-	0.000	0.043	0.033

Note: The Wilcoxon p-value refers to the comparison between standard B&D and the respective methods.

improvement over the standard B&D method, we have investigated using some other common classification method, such as the Support Vector Machine (Method B below), or a combination of clustering with other techniques (Method C).

Method B: SVM only

Support Vector Machine (SVM) is a popular classification method in machine learning (e.g. Burges [4]). Given a set of training samples, each marked as belonging to one of two categories, a SVM training algorithm builds a model that will determine which of the categories a new test sample belongs to. Intuitively, a SVM model represents the samples as points in high dimensional space, mapped so that the samples belonging to the two categories are separated as wide as possible. A hyperplane is then constructed to separate the samples belonging to the two categories. A new test sample is then mapped to the same space, and is classified according to which side of the hyperplane it falls into. In our case, the training samples are the judged relevant and judged irrelevant contexts in RF. SVM may then be used to predict whether an unseen context is relevant or irrelevant.

In our experiments, we have used the SVM^{light} software package, which is an

implementation of SVM by Joachims (e.g. Joachims [36]). For both training and test samples, the size of the contexts is chosen to be 101 words, as in Method A2 and A3 above. Also similar to the experiments for Method A, the test samples are the unseen contexts belonging to the top $N = 80$ unseen documents.

Instead of trying to discover relevant and irrelevant contexts among the unseen samples by means of clustering (Steps 5 and 6 in Fig.5.1), in Method B the unseen contexts are assigned as relevant or irrelevant according to the prediction of the SVM. As for Method A, we have tested two experimental settings — Method B1: Extract both extra boost terms $S_{B,SVM}$ and discount terms $S_{D,SVM}$ from the relevant and irrelevant contexts predicted by SVM; and Method B2: Only add the extra boost terms $S_{B,SVM}$ to the original set S_B.

As summarized in Table 5.2, the MAP values obtained by Method B1 and B2 are 0.2996 and 0.3148 respectively. Hence, just as in Method A, the noise in discount terms introduced in Method B1 harms the RF performance. It happens that the MAP value obtained by Method B2 is actually the same as the standard B&D value (0.3148). On average, there seems to be no advantage in the SVM method compared with standard B&D.

Table 5.2: RF (N_{RF}=20) performance of Method B, averaged over 50 queries of TREC-2005

	Standard B&D	Method B1 $S_{B,SVM}$ and $S_{D,SVM}$ Context size = 101	Method B2 $S_{B,SVM}$ only Context size = 101
MAP	0.3148	0.2996	0.3148
Wilcoxon *p*	-	0.103	0.650

Method C: SVM applied to clustered contexts

Method B involves a single SVM model based on a single set of training samples consisting of all the judged relevant and irrelevant contexts. However, this approach may suffer the following problem. In the set of judged relevant contexts, the contexts may fall into different subtopics. In this case, contexts belonging to the same subtopic are expected to be similar to one another, but may not be similar to those which belong to a different subtopic. In Method B, all the judged relevant contexts are treated as positive training samples, without considering the possibility that these samples falling into different subtopics may actually be quite different. This may cause problems when the SVM model defines the boundary between the positive and negative training samples.

We consider an alternative method (Method C) to take into account the above problem which occurs in Method B. In Method C, the judged relevant and irrelevant contexts are firstly grouped into separate clusters, which would correspond to different subtopics. Suppose we form n_R clusters among the judged relevant contexts, and n_I clusters among the judged irrelevant contexts. Suppose the clusters of judged relevant contexts are denoted by $C_{R,j}$, where $j = 1, 2, \ldots, n_R$. We first define a SVM model, SVM$_{R,1}$ in which the contexts belonging to cluster $C_{R,1}$ are taken as positive training samples, while all the remaining judged relevant contexts, together with the judged irrelevant contexts, are taken as negative training samples. Then, suppose an unseen context c_U receives a score $S_1(c_U)$ according to the prediction of SVM$_{R,1}$. Likewise, for each of the n_R clusters we define a SVM model: SVM$_{R,j}$, with $j = 1, 2, \ldots, n_R$.

Each of these models is used to predict the class of the unseen context c_U, yielding a set of scores $S_{R,j}(c_U)$. Similarly, we define SVM models based on the judged irrelevant contexts, SVM$_I, k$, with $k = 1, 2, \ldots, n_I$. The corresponding scores that each of these models assigns to the unseen context c_U is denoted $S_{I,k}(c_U)$.

In Method C, we wish to define an overall score for the unseen context c_U that can sum up all the scores predicted by the various models, to indicate whether c_U are mostly predicted to belong to the class of a relevant context or an irrelevant context. One such possible overall score is:

$$S(c_U) = \sum_{j=1}^{n_R} S_{R,j}(c_U) - \sum_{k=1}^{n_I} S_{I,k}(c_U). \tag{5.5}$$

If $S(c_U)$ is positive, it means c_U is overall mostly predicted to be a relevant context. On the other hand, if $S(c_U)$ is negative, we assign c_U to be irrelevant.

In Method C, it is necessary to set the parameters n_R and n_I, i.e. the number of clusters formed among the judged relevant and judged irrelevant contexts, respectively. We have considered two schemes:

- Method C1. Fix n_R and n_I to certain constant values;
- Method C2. In defining the context similarity score, we use the cosine similarity Eq.5.1, with one additional condition: If two contexts have fewer than a threshold of $nmatch$ words in common, their similarity is set to zero. Clusters are obtained by cutting the hierarchical tree at the similarity level $Sim = 0$. Increasing the value of $nmatch$ has the effect of separating the contexts into a larger number of smaller clusters, such that the contexts in each cluster have

many words in common.

We have tried various parameter values for both Method C1 and Method C2. For Method C1, we find the best parameter values to be: $n_R = 2$, $n_I = 3$. For Method C2, we find the best parameter is $nmatch = 11$. The corresponding MAP values obtained by these methods are 0.3158 and 0.3157 respectively (Table 5.3). These values are numerically better than the standard B&D values (0.3148), but the differences are not statistically significant at the 95% confidence level. Analysing the results, we plot the query-by-query results of the MAP differences of the these methods and standard B&D, in Fig. 5.2. The plot for Method C1 shows that for queries with small N_R values (say, $N_R \leq 10$) Method C1 can mostly improve MAP over standard B&D. However, for queries with larger N_R (say, $N_R > 10$) the method tends to give poorer results. This may indicate that the values $n_R = 2$ and $n_I = 3$ that we obtained based on averaging over 50 queries may not be appropriate for queries with large N_R. For queries with large N_R, there are many judged relevant contexts, which may fall into many subtopics. Hence, using the small values of the number of clusters, $n_R = 2$ and $n_I = 3$, may not be sufficient for these queries. This suggests there may be a need to use separate calibrations for queries with $N_R \leq 10$ and $N_R > 10$. For Method C2, the plot shows that the difference from standard B&D is quite small, as indicated in the average values being 0.3157 and 0.3148 respectively. We find that the reason for the small difference is because Method C2 generally discover very small numbers of relevant contexts among the unseen contexts, so few extra boost terms are discovered. Nonetheless, the results suggest that it may be beneficial to mix the conidtions of Method C1 and Method C2 for different queries, according to the value of N_R. We

tried this scheme, as shown by Method C3 in Fig.5.2 and Table 5.3. Specifically, for this method, we use the Method C1 conditions for queries with $N_R \leq 10$, and use the Method C2 conditions for queries with $N_R > 10$. The resulting MAP is found to be 0.3197 (Table 5.3). This is a small improvement over standard B&D (relative amount being 1.56%), but the improvement is statistically significant at the 99% confidence level based on the Wilcoxon p-value.

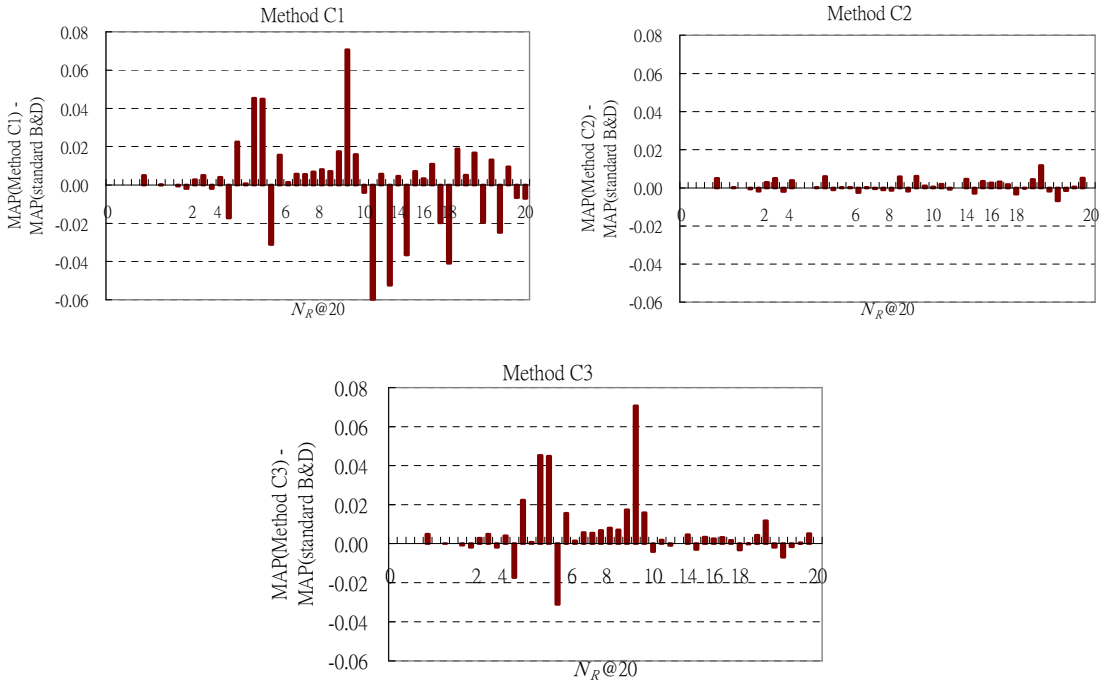

Figure 5.2: Difference of the residue MAP values obtained by various Method C conditions and standard B&D, for the 50 title queries of TREC 2005. The queries are sorted in increasing order of N_R@20, which is indicated by the X-axis.

We have also tested the settings of Method C3 on the TREC-6, and -7 collections. As summarized in Table 5.4 , we find that statistically significant performance improvement over the standard B&D value is not observed in these other collections. It is possible that the specific settings used in Method C3 may not be optimal for the other

Table 5.3: RF (N_{RF}=20) performance with various settings of Method C, averaged over 50 queries of TREC-2005

	Standard B&D	Method C1 n_r=2, n_i=3 $S_{B,clus}$ only Context size = 101	Method C2 *nmatch*=11 $S_{B,clus}$ only Context size = 101	Method C3 Mixture $S_{B,clus}$ only Context size = 101
MAP	0.3148	0.3158	0.3157	0.3197
Wilcoxon *p*	-	0.244	0.064	0.003

Note. Method C3 is a mixture of conditions — For queries with $N_R <= 10$: follows Method C1 conditions; for queries with $N_R > 10$: follows Method C2 conditions.

Table 5.4: MAP values obtained in RF (N_{RF}=20) with Method C3, averaged over 50 queries of various TREC collections

	TREC-2005	TREC-6	TREC-7
standard B&D	0.3148	0.2554	0.2302
Method C3	0.3197	0.2534	0.2310
Wilcoxon p	0.0016	0.762	0.925

collections, i.e. the n_R, n_I, $nmatch$ values, as well as the boundary value of N_R for the mixture of Method C1 and Method C2 conditions. More detailed calibration of the parameters needs to be performed to see whether the performance of these other collections can be improved also. This will be a direction for our ongoing research.

In summary, in our experiments we find that it is generally very hard to utilize clustering methods to achieve performance in RF that is better than our standard B&D results. There seems to be some promising results for the use of clustering in RF in the TREC-2005 collection, for which all the calibrations are performed. However, the effectiveness of the method has not been observed in the other test collections. One question is whether proper calibration of the various parameters can lead to better

performance for all the collections. Alternatively, other schemes utilizing clustering techniques need to be devised and tested. The use of clustering for RF tasks is a topic that is worth further investigation.

Chapter 6

Conclusion and Future Work

We have investigated new methods to tackle the polysemy and word mismatch problems in IR. Among the methods that have been studied in past research, query expansion via relevance feedback (RF) is a well established solution to both of these problems. In addition, Wu et al. [92] showed that using 'document-contexts' to tackle the polysemy problem was a promising direction. As for word mismatch, document clustering has been studied by various researchers in the past, though its effectiveness has been inconclusive. More recently, Tombros et al. [81] suggested query-specific clustering, rather than static clustering of the whole corpus, might be a promising approach. The new methods that we have studied are based on these past results.

This chapter presents a summary of our work and states its main contributions. Some items are also proposed for possible future studies in our ongoing research.

6.1 Summary and contributions

Context-dependent term weights

We have studied context-dependent term weights as a new solution to the polysemy problem. The new weighting of a term t depends not only on the occurrence statistics of the term itself, but also on the evidence of relevance of the document-contexts of the term. A document-context is a fixed size text window in a document. We introduce a Boost and Discount (B&D) procedure to compute the new term weights in the setting of relevance feedback (RF). While the traditional query expansion method in RF utilizes the information of the known relevant or irrelevant documents to add extra terms to the search query, our method utilizes this information to modify the term weights of the original query terms. This work represents the first experimental instantiation of context-dependent term weights that are used for retrieval.

We have performed extensive experiments to evaluate the effectiveness of the new term weights as compared with the context-independent BM25 weights. Our experiments use the title queries of the TREC-6, -7, -8, and 2005 test collections. With either 10 or 20 relevance judgments, we find that using the new term weights yields improvements compared with the baseline BM25 term weights. We find that the new term weights yield relative improvements in MAP over the baseline ranging from 3.3% to 15.2%, with statistical significance at the 95% confidence level across all four test collections.

While we have demonstrated the effectiveness of context-dependent term weights in RF, these new term weights may also be used in other applications such as text cate-

gorization. Because our B&D procedure computes the new term weights by calculating shifts to the widely used BM25 term weights, the method can readily be implemented in systems that use the BM25 weights.

A new clustering evaluation measure

Before we investigate the use of clustering techniques in IR, we need to find an effective clustering algorithm for our purpose. Hence, we have first addressed the issue of how to define an appropriate clustering evaluation measure.

A traditional clustering effectiveness measure is MK1, which is based on finding the single optimal cluster that can be got from the clustering result. We have proposed a new optimal clustering effectiveness measure based on a combination of clusters rather than a single cluster. This new measure, called CS ('Combination of Subclusters'), will reflect more truly the performance of a clustering algorithm for applications where it is desirable for relevant documents to be grouped together in tight and high-precision subclusters, corresponding to different subtopics. For cases when clusters are disjoint, whence the measure is called CS1, we show that the problem becomes a linear fractional 0-1 optimization problem which can be solved by linear time algorithms. Numerically, CS1 is theoretically smaller than or equal to MK1, as we have confirmed experimentally by an implementation of an exact algorithm.

We further discussed how our approach can be generalized to more general problems involving overlapping clusters, whence the new measure is called CS2. We show how optimal estimates of CS2 can be obtained by greedy algorithms. We prove that a greedy algorithm with a 'cost-effectiveness' heuristics yields the global optimal solu-

tion for the class of clustering algorithms which generate either clusters that overlap by nesting within each other, or disjoint clusters that do not overlap. The uniqueness of the optimal solution is also proved. For a family of n clusters containing a total of N objects, this greedy algorithm has a worst case time complexity of $O(n^2)$ and a space complexity of $O(N)$.

Based on our new CS1 or CS2 measures, we find that among the hierarchical clustering algorithms that we tested, complete linkage clustering is the most effective in forming multiple high precision clusters. Therefore we choose to employ complete linkage in our experiments to study the use of clustering in IR.

Clustering for relevance feedback

We have investigated whether augmenting clustering methods to our B&D procedure, which computes context-dependent term weights, can further enhance the performance of RF. The B&D algorithm utilizes information extracted from the judged documents to provide evidence of relevance or non-relevance in the unseen documents. Our general approach is to use clustering methods to discover more boost and discount terms used by B&D.

We have performed an extensive set of experiments in RF, comprised of various ways of applying clustering to find more relevant or irrelevant contexts apart from those contained in the judged documents. In our experiments, we find that it is generally very hard to achieve further performance improvement over the context-dependent term weight approach in which no clustering is applied.

As we do not observe performance improvement by augmenting clustering tech-

niques on their own to the B&D procedure, we have investigated using a combination of clustering and another common classification method, the Support Vector Machine (SVM). We find that a scheme in which SVM is applied to clustered samples of the judged relevant contexts is effective in improving RF performance. With 20 relevance judgments, we show that for TREC2005 a relative improvement of 1.6% in MAP over standard B&D (i.e. without clustering) can be obtained, with statistical significance at the 95% confidence level. However, this improvement is not reproduced on the TREC-6, 7 or 8 collections. Hence, the use of clustering for RF tasks in IR is a topic that is worth further investigation in our ongoing research

6.2 Future research

In this section, we outline several possible items for future research in context-dependent term weights and clustering as applied in IR.

- **Generalize B&D procedure to use n-grams.** In the method described in this chapter, the B&D procedure that compute the context-dependent term weights is based on matching single terms (i.e. unigrams) in the contexts within unseen documents, with the boost and discount terms, which are also unigrams. We will further investigate whether there is any advantage in using n-grams (i.e. bigrams, trigrams, etc.) instead of unigrams.

- **Various relaxations in the B&D procedure.** While we have reported a pilot study of the B&D procedure, various relaxations of the Boost and Discount term selection and weighting may be investigated. In particularly, relaxation of the

Location-Invariance Decision assumption may be considered, such as by assigning a larger weight to terms that occur near to the beginning of a document. Weightings based on the position of occurence of a term within a context can also be studied. Variations in the selection of the B&D terms include imposing criteria based on the number of occurence of the terms in the judged documents (see Section 3.2.1).

- **Compare RF results with other recent methods.** While we have demonstrated the effectiveness of using the new term weights in RF based on query expansion, it is of interest to compare the performance of our method with other RF methods. For example, there are methods based on language models, such as the relevance model of Lavrenko and Croft [49].

- **Post-clustering relevance judgments.** As explained in Section 5.2, in our clustering RF experiments were performed with 'pre-clustering relevance judgments'. It is also of interest to study 'post-clustering relevance judgments' to see whether clustering can enhance RF performance in this setting.

- **Use of other clustering algorithms for RF experiments.** While we have used hierarchical (complete linkage) clustering for our RF experiments reported in this thesis, other clustering algorithms may be used in future studies. Examples of these algorithms include k-means and fuzzy c-means clustering.

Bibliography

[1] BUCKLEY, C., ALLAN, J., SALTON, G., AND SINGHAL, A. Automatic query expansion using smart: Trec 3. In *Proceedings of the Third Text REtrieval Conference (TREC-3)* (1995), pp. 69–80.

[2] BUCKLEY, C., SALTON, G., AND ALLAN, J. The effect of adding relevance information in a relevance feedback environment. In *Proceedings of the 17th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval* (1994), pp. 292–300.

[3] BUCKLEY, C., AND VOORHEES, E. M. Evaluating evaluation measure stability. In *Proceedings of the 23rd Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval* (2000), pp. 33–40.

[4] BURGES, C. J. C. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery 2* (1998), 121–167.

[5] CALLAN, J. P. Passage-based evidence in document retrieval. In *Proceedings of the 17th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval* (1994), pp. 302–310.

[6] CARPINETO, C., AND ROMANO, G. A lattice conceptual clustering system and its application to browsing retrieval. *Machine Learning 24* (1996), 95–122.

[7] CARR, R. D., DODDI, S., KONJEVOD, G., AND MARATHE, M. On the red-blue set cover problem. In *Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms* (2000), pp. 345–353.

[8] CHIK, F. C. Y., LUK, R. W. P., AND CHUNG, K. F. L. Text categorization based on subtopic clusters. In *Proceedings of the 10th International Conference on Applications of Natural Language to Information Systems (NLDB 2005)* (2005), pp. 203–214.

[9] COOPER, W. S., GEY, F. C., AND DABNEY, D. P. Probabilistic retrieval model based on staged logistic regression. In *Proceedings of the 15th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval* (1992), pp. 198–210.

[10] CROFT, W., AND HARPER, D. Using probabilistic models of information retrieval without relevance information. *Journal of Documentation 35*, 4 (1979), 285–295.

[11] CROFT, W. B. A model of cluster searching based on classification. *Information Systems*, 5 (1980), 189–195.

[12] DANG, E. K. F., LUK, R. W. P., HO, K. S., CHAN, S. C. F., AND LEE, D. L. A new measure of clustering effectiveness: Algorithms and experimental studies. *Journal of the American Society for Information Science and Technology 3*, 59 (2008), 390–406.

[13] DANG, E. K. F., LUK, R. W. P., LEE, D. L., HO, K. S., AND CHAN, S. C. F. Query-specific clustering of search results based on document-context similarity scores. In *Proceedings of the 15th Intl. Conf. on Information and Knowledge Management (CIKM'06)* (2006), P. S. Yu, V. Tsotras, E. Fox, and B. Liu, Eds., pp. 886–887.

[14] DANG, E. K. F., LUK, R. W. P., LEE, D. L., HO, K. S., AND CHAN, S. C. F. Optimal combination of nested clusters by a greedy approximation algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence 31*, 11 (2009), 2083–2087.

[15] DANG, E. K. F., WU, H. C., LUK, R. W. P., AND WONG, K. F. Building a framework for the probability ranking principle by a family of expected weighted rank. *ACM Transactions on Information Systems 27*, 4, Article 20 (2009), 1–37.

[16] DAS SARMA, A., GOLLAPUDI, S., AND IEONG, S. Bypass rates: reducing query abandonment using negative inferences. In *Proc. ACM SIGKDD Conf. Knowledge Discovery and Data Mining* (2008), J. Ghosh and D. Lambert, Eds., pp. 177–185.

[17] DE HOON, M. J. L., IMOTO, S., NOLAN, J., AND MIYANO, S. Open source clustering software. *Bioinformatics 20* (2004), 1453–1454.

[18] DEERWESTER, S., DUMAIS, S. T., FURNAS, G. W., LANDAUER, T. K., AND HARSHMAN, R. Indexing by latent semantic analysis. *Journal of the American Society for Information Science 41* (1990), 391–407.

[19] DUNLOP, M. The effect of accessing non-matching documents on relevance feedback. *ACM Transactions on Information Systems 15*, 2 (1997), 137–153.

[20] EL-HAMDOUCHI, A., AND WILLETT, P. Comparison of hierarchical agglomerative clustering methods for document retrieval. *The Computer Journal 3*, 32 (1989), 220–227.

[21] GAO, B. J., AND ESTER, M. Clusters description formats, problems and algorithms. In *SIAM International Conference on Data Mining* (2006), J. Ghosh and D. Lambert, Eds., pp. 462–466.

[22] GRIFFITHS, A., LUCKHURST, H. C., AND WILLETT, P. Using interdocument similarity information in document retrieval systems. *Journal of the American Society for Information Science 1*, 37 (1986), 3–11.

[23] HALKIDI, M., BATISTAKIS, Y., AND VAZIRGIANNIS, M. On clustering validation techniques. *Journal of Intelligent Information Systems 17*, 2 (2001), 107–145.

[24] HAMMER, P. L., AND RUDEANU, S. *Boolean methods in operation research*. Berlin: Springer, 1968.

[25] HANSEN, P., POGGI DE ARAGAO, M. V., AND RIBEIRO, C. C. Hyperbolic 0-1 programming and query optimization in information retrieval. *Mathematical Programming*, 52 (1991), 255–263.

[26] HARMAN, D. Relevance feedback revisited. In *Proceedings of the 15th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval* (1992), E. Fox, N. Belkin, P. Ingwersen, and A. M. Pejtersen, Eds., pp. 1–10.

[27] HARMAN, D. Overview of the first trec conference. In *Proceedings of the 16th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval* (1993), pp. 36–47.

[28] HARTER, S. P. A probabilistic approach to automatic keyword indexing (part 1). *Journal of the American Society for Information Science and Technology 26*, 4 (1975), 197–206.

[29] HEARST, M. A., AND PEDERSEN, J. O. Re-examining the cluster hypothesis: Scatter/gather on retrieval results. In *Proceedings of the 19th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval* (1996), H. P. Frei, D. Harman, P. Schaubie, and R. Wilkinson, Eds., pp. 76–84.

[30] HOPPNER, F., KLAWONN, F., AND KRUSE, R. *Fuzzy Cluster Analysis: Methods for Classification, Data Analysis, and Image Recognition*. New York: Wiley, 1999.

[31] HUBERT, L., AND ARABIE, P. Comparing partitions. *Journal of Classification 2* (1985), 193–218.

[32] IDE, E. New experiments in relevance feedback. In *The SMART Retrieval System*, G. Salton, Ed. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1971, pp. 337–354.

[33] IWAYAMA, M. Relevance feedback with a small number of relevance judgments: Incremental feedback vs. document clustering. In *Proceedings of the 23rd Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval* (2000), pp. 10–16.

[34] JAIN, A. K., DUIN, R. P. W., AND MAO, J. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22* (2000), 4–37.

[35] JARDINE, N., AND VAN RIJSBERGEN, C. J. The use of hierarchical clustering in information retrieval. *Information Storage and Retrieval*, 7 (1971), 217–240.

[36] JOACHIMS, T. Making large-scale svm learning practical. In *Advances in Kernel Methods – Support Vector Learning*, B. Scholkopf, C. Burges, and A. Smola, Eds. MIT Press, Cambridge, MA, 1999, pp. 337–354.

[37] JOACHIMS, T., GRANKA, L., PAN, B., HEMBROKE, H., AND GAY, G. Accurately interpreting clickthrough data as implicit feedback. In *Proceedings of the 28th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval* (2005), pp. 154–161.

[38] JUDD, D., MCKINLEY, P. K., AND JAIN, A. K. Large-scale parallel data clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence 20* (1998), 871–876.

[39] KASZKIEL, M., AND ZOBEL, J. Passage retrieval revisited. In *Proceedings of the 20th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval* (1997), pp. 178–185.

[40] KEENAN, S., SMEATON, A. F., AND KEOGH, G. The effect of pool depth on system evaluation in trec. *Journal of the American Society for Information Science and Technology 7*, 52 (2001), 570–574.

[41] KLEINBAUM, D. G. *Logistic regression: a self-learning text*. New York: Springer, 2002.

[42] KONG, Y. K., LUK, R. W. P., W., L., HO, K. S., AND CHUNG, F. L. Passage-based retrieval based on parameterized fuzzy operators. In *ACM SIGIR Workshop on Mathematical/Formal Methods for Information Retrieval* (2004).

[43] KOSHMAN, S., SPINK, A., AND JANSEN, B. J. Web searching on the vivisimo search engine. *Journal of the American Society for Information Science and Technology 14*, 57 (2006), 1875–1887.

[44] KUMMAMURU, K., DHAWALE, A., AND KRISHNAPURAM, R. Fuzzy co-clustering of documents and keywords. In *Proceedings of the 12th IEEE International Conference on Fuzzy Systems* (2003), pp. 772–777.

[45] KURIYAMA, K., KANDO, N., NOZUE, T., AND EGUCHI, K. Pooling for a large-scale test collection: An analysis of the search results from the 1st ntcir workshop. *Information Retrieval*, 5 (2002), 41–59.

[46] KURLAND, O., AND DOMSHLAK, C. A rank-aggregation approach to searching for optimal query-specific clusters. In *Proceedings of the 31st Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval* (2008), pp. 547–554.

[47] LAFFERTY, J., AND ZHAI, C. Document language models, query models and risk minimization for information retrieval. In *Proceedings of the 24th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval* (2001), pp. 111–119.

[48] LARSEN, B., AND AONE, C. Fast and effective text mining using linear-time document clustering. In *Proceedings of the 5th International Conference on Knowledge Discovery and Data Mining* (1999), pp. 16–22.

[49] LAVRENKO, V., AND CROFT, W. B. Relevance-based language model. In *Proceedings of the 24th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval* (2001), pp. 120–127.

[50] LEE, K. S., CROFT, W. B., AND ALLAN, J. A cluster-based resampling method for pseudo-relevance feedback. In *Proceedings of the 31st Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval* (2008), pp. 235–242.

[51] LEUSKI, A. Evaluating documents clustering of interactive information retrieval. In *Proceedings of the 10th Intl. Conf. on Information and Knowledge Management (CIKM'01)* (2001), C. Yu, H. Paques, L. Liu, and D. Grossman, Eds., pp. 33–40.

[52] LIU, X., AND CROFT, W. B. Cluster-based retrieval using language models. In *Proceedings of the 27th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval* (2004), M. Sanderson, K. Jarvelin, J. Allan, and P. Bruza, Eds., pp. 186–193.

[53] LUHN, H. P. The automatic creation of literature abstracts. *IBM Journal of Research and Development 2* (1958), 159–165.

[54] MARON, M. E., AND KUHNS, J. K. On relevance, probabilistic indexing and information retrieval. *Journal of the ACM 7* (1960), 216–244.

[55] MILLER, G. A. Wordnet: a lexical database for english. *Communications of the ACM 38*, 11 (1995), 39–41.

[56] NAGIH, A., AND PLATEAU, G. A partition algorithm for 0-1 unconstrained hyperbolic programs. *Investigacion Operativa 1,2 and 3*, 9 (2000), 167–178.

[57] PONTE, J., AND CROFT, B. A language modeling approach in information retrieval. In *Proceedings of the 21st Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval* (1998), pp. 275–281.

[58] PORTER, M. An algorithm for suffix stripping. *Program 14*, 3 (1980), 130–137.

[59] ROBERTSON, S., AND WALKER, S. Some simple approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proceedings of the 17th*

Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval (1994), W. B. Croft and C. J. van Rijsbergen, Eds., pp. 186–193.

[60] ROBERTSON, S. E. The probabilistic ranking principle in ir. *Journal of Documentation 33* (1977), 294–304.

[61] ROBERTSON, S. E. Understanding inverse document frequency: On theoretical arguments for idf. *Journal of Documentation 60*, 5 (2004), 503–520.

[62] ROBERTSON, S. E., AND SPÄRCK JONES, K. Relevance wighting of search terms. *Journal of the American Society for Information Science 3*, 27 (1976), 129–146.

[63] ROBERTSON, S. E., WALKER, S., AND BEAULIEU, M. Experimentation as a way of life: Okapi at trec. *Information Processing and Management 36* (2000), 95–108.

[64] ROBILLARD, P. (0,1) hyperbolic programming problems. *Naval Research Logistic Quarterly*, 18 (1971), 47–58.

[65] ROCCHIO, J. J. Relevance feedback in information retrieval. In *The SMART Retrieval System*, G. Salton, Ed. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1971, pp. 313–323.

[66] RUTHVEN, I., AND LALMAS, M. A survey on the use of relevance feedback for information access systems. *The Knowledge Engineering Review 2*, 18 (2003), 95–145.

[67] SAKAI, T., MANABE, T., AND KOYAMA, M. Flexible pseudo-relevance feedback via selective sampling. *ACM Transactions on Asian Language Information Processing 4*, 2 (2005), 111–135.

[68] SALTON, G., AND BUCKLEY, C. Term-weighting approaches in automatic text retrieval. *Information Processing and Management 24*, 5 (1988), 513–523.

[69] SALTON, G., AND LESK, M. E. Computer evaluation of indexing and text processing. In *The SMART Retrieval System*, G. Salton, Ed. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1971, pp. 143–180.

[70] SALTON, G., WONG, A., AND YANG, C. S. A vector space model for information retrieval. *Communications of the ACM 18*, 11 (1975), 613–620.

[71] SALTON, G. E. *The SMART retrieval system*. Englewood Cliffs, NJ: Prentice-Hall, 1971.

[72] SEBASTIANI, F. Machine learning in automated text categorization. *ACM Computing Surveys 34*, 1 (2002), 1–47.

[73] SINGHAL, A. Modern information retrieval: A brief overview. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering 24*, 4 (2001), 35–43.

[74] SINGHAL, A., BUCKLEY, C., AND MITRA, M. Pivoted document length normalization. In *Proceedings of the 19th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval* (1996), pp. 21–29.

[75] SOBOROFF, I. A comparison of pooled and sampled relevance judgments in the trec2006 terabyte track. In *Proceedings of the 1st International Workshop on Evaluating Information Access* (2007), pp. 22–31.

[76] SPÄRCK JONES, K. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation 28* (1972), 11–21.

[77] SPÄRCK JONES, K., ROBERTSON, S. E., AND SANDERSON, M. Ambiguous requests: implications for retrieval tests, systems and theories. *ACM SIGIR Forum 41*, 2 (2007), 8–17.

[78] SPÄRCK JONES, K., WALKER, S., AND ROBERTSON, S. E. A probabilistic model of information retrieval: development and comparative experiments part 2. *Information Processing and Management 36* (2000), 809–840.

[79] SPINK, A., JANSEN, B. J., WOLFRAM, D., AND SARACEVIC, T. From e-sex to e-commerce: Web search changes. *Computer 3*, 35 (2002), 107–109.

[80] STEINBACH, M. A comparison of document clustering techniques. In *KDD Workshop on Text Mining* (2000).

[81] TOMBROS, A., VILLA, R., AND VAN RIJSBERGEN, C. J. The effectiveness of query-specific hierarchical clustering in information retrieval. *Information Processing and Management*, 38 (2002), 559–582.

[82] TROY, A. D., AND ZHANG, G. Q. Enhancing relevance scoring with chronological term rank. In *Proceedings of the 30th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval* (2007), pp. 599–606.

[83] VAN RIJSBERGEN, C. J. *Information Retrieval (2nd ed.)*. London: Butterworth, 1979.

[84] VAN RIJSBERGEN, C. J., AND CROFT, W. B. Document clustering: An evaluation of some experiments with the cranfield 1400 collection. *Information Processing and Management*, 11 (1975), 171–182.

[85] VOORHEES, E. M. The clustering hypothesis revisited. In *Proceedings of the 8th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval* (1985), J. M. Tague, Ed., pp. 188–196.

[86] VOORHEES, E. M. Query expansion using lexical-semantic relations. In *Proceedings of the 17th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval* (1994), pp. 61–69.

[87] VOORHEES, E. M. Variations in relevance judgments and the measurement of retrieval effectiveness. In *Proceedings of the 21st Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval* (1998), W. B. Croft, A. Moffat, C. J. van Rijsbergen, and J. Zobel, Eds., pp. 315–323.

[88] WILLETT, P. Recent trends in hierarchic document clustering: A critical review. *Information Processing and Management*, 24 (1988), 577–597.

[89] WONG, W. S., LUK, R. W. P., LEONG, H. V., HO, K. S., AND LEE, D. H. Re-examining the effects of adding relevance information in a relevance feedback environment. *Information Processing and Management 44* (2008), 1086–1116.

[90] WU, H. C., LUK, R. W. P., WONG, K. F., AND KWOK, K. L. A retrospective study of a hybrid document-context based retrieval model. *Information Processing and Management 43* (2007), 1308–1331.

[91] WU, H. C., LUK, R. W. P., WONG, K. F., AND KWOK, K. L. Interpreting tf-idf weights as making relevance decisions. *ACM Transactions on Information Systems 26*, 3, Article 13 (2008), 1–37.

[92] WU, H. C., LUK, R. W. P., WONG, K. F., KWOK, K. L., AND LI, W. J. A retrospective study of probabilistic context-based retrieval. In *Proceedings of the 28th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval* (2005), pp. 663–664.

[93] XU, J., AND CROFT, W. B. Improving the effectiveness of information retrieval with local context analysis. *ACM Transactions on Information Systems 18*, 1 (2000), 79–112.

[94] XU, R., AND WUNSCH II, D. Survey of clustering algorithms. *IEEE Transactions on Neural Networks 16*, 3 (2005), 645–678.

[95] YANG, Y. An evaluation of statistical approaches to text categorization. *Information Retrieval 1* (1999), 69–90.

[96] ZAMIR, O., AND ETZIONI, O. How reliable are the results of large-scale information retrieval experiments? In *Proceedings of the 21st Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval* (1998), W. B. Croft, A. Moffat, C. J. van Rijsbergen, and J. Zobel, Eds., pp. 307–314.

[97] ZAMIR, O., AND ETZIONI, O. Web document clustering: A feasibility demonstration. In *Proceedings of the 21st Ann. Intl. ACM SIGIR Conf. on Research and*

Development in Information Retrieval (1998), W. B. Croft, A. Moffat, C. J. van Rijsbergen, and J. Zobel, Eds., pp. 46–54.

[98] ZHAO, Y., AND KARYPIS, G. Evaluation of hierarchical clustering algorithms for document datasets. In *Proceedings of the 25th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval* (2002), pp. 515–524.