Incorporating Context within the Language Modeling Approach for *ad hoc* Information Retrieval

Leif Azzopardi

Thesis Submitted in partial fulfilment of the requirements of the University of Paisley for the award of Doctor of Philosophy

School of Computing
University of Paisley

2005

Abstract

In this thesis, we investigate using the Language Modeling approach for *ad hoc* Information Retrieval as a theoretically principled framework for encoding contextual evidence. Using context to improve retrieval performance is a current challenge within the discipline and presents a major challenge to the research community. The Language Modeling approach provides a natural and intuitive means of encoding the context associated with a document. However, the Language Modeling approach also represents a change to the way probability theory is applied in *ad hoc* Information Retrieval and makes several assumptions for its application[112, 113, 57, 96]. We consider these assumptions and study them in detail during the course of this thesis. Central to the assumptions is the key implication that better retrieval performance can be obtained through developing better representation of the documents. We posit that the context associated with a document will enable the development of such representations - *context based document models*. This premise relies upon the explicit and implicit assumptions of the Language Modeling approach being valid, which have, up until now, not been fully tested or verified. Through the course of this thesis we (1) formalize the assumptions of the Language Modeling approach; (2) motivated by the implications of these assumptions we present our framework for estimating context based document models; (3) perform a comprehensive analysis of the main assumptions underlying the Language Modeling approach, not only to validate the approach, but to deepen our understanding of the retrieval model itself, and; (4) empirically assess the performance of the context based document models against the standard document models on various test collections and contexts. Our findings show that there are occasions when context based document models outperform the standard document model. Further analysis with respect to underlying assumptions though reveals some of the limitations of the Language Modeling approach. We discuss these limitations and suggest an alternative approach for embedding context within the model. Finally, we propose an Integrated Language Modeling approach which formalizes the existing theory and practice within one framework. This not only addresses some of the concerns over the standard Language Modeling approach, but also enables the integration of various forms of context within the one framework.

Acknowledgements

To my supervisors, Mark Girolami, Keith van Rijsbergen and Malcolm Crowe, thank you for your support and encouragement throughout the course of this thesis. In particular, I would like to thank Mark Girolami for his primary role in my studies and for showing me the beauty (and beast) of probabilistic modeling.

Many thanks to Mark Baillie for providing encouragement in the final stages and reviewing much of this thesis. Thanks to Iraklis Klampanos for helping me with the diagrams. Also, thanks to David Losada for providing many insightful comments and questions in preparation for the defense of this thesis.

Best wishes to the Information Retrieval Group at the University of Glasgow for kindly hosting me during the latter stages of my thesis and thanks for providing access to the web collection used in my experiments. Also, thanks to ScotGRID at the University of Glasgow for providing the processing power to run the experiments conducted in this thesis.

This PhD was funded by the University of Paisley, Memex Technology Ltd and the Overseas Research Students Award Scheme.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Leif Azzopardi*)

Table of Contents

List of Figures

List of Tables

Nomenclature

Throughout this thesis the following notion will be used, unless stated otherwise.

C the collection

$d \in D = \{d_1, \ldots, d_{|D|}\}$ the document space, where $|D|$ the number of documents in the collection

$t \in T = \{t_1, \ldots, t_{|T|}\}$ the term space, where $|T|$ denotes the number of terms in the collection

$z \in Z = \{z_1, \ldots, z_{|Z|}\}$ the latent factors, where $|Z|$ denotes the number of latent factors

$k \in K = \{k_1, \ldots, k_{|K|}\}$ the topic space, where $|K|$ denotes the number of topics

r denotes the binary random variable relevance[1], which takes the two values, R and N.

R denotes relevance (to mean the document is judged relevant to the information need)

N denotes non-relevance (to mean the document is judged not relevant to the information need)

$n(x,y)$ the number of times x occurs with or in y, usually in reference to the number of times a term t occurs in document d, that is $n(t,d)$

$n(d)$ the total number of terms that occur in document d, that is $\sum_{t \in T} n(t,d)$

$p(x|y)$ the probability of x given y

$idf(t)$ the inverse document frequency of term t

ℓ log-likelihood of a document, $\log p(d)$

[1] Note this deviates from the standard way of referring to a random variable.

Language Models

θ_y a model of y, where y refers to the particular type of model. The model produces/generate terms from the vocabulary with a probability $p(t|\theta_y)$.

θ_C a model of the collection C, referred to as the collection background model.

θ_R a model of the relevance R, referred to as the relevance model.

θ_N a model of the non relevance N, referred to as the non relevance model.

Θ_X a model of the context defined by X for a given d, referred to as the context model and defined as $p_d(t|\Theta_X)$, (unsmoothed).

θ_x a model of the background context defined by x for a given d, referred to as the context background model and defined as $p_d(t|\theta_x)$, (smoothed).

Abbreviations

A1 Assumption One - Correlation

A2 Assumption Two - Unification

A3 Assumption Three - Discrimination

ASK Anomalous State of Knowledge

BIM Binary Independence Model

BS Bayes Smoothing

CLM Collection Langauge Model

DC Document Classification

DFR Divergence From Randomness

DIR Distributed Information Retrieval

EM Expectation Maximization

HA1 Hypothesis of Assumption One - Correlation

HA2 Hypothesis of Assumption Two - Unification

HA3 Hypothesis of Assumption Three - Discrimination

HMM Hidden Markov Model

IDF Inverse Document Frequency

IF Information Filtering

IR Information Retrieval

IRS Information Retrieval System

JM Jelinek Mercer

KL Kullback-Leibler

LDA Latent Dirichlet Allocation

LLR Log Likelihood Ratio

LM Language Model

LP Laplace

LSA Latent Semantic Analysis

LSI Latent Semantic Indexing

NLP Natural Language Processing

PLSA Probabilistic Latent Semantic Analysis

PLSI Probabilistic Latent Semantic Indexing

PRP Probability Ranking Principle

RM Relevance Model

SLM Statistical Language Modelling

TF Term Frequency

VSM Vector Space Model

WWW World Wide Web

XLM Context Language Model

ZPP Zero Probability Problem

Collections

AP Applied Press Collection

CACM Computer Abstracts from the ACM

CISI CISI Abstracts

CRAN Cranfield Aeronautical Abstracts

MED MedLine Abstracts

WSJ Wall Street Journal Collection

Measures

BDM 'best' data model

BRM 'best' retrieval model

mAP mean Average Precision

mPL mean Predictive Likelihood

p@0% precision at 0% recall

p@10% precision at 10% recall

p@30docs precision at 30 documents

Chapter 1

Introduction

Searching and finding useful information can be quite an arduous task. For today's searcher the amount of information accessible is almost limitless, but only a very small fraction of this available information will actually be relevant or useful to the searcher and their information need(s).

This accessibility to information has resulted from a worldwide process of computerization and networking of computing systems, referred to as the Internet or the World Wide Web (WWW). Since the introduction of the World Wide Web there has been a very large increase in the amount of digital information available, which continues to grow at a phenomenal rate. The predominant kind of information on the Web is textual documents, although other forms of digital data are becoming more prevalent including images, speech, audio and video.

It is natural that an information seeker will resort to the WWW (or some other specific information repository located in the WWW) in order to satisfy their information needs. However, given the overwhelming amount of information available they will eventually suffer from Information Overload[133]. This is where the amount of information presented, pushed or pulled, exceeds the cognitive capacity of the user. One of the most effective ways to deal with the information overload is through the use of Information Retrieval (IR) techniques.

Information Retrieval is the discipline concerned with the representation, storage, organization, analysis, searching and dissemination and access to information objects[148]. This has culminated in a range of strategies for dealing with information being developed, such as: *ad hoc* Retrieval, Information Filtering, Document Classification, Clustering, Summarization, Machine Translation, Information Visualization, Topic Detecting and Tracking, and Information Extraction (See [7, 6, 141, 148, 128] for details of such tasks). Each technique addresses a sub-problem when dealing with information and attempts to reduce the Information Overload. For instance, if the information seeker has a persistent long term information need, such as wanting to know all gossip, rumors and discussions about their favorite sporting team, then topic tracking and detection may be employed to automatically find information relevant to this desire[72]. The information seeker is provided with an aggregated result set updated over time, and is not burden by repeatedly searching. Another example is when a short document summary is provided along with each of the document references in the ranked list of search results[152]. It has been shown that the information seeker can identify relevant material more readily, because they can examine the context in which the query appears in the document. Such methods reduce the amount of time and effort required in obtaining relevant documents.

The focus of this thesis is on the most popularly used IR technique, *ad hoc* Information Retrieval. This is the main service rendered by a search engine, showing the reliance information seekers place on IR technology. In a survey of Internet users during 1999, it was shown that 85 percent have used a search engine[84]. The success of search engine related companies over the past five years would suggest that this figure would be much higher today. The purpose of a search engine, like that of any other Information Retrieval System (IRS), is to satisfy the user's underlying information need, by accepting a request in the form of a query, and returning a set of references to documents which will satisfy the user's information need.

An information need could be as simple as locating reviews of the latest movies, or more complex needs, such as locating Shakespeare prose containing the use of imagery about life and death. Providing the information seeker with the ability to submit one off queries about a particular topics enables them to quickly shift through documents

in the collection. Since these queries are very dynamic and formulated on the fly, the task is referred to as *ad hoc*.

The goal of *ad hoc* Information Retrieval is to return all those references which will be relevant to information seeker's information need, and ideally only those relevant to be returned. Unfortunately this is seldom the case because of the inherent uncertainty in the retrieval process. Consequently, a ranked list is usually provided, where the references are ordered in decreasing relevance or usefulness. In the past, IRS have relied upon using the content features within a document (i.e. the terms occurring within the documents) to retrieve documents by matching the query terms to the document terms, ignoring any contextual features internal or otherwise. Currently, IR researchers have been exploring the use of context in an effort to improve the quality of the results returned. A notable example is the PageRank[109] algorithm which attempts to gauge the popularity of a page by the number and quality of incoming links. Often such contextual information is mixed together with a document's content score in a heuristical manner, without any clear rationale or basis. This is problematic because invariably mixing parameters between the different scores need to be estimated and typically can only be set according to the end result (i.e. try it and find out). The solutions obtained are then specifically tailored for that particular case and may not generalize to other cases. Preferably, we would like to incorporate the contextual evidence within the model, such that: (1) it is done so in a principled manner, (2) any parameters can be estimated *a priori*, and (3) that these settings provide comparable or superior retrieval performance.

1.1 Research Questions and Hypothesis

In this thesis, we attempt to use context in a principled[1] manner under the language modeling framework for *ad hoc* text retrieval. By context, we mean the semantic associations between documents, which can be expressed as a relationship between

[1]By principled we mean that there is a theoretical basis which the model is grounded in, where the theoretical basis is derived from a particular branch of mathematics such as logic, probability or geometry.

one document and another. Semantic associations may be formed in number of ways, not limited to but including; collaborative interactions with the collection, the topical grouping of documents and references/links between documents. Essentially, a semantic association is an association between documents made by a user of the system which reflects in some way how the document relates to other documents within the collection. For instance, when a journal article is written the author includes a list of references, which may or may not be similar, but for the purposes of the document are semantically related through the author's text. Hence, we are trying to capture how the user actually views the relationship between documents and exploit this during retrieval. In Chapter 4, we shall see how this notion of context can be naturally incorporated into the Language Modeling approach.

Considering context as the semantic associations between documents is related to the *Cluster Hypothesis*[67], which states that:

> Closely associated documents tend to be relevant to the same request

Traditionally, the association between documents has been implemented through clustering the collection of documents in accordance to some *similarity* metric (such as Dice's co-efficient [148]). This uses the document's content to form associations, such that similar documents are grouped together. However, if the associations between documents are defined *semantically*, then a corresponding context based hypothesis can be expressed. We formulate this as the *Context Hypothesis*, which states:

> Semantically associated documents tend to be relevant to the same request

To evaluate this hypothesis, as we have already mentioned, we employ the recently proposed Language Modeling (LM) approach to *ad hoc* Information Retrieval. We develop a principled approach that incorporates semantic associations within the document language modeling process and contend that this is an implementation of the *Context Hypothesis*.

The Language Modeling approach attempts to capture the statistical regularities within the text of a document, and model the language as a probability distribution over all the terms in the vocabulary. For each document, a document model which captures its underlying generative nature of documents text is defined. Given the query text, a

prediction can be made of the likelihood that the query text would have been generated by each document model. This prediction is then used to rank the documents.

However, the LM approach makes three key assumptions about the IR process. The second of these directly relates to the validity of the Context Hypothesis. The assumptions imply that the user has some knowledge of the distribution of terms within documents, and that the document models should reflect this understanding. Their understanding constitutes the context of the document. So implicitly the context of the document, or what we associate it with, will affect our model of the document. This is how we think of the document, and this should be reflected in the model of the document - hence the creation of *context based document models*. By doing so should, according to the assumptions, result in better retrieval performance.

In this thesis, we formalize the assumptions of Language Modeling and then provide a comprehensive study to determine how well these assumptions hold in practice. Also, we present our framework for generating context based document models, which is motivated by the implications of the assumptions of Language Modeling. We perform an empirical evaluation of these context based models and determine whether we can generate a better representation of the documents using context, and whether this leads to better retrieval performance. Finally, after considering the outcomes from our analysis, we propose an integrated Language Model which enables the user's context to be incorporated into the retrieval process in an alternative and seamless manner.

1.2 Structure of thesis

After the introduction chapter, the content of this thesis is divided into seven chapters. An outline of the remaining chapters is detailed below:

- **Chapter 2:** The basic concepts pertaining to an Information Retrieval System, Information Retrieval models, their implementation and evaluation are presented.
- **Chapter 3:** A survey of the current literature about Language Modeling for *ad*

hoc text retrieval is presented. This includes our formalization of the underlying assumptions of the Language Modeling Approach.

- **Chapter 4:** Our approach to developing context based document models is proposed. We include discussion on how this approach relates and differs from other models previously proposed.
- **Chapter 5:** The underlying assumptions of the Language Modeling Approach are empirically tested. We state each assumption as a hypothesis and then apply tests to determine whether the assumptions hold.
- **Chapter 6:** Several different types of semantic association are used as the context. Each is used to build context based document models which are tested against the standard document modeling approaches. A continuation of research on the second assumption is also included.
- **Chapter 7:** The results from all our experimental work are considered and interpreted with respect to the hypotheses outlined during the course of the thesis and how the findings relate or impact upon other research.
- **Chapter 8:** A summary of the thesis is presented along with areas of future research directions stemming from the work presented herein.

1.3 Novel Contributions

Within this thesis there are several notable contributions. These are outlined below:

- The formalization of the underlying assumptions of the Language Modeling Approach for *ad hoc* Information Retrieval (See Section 3.2 and Appendix A).
- The development of a principled framework for context based document modeling (See Chapter 4).
- The analysis of the underlying assumptions of the Language Modeling approach to *ad hoc* Information Retrieval (See Chapter 5).

- An evaluation of Probabilistic Latent Semantic Analysis within a language modeling framework (See Section 6).
- An empirical analysis of context based document language models on different collections. Contexts were represented by the association between documents, either through unsupervised learning techniques, user interaction, or through explicit user reference, such as hyperlinks and citations (see Chapter 6).
- An alternative explanation of relevance in the query likelihood approach (See Chapter 7) which provides a different explanation of the role of smoothing.
- Comments on the estimation and assignment of parameters for document models (See Chapter 7).
- The proposal of the integrated Language Modeling approach (See Chapter 7).

1.4 Publications

Publications arising as part of the thesis work:

1. *Azzopardi, L. and Girolami, M. and van Rijsbergen, C. J.* , Investigating the Relationship between Language Model Perplexity and IR Precision-Recall Measures, **In the Proceedings of the 26th Annual ACM Conference on Research and Development in Information Retrieval**, 2003.
2. *Azzopardi, L. and Girolami, M. and van Rijsbergen, C. J.*, User Biased Document Language Modeling, **In the proceedings of the 27th Annual ACM Conference on Research and Development in Information Retrieval**, 2004.
3. *Azzopardi, L. and Girolami, M. and van Rijsbergen, C. J.*, Topic Based Language Models for *ad hoc* Information Retrieval, **In the Proceedings of the International Joint Conference in Neural Networks**, 2004.
4. *Azzopardi, L. and Girolami, M. and Crowe, M.*, Probabilistic Hyperspace Analogue to Language, **In the proceedings of the 28th Annual ACM Conference on Research and Development in Information Retrieval**, 2005.

Chapter 2

Basic Information Retrieval Concepts

In the previous chapter, we briefly outlined the purpose and tasks of *ad hoc* Information Retrieval. *Ad hoc* Information Retrieval strives to provide the user with information items which are *relevant* to the user given the user's *information need*. This objective exceeds the mandate of data retrieval, which strives to retrieve all items which satisfy clearly defined conditions[148]. The retrieval of relevant information objects given an information need, is less clearly defined as the information need is expressed as a query. This is an imprecise description of the underlying information need and this introduces an inherent uncertainty in the retrieval process. This chapter introduces the basic concepts relating to *ad hoc* Information Retrieval, the different types of models and the implementation, interaction and evaluation of Information Retrieval Systems.

2.1 Ad Hoc Information Retrieval

As previously mentioned, according to van Rijsbergen[148]:

> Information Retrieval deals with the presentation, storage, organization of, and access to information items.

This definition encapsulates the essence of any Information Retrieval System (IRS), where the Information items could be text, images, audio, video or a combination of,

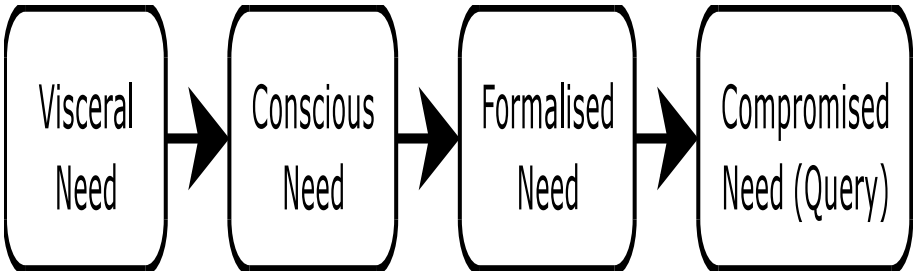

Figure 2.1: The Translation of a user's information need to query according to Taylor[145].

all four, multimedia. Traditionally, the focus has been on text retrieval, since this is the most predominant and abundant type of information available. The remainder of this chapter and indeed this thesis, focuses on the retrieval of text documents. Furthermore, this thesis is predominantly concerned with *ad hoc* retrieval for text documents. *Ad hoc* text retrieval is the core Information Retrieval task, not only because it is most often used and familiar to the general public but also because much of the research in other IR tasks has drawn upon the research in this area (such as document classification, filtering and summarization).

The goal of *ad hoc* Information Retrieval is to retrieve a set of references that will satisfy the user's information need. An information need arises when the user realizes that their state of knowledge is inadequate to achieve their task or general goal [9]. This inadequacy can be of many sorts, stemming from a gap or lack in knowledge, uncertainty of current knowledge and/or incoherence of available knowledge. It also covers when the user is unable to specify their information need. This is referred to as the Anomalous State of Knowledge (ASK)[9]. To address this ASK, the user may consult a colleague, read a book, or in this case use an IRS. To use an IRS, the user must express their information need as a query. Ideally, the IRS would then respond by returning only those documents which would satisfy the user's information need. However, this is seldom the case, because of the inherent uncertainty in the querying process.

The representation and expression of the information need has been long recognized as a fundamental problem in Information Retrieval. In 1968, Taylor [145] recognized

this, and suggested that the querying process goes through four levels (see Figure 2.1). The first level is the visceral need, which is internal and actual but unexpressed and perhaps even an inexpressible need for information. At the next level is the conscious need, where there is a realization within the brain that a need exists and this need is usually ambiguous and ill-defined. This conscious need then becomes the formalized need, and is an expression of the user's need. It is then reformulated into a query (now representing the 'compromised need' in Taylor's terminology) so it can be presented to an Information Retrieval system. The query is a very sparse and poor substitute for the user's underlying information need.

At each level information is lost in translation from visceral need to the query, as a result of three well known problems[102]. The first two problems, which degrade the information need, occur when expressing the conscious need. They have been referred to as the label effect[66] and the vocabulary problem[39]. The *label effect* results when the user expresses his need in terms of 'labels' or 'keywords' and not as a complete sentence. The *vocabulary problem* surfaces when there is a mismatch between the 'labels' or 'keywords' used in the document and those used in the query. The third problem is the *formalization operation*, which arises because the system language is often not natural language. Hence, a translation from natural language to the system language is required, further degrading the correspondence to the user's actual information need.

In response to a query the IRS will return a ranked list of documents, where the ranking is in decreasing order of usefulness or *relevance* with respect to the submitted query [1]

Relevance is an integral concept within information retrieval, as the goal is to retrieve relevant information. However, defining what is meant by relevance and deciding what is relevant is very difficult. Relevance may be referred as meaning topicality, usefulness, user satisfaction, situational relevance, similarity, and utility amongst others [101]. For further reading and discussion about relevance see [101, 102, 22, 131, 35, 110]. However, in this thesis, we consider relevance as the concept pertaining to the usefulness of a document with respect to the user's information need[25]. Thus, if a

[1] Strictly speaking an IRS does not actually return the documents, but references to documents. For convenience, we shall dispense with the formality.

document satisfies a user's information need it is said to be relevant, if the document does not, then the document is said to be not relevant. Further, the relevance of a document is only known after the user has examined the document and deems it relevant (or not).

Once the user reviews the documents and makes a judgement about the relevance of the documents in the ranked output, feedback may be given to the system. This interaction with the system may continue in a series of stages of refining/reformulating the information need until the user decides to curtail their search session, hopefully, with their information need satisfied.

2.1.1 IR and other tasks

As we have previously mentioned *ad hoc* IR is related to other IR tasks, such as Information Filtering (IF) and Document Classification (DC). The scenarios for IF and DC are slightly different than *ad hoc* IR, but essentially the need to score and classify a document with respect to an information need (expressed as a query) is the same. In *ad hoc* IR, the collection of documents is assumed to be relatively static and single uses of the IRS for a one time information need, hence *ad hoc* where the query is an impoverished representation of the information need. On the other hand, in Information Filtering, new incoming documents are continually being added to the collection (in a document stream), and the information need is expressed adequately or at least somewhat more verbosely than in an *ad hoc* IR. This is because the user is expected to dedicate a reasonable amount of time to constructing and refining the filter (query). The filter represents the users's long term and persistent information need, constructed from a set of example relevant documents. When new documents are added to the collection they are classified as relevant or not to the filter. Hence the observation that *ad hoc* IR and IF are two sides of the same coin[8], as the tasks are essentially the same but from two different perspectives. IF is dynamic with a well defined information need, while *ad hoc* IR is static, with a vague, ill defined information need, though both *ad hoc* IR and IF can be viewed as addressing a document classification problem. In Document Classification, there is a set of classes which have a set of documents assigned

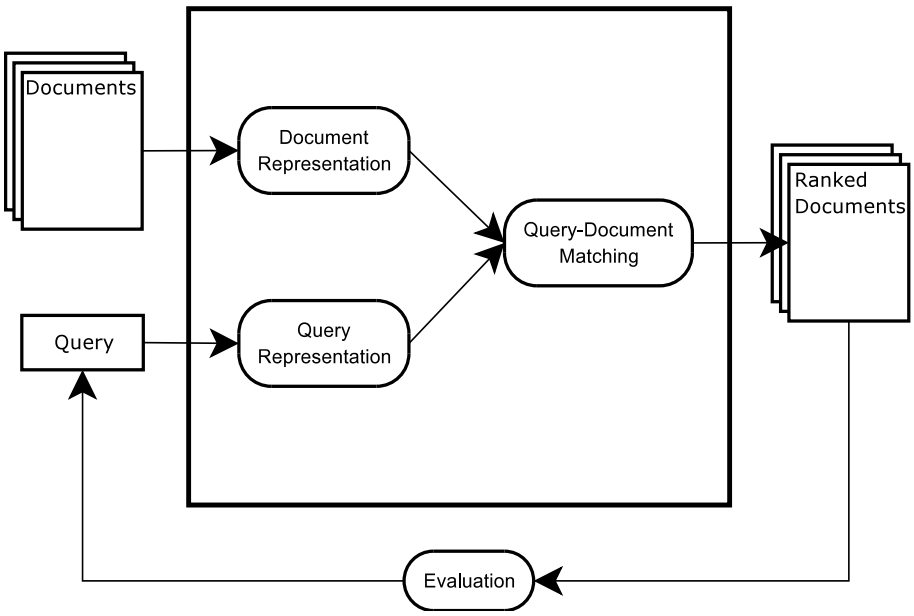

Figure 2.2: A typical view of an Information Retrieval System[148].

to each class. Class assignment can be determined using the probability of a document given each class. Now, if there are only two classes, relevant and non-relevant with respect to the query, then the document classification task is essentially *ad hoc* IR[86]. The documents from the collection or stream are ranked or scored according to the probability of the document belonging to the relevant class or not. In the case of IF, it is generally assumed that a set of exemplars exists for the relevant and non relevant classes. This is a luxury not afforded to *ad hoc* IR, under the DC interpretation, the query terms form a very sparse and solitary 'document' example of the relevant class.

So far we have described the process of *ad hoc* retrieval, and how it compares to some of the other tasks in IR. In the next section, we describe the components of an Information Retrieval System.

2.2 Information Retrieval System

The main components of an Information Retrieval System to facilitate the goal of ad hoc information retrieval are shown in Figure 2.2. An IRS takes as input a set of

documents and the user's query. The documents and query are transformed into the representation expected by the retrieval model. The matching function, which is defined by the IR model, is then used to rank documents according to the query. The user evaluates the ranked list and may provide some kind of feedback to the IRS. The main components are:

- **Document and Query Representations** The document/query is parsed into tokens, before an undergoing a series of operations to transform the raw tokens into indexable features. The indexable features are then used to represent the document and query. The final representation will depend on the model/matching function employed by the IRS.
- **Model - Matching Function** The Information Retrieval Model determines the matching function employed. The matching function is used to score document representations against query representations. A ranked list of documents is presented to the user for evaluation[2]. Ideally, the ranked list is ordered in decreasing usefulness or relevance[120].
- **Evaluation** The user inspects the ranked list and determines which documents are relevant. These judgements are used to evaluate the performance according to the position of the relevant documents in the ranked list.
- **Feedback** The most general interpretation of feedback is any interaction with the system following retrieval. However, feedback is generally limited to relevance feedback[124], which refers to the set of documents marked as relevant and non-relevant by the user. The Relevance feedback is used to refine and reformulate the query.

A detailed description of each component is presented in the following sections.

[2]However, not all models will provide a ranked list. For example, the Boolean Model returns a set of documents which satisfy the query expression.

2.3 Document and Query Representation

The indexing process is the pre-processing of the document (or query) to extract a set of features to represent that document (or query). The three most common preprocessing steps are lexical analysis, morphological normalization and term feature selection.

The extracted set of features for a document is used to construct a document representation and this serves as a model of the original information contained within the document. Whilst the aim is to model the original information content of the document as accurately as possible, simplifications are required to cater for computational and storage constraints. The indexing features could consist of phrases, grammatical structure, amongst other units but are typically words (or word derivatives) extracted from the document. If these are selected by the indexing process, then the indexing unit is referred to as a *term*.

The following subsections describe the types of actions performed at each step of the indexing process. However, not all actions or steps need be applied. In the simplest case, a term could be simply a token from the document. However, various experiments have shown that performing pre-processing can improve retrieval effectiveness, reduce storage costs and improve efficiency. An example of transforming original document text into indexable units is also provided to complement the explanation. The original text is shown in Figure 2.3 and the transformations (tokenization, stemming and stopping) are shown in Figure 2.4.

2.3.1 Lexical Analysis

This is an essential part of the indexing process where the text in the document is broken into tokens (tokenization). This process may also include ignoring any text which is not a word, transforming the words to all one case, and addressing any punctuation such as apostrophes and hyphens[6]. Other considerations may also include[34]:

- structure within the document such as XML[3] and HTML[4]
- formatted documents such as portable document format and postscript
- tag or meta data
- numerical equations, scientific formula and other symbolic notation

Sometimes more implicit structure within the text such as phrases, entities, and acronyms may also be considered and indexed. However, more sophisticated techniques from Natural Language Processing (NLP) that have been applied to extract more useful units of representation have been met with limited success[138]. The more successful approaches have tended to be 'shallow tools' that do not perform a complete parse of the document, for instance, the extraction of entities from documents and the disambiguation of words. It has been shown that linguistically motivated features are not necessary to achieve effective IR [138]. Indeed, the marriage between NLP and IR has been described as a loose coupling as opposed to an integrated pair [136]. NLP tends to focus on specific well understood problem domains within text, while IR is concerned with the retrieval of relevant information, which is characteristically uncertain in nature.

2.3.2 Morphological Normalization

The most commonly applied form of Morphological Normalization within IR is suffix stripping or stemming and is used to reduce words down to their base word variant[114, 74, 65]. The Porter Stemming[114] algorithm, one of the most popular stemmers, which extracts the term stem by utilizing the structure within complex suffixes that occur in the English langauge. Since suffixes are composed of simpler suffixes, each simpler suffix is removed in turn to reduce the word to its stem. For instance, 'terminator', 'terminating', 'termination', 'terminate' all have a word stem 'termin'. This has the advantage of reducing the number of terms that are indexed and allows greater matches between word stems. However it also results in the loss of some information.

[3]XML is Extensible Markup Langauge.
[4]HTML is Hyper Text Markup Language.

SCENE II.
A bedchamber in the castle: DESDEMONA in bed asleep; a light burning.
Enter OTHELLO.
OTHELLO:
It is the cause, it is the cause, my soul,
Let me not name it to you, you chaste stars!
It is the cause. Yet I'll not shed her blood;
Nor scar that whiter skin of hers than snow,
And smooth as monumental alabaster.
Yet she must die, else she'll betray more men.
Put out the light, and then put out the light:
If I quench thee, thou flaming minister,
I can again thy former light restore,

Figure 2.3: Original Document Text: A short extract from Othello by Shakespeare as example text. See Figure 2.4 for the transformation to indexable units (terms).

For instance when the word 'terminal' is used to refer to an airport building, as opposed to implying death, then stemming 'terminal' back to 'termin' will match stems of a different context. Nonetheless, experimental analysis has confirmed that suffix stripping tends to improve IR performance [46, 65].

Other normalization techniques also include, but are not limited to:

- Synonym normalization where words that have the same meaning are transformed to the same term,
- Polysemy normalization where the converse is performed. A word that has different meanings, is assigned to different terms.

Tokenised:
scene ii a bedchamber in the castle desdemona in bed asleep a light burning enter othello othello it is the cause it is the cause my soul let me not name it to you, you chaste stars it is the cause yet i ll not shed her blood nor scar that whiter skin of hers than snow and smooth as monumental alabaster yet she must die else she ll betray more men put out the light and then put out the light if i quench thee thou flaming minister i can again thy former light restore

Stopwords Removed:
scene ii bedchamber castle desdemona bed asleep light burning enter othello othello soul chaste stars shed blood scar whiter skin snow smooth monumental alabaster die betray men light put light quench thee thou flaming minister thy light restore

Stemmed:
scene ii bedchamb castl desdemona bed asleep light burn enter othello othello soul chast star shed blood scar whiter skin snow smooth monument alabast die betrai men put light put light quench thee thou flame minist thy light restor

Figure 2.4: Transformations: From top to bottom: Tokenised, Stopped, and then Stemmed. Notice that the final representation does not distinguish between the different parts within the document and this translates into a loss of information and meaning.

2.3.3 Term Feature Selection

The selection of appropriate indexing units is an important problem in the representation process. In considering this problem, Luhn[91] hypothesized that the frequency data could be used to select sentences that best represent a document[148]. When the frequency f of a term occurring in the collection is plotted against its rank r, where the terms are ranked in decreasing order of their term frequency, a hyperbolic distribution relating f to r is witnessed (See Figure 2.5). This distribution is usually referred to as Zipf's law which states that the product of the term frequency by rank is approximately constant[166]. Luhn's idea was that the terms that occur most and least frequently in the collection are not very good features to use when representing documents. He postulated that the significance of a term was relative to its rank, such that as the frequency of terms decreased the significance also would increase to some maximum value. Then as the frequency of terms continued to decrease the significance of the terms would also decrease (see Figure 2.5, where the dashed curve indicates the importance of a term, and the solid line the frequency of the term).

The lower cut off defines the point beyond which a term is used so infrequently that the term will be too specific and not contributing significantly to the content of document, while the upper cut off defines the point where a term is considered too common and lacking in discriminative power between documents. Terms that occur in over approximately 80% of the documents tend to be poor discriminators and are ineffective for IR[148]. Such terms are often referred to as stop words. The two standard approaches to stop word removal are: (1) discard terms that occur in a predefined stop list (see [148]). Typical examples of terms on a stop list are 'the', 'of', 'a', 'there', 'then', 'though'; and (2) select the most useful terms by employing statistical methods, such as, a cut off based on frequency, the information gain measure, mutual information, χ squared statistic, or the strength method[158].

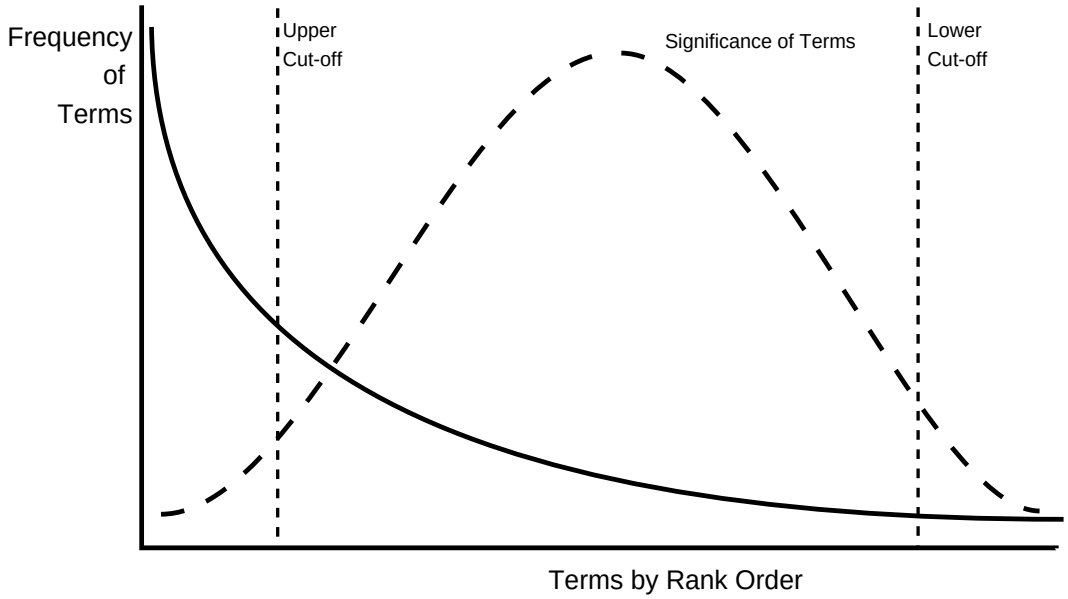

Figure 2.5: The distribution of terms ordered by term frequency (solid line) and the significance of terms according to Luhn[91] (dashed line).

2.3.4 Document and Query representation

The final stage of the pre-processing task is to use the indexed terms to build a representation of the document and query so that it can be used by the IR model's matching function. This representation typically consists of an inverted list of the terms that are present within the document. A weight is often attached to the presence of each term within the document depending on the representation required. Most representations do not consider the dependence of terms, this is referred to as the Independence Assumption. Such a representation is often described as a 'bag of terms' because it suggests that the ordering or sequential nature of the underlying text is ignored.

Given the set of documents D, where document $d \in D$ is represented by a $|T|$ dimensional vector $(t_1, \ldots t_{|T|})$, where $|T|$ is the number of terms within the collection given the vocabulary T. For each t_i a value is assigned depending on the representation. Sometimes, it is convenient to also refer to the representation as a set of terms such that $t \in d$ if the value assigned is non zero.

The index terms that occur within the document are assigned a weight specific to the representation employed by the IRS. Below are the main types of representation that are used, along with a brief description.

- **Binary Representation** The document is represented as either containing the term or not. We represent this by denoting the presence of a term by $t_i = 1$, and the absence as $t_i = 0$ within a document d. This is the simplest representation, storing the least amount of information about the relationship of terms in documents. The document representation consists of a set of terms and this representation is typically employed in logic based models, but may also be employed in the other IR models. For instance, the Binary Independence Model makes use of the Binary Representation (See Section 2.5.2 for details).

- **Weighted Representation** Luhn suggested that multiple occurrences of the same term indicated that the term was emphasized within the document, and this is representative of how important that term is within the document. So instead of a binary representation, a weighted representation may be employed to encode the term frequency information, the simplest of which is to represent the document by the term frequency information, such that the weight of a term t in a document d is equal to the number of times the term t occurs in document d, that is $n(t,d)$. However, more sophisticated weightings have been employed, the most popular of which is the Term Frequency (TF) by Inverse Document Frequency (IDF). This is usually referred to as *TF.IDF*, where the weight $w(t,d)$ is calculated as shown in Equation 2.1,

$$w(t,d) = n(t,d).idf(t) \tag{2.1}$$

  where $idf(t)$ is the inverse document frequency weight for the term t. This is determined by Equation 2.2,

$$idf(t) = \log \frac{|D|}{df(t)} \tag{2.2}$$

  where $df(t)$ is the number of documents in which term t occurs and $|D|$ is the

number of documents in the collection. Such weighted representations are used predominately within the Vector Space Model.

- **Probabilistic Representation** A special case of a weighted representation is the probabilistic representation, where the following constraints are imposed. The sum of the document's term vector must equal one and all the term weights must be positive. This representation of the terms as probability distribution over the vocabulary and is considered as a multinomial probability distribution. This representation really exploits the idea that the document is just a 'bag of words'. Hence, we can represent the distribution of terms within this bag quite easily by expressing the occurrence of terms as the probability of a term given a document, $p(t|d)$. The simplest estimate of $p(t|d)$ is the *maximum likelihood* estimate, which is the number of times the term occurs in a document divided by the total number of terms in the document (See Equation 2.3).

$$p(t|d) = \frac{n(t,d)}{\sum_{t' \in d} n(t',d)} \tag{2.3}$$

  In the next chapter we introduce the Language Modelling approach to *ad hoc* Information Retrieval which represents documents as a probability distribution over the vocabulary.

2.4 Information Retrieval Models

When discussing different document and query representations, we have already mentioned some of the models used in IR. The three classic IR Modelling approaches are the Boolean, Vector Space and Probabilistic models. Each model is an abstraction of the retrieval task, which makes assumptions about the retrieval process. These define how the model should be used or can be used and often lead to inherent limitations of the approach.

2.4.1 The Boolean Model

One of the earliest models proposed for Information Retrieval was the Boolean model[6]. It uses logical operators to define the set of documents to be retrieved for a given query. The Boolean Model represents documents as a binary representation and queries are expressed as a combination of terms and operators. The three basic Boolean operators are : (1) the logical product, AND; (2) the logical sum, OR, and; (3) the logical negation, NOT. Each operator will affect the set of results returned by a Boolean model. For instance, the query 'cat AND kitten' will return the set of documents which contain the both terms, whereas the query 'cat OR kitten' will return the set of documents which contains either terms.

Many commercial systems employ the Boolean model since it has a clean formalism and is conceptually simple, so that the query expression has precise semantics. Typically such precise expressions are not written by the user, who is generally content to submit a few key terms (where the AND operator is implicitly assumed between terms), as opposed to expressing specific and precise query expressions, which become difficult and laborious to formulate, since the boolean model will only return a set of documents, without recourse to the degree of their usefulness/relevance (ie. no ranking). This is clearly a major drawback of the model, evident when a large set of documents is returned. Consequently, several developments of the Boolean model have been proposed. One extension is the Fuzzy Set model[107]. This approach enables ranking through the assigning of a membership value reflecting to what extent the document satisfies the query. Non-binary term weightings are used to represent the degree of belief that the document belongs to the set of documents containing that term. Another such approach is the extended Boolean model[126] which incorporates term weightings and a distance measure within the Boolean model in order to obtain a ranked list of documents. These later variants have not been widely adopted nor thoroughly tested on large scale collections. This is mainly due to the computational complexity of the approaches and their inability to scale to large document collections.

Other types of logic based models exist such as those based around non-classical logic[149]. These approaches view the retrieval process as a logical implication, where

the document implies the query, i.e. $d \rightarrow q$. Due to the inherent uncertainty involved in the retrieval process, non-classical logic is more appropriate than first order logic. While there have been several implementations using Logical Imaging [24], Information Flow [80] and Belief Revision [81] they have had variable success in improving retrieval performance. The main problem with these models is the computational complexity introduced when moving beyond first order logic. For further details see [25] for a review of logical and uncertainty models for IR.

2.4.2 Vector Space Models

The Vector Space Model (VSM) represents both the query and document as weighted vector representations, q and d respectively. A predefined geometric function is used to compare the two vectors as a measure of their similarity. The degree of similarity between d and q vectors can be thought of as the correlation between the two vectors. The most popular function used is the cosine of the angle between d and q, as shown in Equation 2.4.

$$sim(d,q) = \frac{\sum_{t \in T} w(t,d).w(t,q)}{\sqrt{\sum_{t \in T} w(t,d)^2}\sqrt{\sum_{t \in T} w(t,q)^2}} \tag{2.4}$$

The smaller the angle between the two vectors the more similar the query is to the document. A ranked list is produced by ranking the documents in descending order of similarity. The weighting $w(t,d)$ assigned to a term in a document is usually the normalized tf.idf weight. However, there are a host of different possible weightings that could be employed. In fact over 1600 weighting schemes were tested by Salton and Buckley[125] which were variations of TF.IDF. Others have used genetic algorithms, or some other heuristical approach, to determine the ultimate term weighting combination[32]. In the absence of any understanding, intuition or rationale of, or behind, the weighting scheme employed it might be thought to be of limited utility, because it does not make the retrieval task any clearer. Regardless, the model provides a substantial improvement over the Boolean model, and remains a very popular and widely employed approach.

Various extensions of the VSM model have been proposed, such as the generalized vector space model[154] and Latent Semantic Indexing (LSI)[28]. The Generalized VSM assumes that terms are not independent, but that the co-occurrence of terms within documents induces dependencies amongst terms. However, such a representation has not been shown to outperform the standard VSM. LSI attempts to model concepts within the documents by projecting documents and queries into a lower dimensional latent space, then matching in this latent space instead of the term space. This model has attracted much attention in the IR community, as it has been shown to provide superior performance on some test collections, though its application to large scale test collections still remains a difficult challenge because of the huge computational cost required to perform LSI. In chapter 4, we describe the probabilistic version of LSI.

2.5 Probabilistic Models

Probability Theory has been applied to ad hoc IR in various ways. However, in this section, we present an overview of traditional probabilistic models and dedicate the next chapter to introducing the generative probabilistic approach, known as Language Modelling. The remainder of this thesis will then focus on Language Modelling for *ad hoc* IR.

In the traditional probabilistic model for IR the notion of relevance is the basis of its claim to be an optimal retrieval model. The most well known example of the traditional probabilistic model is the Binary Independence Model (BIM) which we shall describe in Section 2.5.2. First, we outline the basis of traditional probabilistic models.

2.5.1 Relevance, The probability of, and Ranking

In traditional probabilistic models for Information Retrieval the basic underlying concept is the notion of relevance. Such a model will attempt to rank documents in decreasing order of their estimated probability of relevance given a user's information need, which is represented by a query.

Relevance is assumed to be a dichotomous variable r which is defined by the relationship that may or may not hold between a document and a user of an IRS given a particular information need[25]. If the user believes a particular document is of use, then the relationship holds and the document is deemed relevant (i.e. $r = R$. Conversely, if the user believes that the particular document is of no use, then it is deemed not relevant (i.e. $r = N$). The fundamental question a traditional probabilistic model asks is:

> How probable is it that a document is relevant to a user's information need represented by the query?

In order to assign a relevance value to a document with respect to the information need, it is necessary to define a measure for relevance based on the representation of the document and the representation of the information need (query). Not on the document and information need themselves, as both the original content and need are considered unobservable. The probability of relevance R is computed given the document d and a query q, that is $p(r = R|d,q)$. The probability of non-relevance is also computed in order to separate the two classes. Thus, the probability of non-relevance N is computed given the document d and a query q, i.e. $p(r = N|d,q)$. These probabilities cannot be directly estimated, because Relevance can only be determined after the fact. However, by invoking Bayes' theorem we can estimate the probability of a document given relevance and the query $p(d|r,q)$. The log Odds of the $p(d|r = R,q)$ versus $p(d|r = N,q)$ is used to rank the documents. The log[5] is taken for mathematical convenience but still retains the correct ordering as it is a monotonic transformation, while the Odds ratio is used to ensure theoretically optimal retrieval [10, 115]. Both will be made apparent when we examine a specific implementation of the model and discuss the *Probability Ranking Principle* (PRP) [120]. The underlying idea is that terms within relevant and non-relevant documents are distributed differently. This is typically referred to as the *Cluster Hypothesis* [67]. The use of Odds ratios aims to discriminate relevant from non relevant based on the two different distributions.

The log Odds ratio of the probability of Relevance given the document and query $p(R|d,q)$ over the probability of Non-Relevance given the document and query $p(N|d,q)$

[5] Also, note that through the course of this thesis we will assume that the log is the natural logarithm.

can be expressed as being proportional to Equation 2.6

$$\begin{aligned}
\log O(r|d,q) &= \log \frac{p(R|d,q)}{p(N|d,q)} \\
&= \log \left(\frac{p(d|q,R)}{p(d|q,N)} \times \frac{p(q|R)}{p(q|N)} \times \frac{p(R)}{p(N)} \right) \qquad (2.5) \\
&= \log \left(\frac{p(d|q,R)}{p(d|q,N)} \times \frac{p(R|q)}{p(N|q)} \right) \\
&= \log \frac{p(d|q,R)}{p(d|q,N)} + \log \frac{p(R|q)}{p(N|q)} \\
&\propto \log \frac{p(d|q,R)}{p(d|q,N)} \qquad (2.6)
\end{aligned}$$

The expression $\frac{p(R|q)}{p(N|q)}$ is the prior log Odds, which is determined before the witnessing of any document, and as such is independent of the document. This constant can be ignored for the purposes of ranking. Ranking by the Odds ratio provides the most powerful statistical test according to the Neyman-Pearson Lemma[85] and in section 2.5.3 we show how this ranking can be shown to be optimal.

2.5.2 The Binary Independence Model

The Binary Independence Model (BIM)[119] provides an implementation of conventional probabilistic model defined in Equation 2.6. In BIM, documents are represented as a binary representation, such that a document is a vector of terms $d = \{t_1, \ldots, t_k\}$, which have a value of one if the term is present $(t_i = 1)$ and zero if the term is absent $(t_i = 0)$ from the document. This set of terms is defined by the k query terms. It is assumed that the presence of terms is independent in the set of relevant documents and the absence of terms is also independent in the set of non-relevant documents.

Let the probability of a term that is present in a relevant document be $pr_i = p(t_i = 1|R,q)$, and let the probability of a term that is present in a non-relevant document be $pn_i = p(t_i = 1|N,q)$. Also, let $p(t_i = 0|R,q) = 1 - pr_i$ and $p(t_i = 0|N,q) = 1 - pn_i$

represent the probability of a term that is not present in a relevant and non-relevant document respectively.

The probability of a document given that it is relevant is defined as:

$$p(d|q,R) = \prod_{i=1}^{k} pr_i^{t_i}.(1-pr_i)^{1-t_i} \tag{2.7}$$

Similarly, the probability of a document given that it is not relevant is defined as:

$$p(d|q,N) = \prod_{i=1}^{k} pn_i^{t_i}.(1-pn_i)^{1-t_i} \tag{2.8}$$

Substituting Equation 2.7 and Equation 2.8 into Equation 2.6 we obtain our ranking function:

$$\begin{aligned} \log O(r|d,q) &\propto \log \frac{p(d|q,R)}{p(d|q,N)} \\ &= \log \frac{\prod_{i=1}^{k} pr_i^{t_i}.(1-pr_i)^{1-t_i}}{\prod_{i=1}^{k} pn_i^{t_i}.(1-pn_i)^{1-t_i}} \end{aligned} \tag{2.9}$$

And the ranking function in Equation 2.9 can be further reduced through some algebraic manipulation[119], such that:

$$\begin{aligned} \log O(R|d,q) &\propto \sum_{i=1}^{k} t_i \log \frac{pr_i.(1-pn_i)}{(1-pr_i).pn_i} + \sum_{i=1}^{k} \log \frac{1-pr_i}{1-pn_i} \\ &\propto \sum_{i=1}^{k} t_i \log \frac{pr_i.(1-pn_i)}{(1-pr_i).pn_i} \end{aligned} \tag{2.10}$$

The latter term in Equation 2.10 is a constant and can also be ignored for ranking purposes. The term relevance weight (trw_i) assigned to each term is determined by $\frac{pr_i.(1-pn_i)}{(1-pr_i).pn_i}$. The assumption that a term contributes to the relevance of a document independently is required (The Independence Assumption). While this is not justified in reality [148], it has three advantages[140]:

1. the formal expression of the model is made easier,
2. it allows the implementation of the model to be tractable, and
3. it provides a strategy for indexing and searching that improves performance over the simple term matching strategies.

Number of	Relevant	Non-Relevant	Totals
Documents with term i	nr_i	$nd_i - nr_i$	nd_i
Documents without term i	$nR - nr_i$	$nD - nd_i - nR + nr_i$	$nD - nd_i$
Totals	nR	$nD - nR$	nD

Table 2.1: Term incidence contingency table

Since the Odds ratio is employed for ranking purposes, it has been shown that the assumption of independence is actually weaker, and can be described as linked dependence [21]. Though there have been attempts to relax the assumption further which consider term co-occurrence [147, 49], they have not be shown to provide significant improvement over the independence assumption.

The estimation of the term relevance weighting is easily explained with the use of the term incidence contingency table[119]. In Table 2.1, nr_i indicates the number of documents that term i occurs in given that the document is relevant, nd_i is the number of documents containing the term i, nR is the total number of relevant documents, and nD is the total number of documents. The optimal weighting for the probability that a term is present and relevant simply the number of times the term occurs in relevant documents divided by the number of relevant documents i.e. $pr_i = \frac{nr_i}{nR}$. And similarly, the probability that a term is present but the document is not relevant is just $pn_i = \frac{nd_i - nr_i}{nD - nR}$. In the case where the term is not present, $(1 - pr_i) = \frac{nR - nr_i}{nR}$ and $1 - pn_i = \frac{nD - nd_i - nR + nr_i}{nD - nR}$ for relevant and non relevant respectively. The term relevance weight assigned for a particular term i would then be:

$$\begin{aligned} trw_i &= \frac{pr_i(1 - pn_i)}{(1 - pr_i)pn_i} \\ &= \frac{nr_i(nD - nd_i - nR + nr_i)}{(nd_i - nr_i)(nR - nr_i)} \end{aligned} \tag{2.11}$$

However, initially we do not have knowledge of the values of the variables nr_i and nR. Thus different starting assumptions result in different formulations of the term weighting [119]. For instance, Croft and Harper [26] make the assumption that the number of relevant documents nR is likely to be relatively small compared to the number of documents in the collection nD, which results in the approximation of the probability of the presence of a term in a non-relevant document being $pn_i = \frac{nd_i}{nD}$. And given that no

evidence is available to estimate the probability of the presence of a term in a relevant document, then the simplest assumption is to assume that pr_i equals some constant, c. When relevance information does become available then further consideration of the estimation problem is required. This is because the values from the contingency table cannot be used directly for estimation purposes as extreme values maybe assigned to term weightings. For instance, a term that does not appear in a subset of the known relevant documents would be assigned a term weighting of zero. This of course is a rather extreme estimation (i.e is it really impossible?). To provision for the inherent uncertainty in the retrieval of further documents, Robertson and Sparck-Jones[119] suggest a simple modification of Equation 2.11 by adding a phantom count of 0.5 to the central cells in Table 2.1. The relevance weighting assigned for a particular term becomes:

$$trw_i = \frac{(nr_i + 0.5)(nD - nd_i - nR + nr_i + 0.5)}{(nd_i - nr_i + 0.5)(nR - nr_i + 0.5)} \tag{2.12}$$

2.5.3 Optimal Retrieval

In Maron and Kuhns [95], they argue that a retrieval system should rank documents in decreasing order of their probability of relevance to a query. This was eventually formalized by Robertson [120] who called this criterion the Probability Ranking Principle (PRP) and this states that:

> If a reference retrieval system's response to each request is a ranking of the documents in the collections in order of decreasing probability of usefulness to the user who submitted the request, where the probabilities are estimates as accurately as possible on the basis of whatever data has been made available to the system for this purpose, then the overall effectiveness of the system to its users will be the best that is obtainable on the basis of that data.

Robertson showed that the optimality of ranking by the probability of relevance could be guaranteed under the following conditions: (1) that relevance is a discrete variable, either the document is relevant or is not relevant, and (2) that the relevance of a document is independent of the other documents in the collection.

Under these conditions, it can be proved theoretically that optimal retrieval will ensue [10, 115]. Given that the representation of a document is d and the representation of the user's information need as a query q, then from the definition of the PRP we can proceed as follows: Let C denote the cost of retrieving a relevant document, and $\bar{C}$ denote the cost of retrieving a non-relevant document, where the cost of retrieving not relevant documents $\bar{C}$ is greater than the cost of retrieving relevant document C, that is $C < \bar{C}$.

The decision rule for the PRP states that document d_i should be retrieved in preference to document d_j in response to a query q in the ranking if:

$$C.p(R|q,d_i) + \bar{C}.p(N|q,d_i) \leq C.p(R|q,d_j) + \bar{C}.p(N|q,d_j) \tag{2.13}$$

Substituting $p(N|q,d) = 1 - p(R|q,d)$ into Equation 2.13, and performing some algebraic manipulation, we arrive at the conclusions that the our decision rule will only hold if the $p(R|q,d_i)$ is greater than $p(R|q,d_j)$. When this is the case the ranking will be optimal.

$$\begin{aligned} C.p(R|q,d_i) + \bar{C}.(1 - p(R|q,d_i)) &\leq C.p(R|q,d_j) + \bar{C}.(1 - p(R|q,d_j)) \\ \bar{C} + (C - \bar{C}).p(R|q,d_i) &\leq \bar{C} + (C - \bar{C}).p(R|q,d_j) \\ p(R|q,d_i) &\geq p(R|q,d_j) \end{aligned}$$

The above holds even in the case where relevance is defined over a multi-valued or continuous relevance scale. In the case of relevance as a continuous variable $r \in [0,1]$, the probability distribution $p(R|d,q)$ will be a probability density function and the costs C and C' will be replaced with a cost function $c(r)$ [36].

2.5.4 Other Probabilistic Models

The initial attempts to utilize probability theory for Information Retrieval were made [95, 97] as an alternative and theoretically sound approach to the similarity based mod-

els. Since then there has been a steady development in probabilistic models that attempt to estimate the probability of relevance. This initially culminated in the popular Binary Independence Model [119] which was eventually finalized as a landmark retrieval model in 1979 [148]. Further research and development of the probabilistic model has been mainly directed in four areas: (1) Models that attempt to relax the independence assumption (2) Models that utilize Bayesian/Causal inference networks (3) Model free attempts that apply regression analysis, and (4) Non-Classical Logical based attempts.

Techniques developed to relax the independence assumption have been extensively investigated [147, 146, 130, 49]. However, relaxing the independence assumption usually means that more parameters need to be estimated. For instance van Rijsbergern [147] estimated relevance based on term dependencies, this required four parameters to be estimated instead of two. And as such it was deemed that the computational expense involved in capturing such dependencies was too high with respect to performance. Other attempts to capture the conditional dependencies between terms have used an explicit network representation through Bayesian inference networks [146, 130]. This approach generalizes the probabilistic approach to Information Retrieval and allows the integration of various sources of evidence to be combined within the one framework. Alternative approaches to probabilistic models applied statistical regression theory in an attempt to remove the independence assumption altogether, effectively creating a model which only relies on the underlying assumptions implicit in statistical regression theory itself. An instantiation of this approach is Darmstadt Indexing approach [37, 38], though the approach has met limited success because of the need to employ heuristics in order to optimize the model. A distinctly different approach to probabilistic models stems from using non-classical logic and expressing its semantics using probability theory [149]. Many attempts have been proposed that use possible worlds analysis (intentional logic) [150], modal logic [106], situation theory [79] and through the integration of Natural Language Processing with logic [16]. For a more comprehensive overview of the differences between these conventional probabilistic models see Crestani *et al.* [25] and Fuhr [36]. Recently, a new era has dawned for probabilistic models, that of statistical language modelling. However, as we shall examine further

in the next Chapter, the application of statistical language modelling to Information Retrieval does not focus on estimating the probability of relevance. Instead, it asks a different question,

> How likely is it that this document generated this query?

Where the problem is to estimate the probability of a query given a document based on sampling. A meta model based on notion of sampling is the recently proposed Divergence from Randomness framework (DFR)[1]. That is the DFR can be specified such that it is equivalent to the LM approach[1].

2.6 Evaluation of an IR System

Evaluating an IRS is an importance aspect of the discipline[148]. The evaluation of an IRS is to determine how well the IRS satisfies the users, past and future, collectively, and on average, not just individually[144]. Ideally, we would like to test our system on real live users, however, this would be a costly exercise. Instead, a simulated testing methodology is usually undertaken. According to Hull [64], to evaluate an IRS in a controlled fashion the following three requirements need to be fulfilled:

1. An information retrieval test collection, consisting of documents, queries, and the relevance judgements associating which documents are relevant for which query
2. Evaluation measures that provide an indication of the effectiveness of the IRS's ability to satisfy the user's information need, and
3. A means for determining whether the results observed from different systems are in fact statistically different.

In this thesis, we employ the use of TREC[6][47] and some earlier IR test collections which have been specifically designed for controlled experiments and reasonably fulfil the first criteria. In the following subsections, we explain the standard evaluation

[6]TREC is the Text REtrieval Conference

measures used to quantify performance and statistical tests used to gauge whether the performance of one system is better than another.

2.6.1 Evaluation Measures

There are many different measures that can employed to assess the performance of an IRS. Six measurable quantities for gauging the performance of an IRS were suggested by Cleverdon [19]. These included; (1) The extent to which a system contains relevant information (coverage); (2) the average length of time between submitting the query and presentation of the ranked list (time lag); (3) the presentation of results; (4) the effort expended in finding relevant information; (5) the proportion of relevant information actually retrieved in response to a query (recall); and (6) the proportion of retrieved information that is actually relevant (precision). It is the last two that have been adopted as common measures of the *effectiveness* of an IRS [18]. Precision and Recall measure the systems's ability to retrieve relevant information while at the same time withholding non relevant information[148].

Precision is the fraction of retrieved documents that are actually relevant; i.e. the number of relevant documents retrieved nr divided by the total number of documents retrieved N.

$$\mathbb{P} = \frac{nr}{N} \tag{2.14}$$

Recall is the fraction of relevant documents that have been retrieved; the number of relevant documents retrieved divided by the total number of relevant documents nR.

$$\mathbb{R} = \frac{nr}{nR}$$

Sometimes it is preferable to have one measure instead of two. Precision and Recall can be combined using the f-measure[67], where an *a priori* weight $0 \leqslant \delta \leqslant 1$ can be assigned to the relative importance of recall versus precision (See Equation 2.15). If

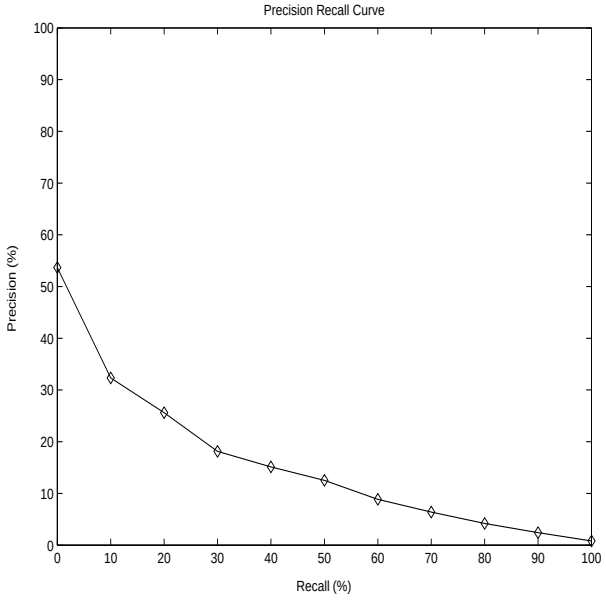

Figure 2.6: Example Precision Recall Graph. Notice the trade off between precision and recall.

equal importance is assigned to precision and recall then the f-measure becomes the harmonic mean of precision and recall.

$$F_\delta = \frac{\mathbb{PR}}{(1-\delta)\mathbb{P} + \delta\mathbb{R}} \tag{2.15}$$

Precision is determined for either a fixed level or fixed number of documents returned, for a particular query given the relevance judgements.

Precision at fixed recall levels By setting fixed recall levels we are able to compute the corresponding precision for each request. Typically, 11 points of recall are selected; $0.0, 0.1, ..., 1$, which correspond to the precision at 0,10,..., 100 percent of the relevant documents.

For overall system performance the precision at each level is averaged over all requests and is often reported in a precision-recall graph (see Figure 2.6). When the Precision and Recall are plotted over eleven points of recall (See Figure 2.6) the curve is characterized as a monotonic decreasing function. This is because there is a trade off between

Precision and Recall, as recall is increased then there will be a decrease in precision, and vice versa. This results because as we transverse down through the ranked list documents are less likely to be relevant, hence the precision degrades.

Unfortunately, we cannot usually estimate the precision at a particular fixed recall level as it may not correspond with the natural recall levels. Hence, interpolation is employed to estimate the precision at fixed levels of recall. See [148] for further details.

Precision at fixed number of documents Instead of using fixed levels of recall, we can specify a cut off point of the number of documents that the user would examine. The precision can then be computed at that fixed number of documents. Usually, it is calculated at the following number of documents, 1, 5, 10, 20, 30, up to 1000. The early values show how well the system performs for applications where high precision is critical(e.g the average web user), whereas 1000 represents the point where the user stops searching.

(Non Interpolated) Average Precision The Non-Interpolated Average is computed by averaging the precision values recorded when each relevant document is encountered whilst traversing the list in decreasing order of relevance. The mean of the non interpolated Average Precision is taken over all queries to give the mean Average Precision (mAP). This statistic is usually taken as a measure of the system's overall performance.

2.6.2 Comparison of IR Systems

Each measure is computed with respect to one query or information need, the results are often aggregated over all queries to obtain the mean performance values. The values must be compared to determine whether one system outperforms the other system. A simple comparison of the mean Average Precision values is not enough to conclude that one system is better than other, especially if there are only very few queries, but it does provide a reasonable indication. However, when there is a reasonable number of queries available then significance testing can be employed[64]. The null hypothesis H_0 is that there will be no difference between the two systems. If H_0 is rejected then

there is a significant difference between the two systems. This implies that one system consistently outperforms the other system. Significance Tests that are applicable include the **paired t-test**, the **paired Wilcoxon Signed Rank test**, and the **paired sign test**[85, 134, 118]. However, the latter makes the least assumptions about the data as it is a non parametric test which only uses the sign of the difference between the two samples. This makes it the most robust test to apply, as the conditions for the other tests are seldom met (see [148] for a detailed discussion on significance testing). Through the course of this thesis, we apply the paired sign test to compare the performance of retrieval experiments.

2.7 Feedback

Feedback with the IR systems allows the user to refine (or re-define) their information need. The feedback can be obtained through various means. The common form is referred to as relevance feedback. This is where the user has judged documents as relevant or not and informed the IRS of their status. Instead of obtaining explicit judgements from users, the top documents returned for a query can be used as relevance feedback, and is referred to as pseudo relevance feedback.

A less clearly defined form of feedback is implicit feedback. Implicit feedback can be obtained from interactions with the IR systems, such as viewing a document or document summary, hovering a mouse over a document link, time spent reading a document, etc[17, 153]. The interactions provide a trail of evidence that may be used to update the query or refine the model of relevance [153].

The information obtained from the feedback is then used to reformulate or redefine the user's information need. For instance, under the BIM model we have already mentioned, we can re-estimate the term relevance weighting with respect to a set of relevant documents. This affects the weighting however often extra terms are added to the query to provide a better description of the information need. This is referred to as query expansion.

2.7.1 Query Expansion

Query Expansion involves the addition of terms to the original query and will usually include re-weighting the query terms as well. One of the earliest methods of query expansion and re-weighting techniques proposed was the Rocchio formula [121]. The new query $q_{new}(t)$ consisted of the original query $q_{old}(t)$ combined with the average weight assigned to terms in the relevant document, and adjusted by the average weight assigned to terms from non relevant documents, such that:

$$q_{new}(t) = \alpha q_{old}(t) + \frac{\beta}{|d \in R|} \sum_{d \in R} n(t,d) - \frac{\gamma}{|d \in N|} \sum_{d \in N} n(t,d) \tag{2.16}$$

where $|d \in R|$ is the number of documents in the relevant set, $|d \in N|$ is the number of documents in the non-relevant set The parameters α, β and γ determine the ratio between the query, and positive and negative feedback, respectively. Automatic query expansion is often employed because initial queries tend to be rather short, 2-3 terms [157]. Other query expansion techniques have been suggested such as Local Context Analysis[157, 98].

2.8 Summary

In this chapter, we have outlined the main components of an Information Retrieval System for *ad hoc* retrieval and introduced the basic models for Information Retrieval. The remainder of this thesis is concerned with probabilistic models, specifically the Language Modelling approach to *ad hoc* retrieval. In the next chapter, we review the area of statistical language models for *ad hoc* Information Retrieval.

Chapter 3

Language Models for IR

The adaptation of statistical language modeling techniques to *ad hoc* retrieval was proposed in 1998, and is typically referred to as the Language Modeling approach. Since then a steady stream of research into Language Modeling has been generated, to the point where it has become widely accepted as an effective and intuitive retrieval model. However, the approach remains controversial as it does not attempt to address relevance, but asks a different question, 'How likely is it that this document would produce this query?'. The resulting probability is referred to as the query likelihood and is used to rank documents. In this chapter, we describe in detail the Language Modeling approach and the subsequent developments of the model, paying particular attention to the aspects that have not been fully addressed or that could be used to inject context into the retrieval process.

3.1 Language Modeling Approach

The three main proponents of Language Modeling were Ponte and Croft [113], Hiemstra [57] and Miller *et al.* [96]. Their attempts have defined Language Modeling for Information Retrieval where the documents are ranked according to the probability of a query given the document. However, the way in which this is derived differs between the proponents. In the following section we present some of the background

about Language Modeling, before introducing the different approaches followed by our formalization of the common assumptions made under these approaches.

Early work on using language models for Information Retrieval was inspired by Statistical Language Modeling (SLM) techniques[93]. The goal in SLM is to predict the next term given the terms previously uttered. This is achieved by developing a generative model based on the underlying data (i.e. the counts of terms and their co-occurrence). This model forms a key component in speech recognition applications[116], where the probability of a term is used in conjunction with audio evidence to decide what term shall be recorded next as the uttered term. An initial attempt to adapt the goal of SLM for IR was in a passage retrieval application[100]. Here the probability of a text fragment (several new terms) given the query (as the previously uttered terms) were estimated and used to rank the text fragments[100]. It was not until 1998 that Ponte and Croft[113] introduced Statistical Language Modeling for document ranking. They adapted the goal so that it would predict the query (new terms) from the document, where the document consists of the previously uttered terms. The score of a document is obtained by estimating how likely the query q would have been produced from the document d (i.e. the probability of the query given the document, $p(q|d)$).

The main assumption engaged by Ponte and Croft [113] is that the $p(q|d)$ is correlated with the probability of document being relevant. They arrive at this conclusion by first assuming that the $p(d|q,R)$ can be approximated by the probability of a document given the query, $p(d|q)$. Then by applying Bayes' rule they obtain the query likelihood $p(q|d)$. The prior probability of a document $p(d)$ and the $p(q)$ are assumed constants and can be dropped for ranking as shown in Equation 3.1.

$$\begin{aligned} p(d|q) &= \frac{p(q|d)p(d)}{p(q)} \\ &\propto p(q|d) \end{aligned} \tag{3.1}$$

The transformation, and hence ranking by $p(q|d)$, is contrary to the approach taken by traditional probabilistic models as this approach ignores relevance. This has led to some controversy about the validity of the Language Modeling approach which we

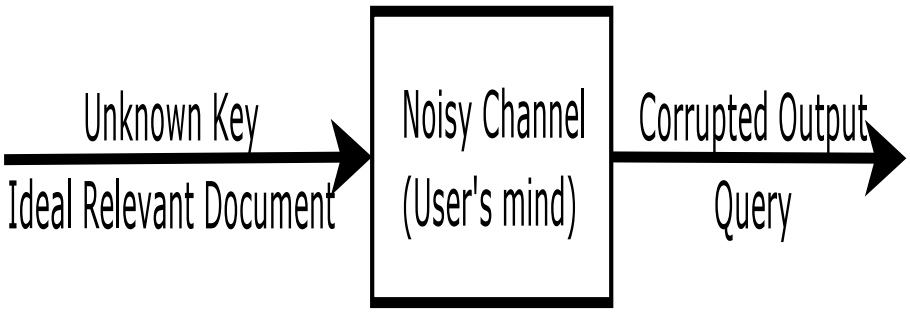

Figure 3.1: Models the corruption of the query.

shall discuss later. However, the intuition that a document is more likely to be relevant if the document is more likely to produce the query terms is very appealing. Ponte and Croft [113] claim that the Language modeling approach offers an explanatory model of retrieval.

This is more obvious when we consider the approach derived by Hiemstra[57], who derives the Language Modeling approach by considering statistical sampling. Sampling is a concept found in most text books on probability theory[118] and usually involves examples involving a bag (document) and coloured balls (terms). The analogy is as follows: Imagine we have a set of documents $d \in D$, where each document d is represented by a bag of terms. First we select a document d with probability $p(d)$. Then from that d, we select a term t_i at random with probability $p(t_i|d)$. We record the term t_i and the replace it back into the bag (i.e. sampling with replacement). We repeat this k times and this becomes our query $q = \{t_1, \ldots, t_k\}$.

We now ask the IR system which document was most likely to have produced this query. Documents are then ranked according to the joint probability of a query and document, $p(q,d) = p(q|d)p(d)$. Since the probability of a document $p(d)$ is assumed to be constant, the scoring for each document is approximated by the query-likelihood $p(q|d)$. This is determined by sampling the query terms from each document. Under this interpretation the notion of relevance is not pivotal to the scoring, and the query likelihood is again assumed to be correlated with the relevance of a document.

Miller *et al.*[96] derive the query likelihood approach from a different point of view altogether. By viewing the process of retrieval as a Hidden Markov Model (HMM)

they formulate the language model as follows: the observed data q, is modelled as being the output produced by passing the document through some noisy channel. The analogy is as follows: The noisy channel is the mind of the user, who is believed to have some notion of the ideal document i that they want to retrieve and translates this notion in the query q. Hence, the probability they attempt to estimate is the probability that d was the relevant document i, given that q was produced. This was expressed as shown in Equation 3.2

$$p(d = i|q) = \frac{p(q|d = i)p(d = i)}{p(q)} \tag{3.2}$$

Up until this proposal, relevance was largely ignored or implicitly assumed within the Language Modeling framework. While this approach considers relevance, it only really considers the case when there is only one relevant document and we wish to find that one. While this is seldom the case, Miller *et al.*[96] advocate that this is a hypothesis; 'Was this the document the user had in mind?'. Hence, documents are ranked in decreasing order of the query likelihood as a means of quantifying how probable this was to the user's ideal document.

So far we have limited discussion about how relevance is represented within the Language Modeling approach because the notion of relevance is still an unresolved issue. It is assumed that the likelihood of generating a query from a document is correlated with the relevance of that document[113]. However, this removes the focus from relevance and requires model specific assumptions to be made.

3.2 Assumptions of Language Modeling

This section presents the underlying assumptions of the Language Modeling approach. From the initial approaches advocated[113, 57, 96], we have surmised three specific assumptions. We state them as follows:

A1 **Correlation** The probability of a query given a document is *correlated* with the probability of a document being relevant[113, 57] . Stated, more firmly, the

probability of a query given a document is *proportional* to the probability of the document being relevant[78].

A2 **Unification** The data model and the retrieval function are one and the same. This is because relevance is subsumed by the document modeling process (data model)[113, 78] and shall become more apparent during the course of our review.

A3 **Discrimination** The terms that a user submits as a query will be sufficient in discriminating relevant from non relevant documents[113, 96].

3.2.1 Assumption One - Correlation of Relevance

The assumption that the relevance of a document is correlated with the likelihood of the query being generated from that document does not appear to be that radical on the surface. Intuitively, we would expect the query terms to be prevalent in the relevant documents, and not so in non-relevant. i.e a good match on query terms *implies*[1] relevance[139]. In the approaches of Ponte and Croft [113] and Hiemstra [57], relevance was assumed to be correlated with the relevance of a document. Typically, probabilistic models have considered relevance as a central notion, and the presumption is that it should be explicitly defined and modelled. The implicit nature of relevance within the Language Modeling approach has therefore attracted some criticism (see [139] for a full account). Such as; how does the language modeling approach handle relevance feedback, without the notion of relevance? Or when relevance was considered in Miller *et al.*[96] with the guise of the ideal document, the language modeling approach assumes that there is only one possible relevant document. That is, the document that generated the query (ideal document). Therefore, how are further relevant documents considered[139]? These are interesting issues which have not been fully addressed and represent just some of the challenges facing Language Modeling for IR.

Such criticisms have been taken seriously and an explicit definition of relevance in the Language Modeling framework has been offered by Lafferty and Zhai[78]. Instead of

[1] Implies as opposed to infers.

assuming a correlation between the relevance of a document and the probability of a query given the document, they claim that it is actually proportional. We present their arguments below. As with traditional probabilistic modeling the log odds ratio forms the basis of the ranking and computed by an approximation. However, Bayes' Theorem is applied differently than in the traditional approach and they arrived at Equation 3.4. Mathematically the different decompositions are equivalent at the point(i.e Equation 3.4 is equivalent to Equation 2.5). However, to proceed to the query likelihood approach from Equation 3.4 two sub assumptions are required.

$$\begin{aligned}
\log O(r|d,q) &= \log \frac{p(R|d,q)}{p(N|d,q)} && (3.3)\\
&= \log \left(\frac{p(q|d,R)p(R|d)}{p(q|d,N)p(N|d)} \right) && (3.4)\\
&= \log \frac{p(q|d,R)}{p(q|d,N)} + \log \frac{p(R|d)}{p(N|d)} && (3.5)\\
&\propto \log \frac{p(q|d,R)}{p(q|N)} + \log \frac{p(R|d)}{p(N|d)} && (3.6)\\
&\propto \log p(q|d,R) + \log \frac{p(R|d)}{p(N|d)} && (3.7)\\
&\propto \log p(q|d,R) + \log \frac{p(R)}{p(N)} && (3.8)\\
&\propto \log p(q|d,R) && (3.9)
\end{aligned}$$

A1.1 The document and query are assumed to be independent given the event of non-relevance i.e. $p(d,q|N) = p(d|N)p(q|N)$ (applied in Equation 3.6). Hence, the $p(q|N)$ can be ignored from the ranking because it is assumed to be constant (Equation 3.7).

A1.2 The probability of a document and relevance(or non relevance) is independent, i.e. $p(d,R) = p(d)p(R)$ and $p(d,N) = p(d)p(N)$ (applied in Equation 3.8). The prior of relevance and non-relevance is also ignored from the ranking because it is again assumed to be constant.

Ultimately, they claim that the log Odds Ratio is proportional to the query likelihood of a document and relevance. This is a much stronger claim than just correlation, one which may not be entirely justifiable. We consider the assumptions they make to

ascertain why.

Their first assumption A1.1 is based on the belief that query terms are only likely from relevant documents and not non relevant documents which is fairly believable and acceptable most of the time. However, when terms have multiple meanings this is not likely to be the case. For instance, if a term is from a document that uses a different sense then this premise would be violated.

Lafferty and Zhai's second assumption A1.2, however is rather more questionable and presumably was made for convenience [2]. Dispensing with the dependence between a document and relevance (non-relevance) seems to be rather inappropriate. Implicitly, the notion of relevance is linked to the document, i.e. either it is relevant or not. In the relevance based language models, this dependency is exactly what is used to score documents (See Section 3.5.5). However, if we are happy to accept this assumption then the inclusion of relevance within the language modeling approach becomes implicit within the document language model[3]. With relevance on the document modeling side, there is now a greater reliance on the document language model to be appropriately estimated/modelled. This is made explicit through assumption A2.

Our interpretation of the query likelihood considers a different and simpler explanation of the correlation in A1. We posit that the joint probability of a query and document can be expressed by the summation over the binary variable relevance of joint probability of query, document and relevance (See Equation 3.10). After re-expressing the joint probability on the right hand side and then dividing both sides in Equation 3.11 by $p(d)$, we obtain the query likelihood $p(q|d)$.

[2]Note: Within their paper they do not provide any rationale for this assumption.

[3]This seems to imply that all the documents are considered relevant, and that the query likelihood will tell us just how relevant they are. This can be restated as the Orwellian Retrieval Model: All documents are relevant, but some documents are more relevant than others.

$$
\begin{aligned}
p(q,d) &= \sum_{r\in(R,N)} p(q,r,d) && (3.10)\\
&= \sum_{r\in(R,N)} p(q|r,d)p(r,d) \\
&= p(q|R,d)p(R,d) + p(q|N,d)p(N,d) && (3.11)\\
p(q|d) &= p(q|R,d)p(R|d) + p(q|N,d)p(N|d) && (3.12)
\end{aligned}
$$

The query likelihood is composed of two parts, the contribution of the query given the document being relevant and the probability of query given the document being non-relevant, weighted by the prior probability of relevance given a document.

$$p(q|d) \rightsquigarrow p(q|R,d) \tag{3.13}$$

Equation 3.13 depicts the correlation where the strength of this correlation will depend on how well we can account for the other parts (i.e. the $p(q|N,d)$).

As mentioned above, the query likelihood approach relies on matching query terms to imply what is relevant. However, this property can be easily violated. For instance, a user unwittingly submits a query with terms that do not occur in relevant documents (i.e the vocabulary problem). This could occur when the user is unsure of language contained in the relevant documents, or it is domain specific. Alternatively, the user may not know how to formulate their query such that the terms they use in the query match the terms the author of the document used. Under such circumstances, we would anticipate that the correlation in Assumption One would not hold. This is why Assumption Three is required.

3.2.2 Assumption Two - Unification

Assumption One places the responsibility of handling relevance with the document language modeling process (i.e the process generating the data). How the document is modelled will directly influence how it is scored with respect to a query as there is no

distinction between the representation of the document (data model) and the matching function (retrieval model) under the language modeling approach. Restated, the data model and the retrieval model are one and the same, or unified[4]. The benefit is that two separate set of inferences for indexing and retrieval are no longer required[113].

Therefore, it is crucial that the estimation of the document model be taken seriously. The document models needs to be an accurate representation of the underlying data but also must consider the user's understanding of the collection. This is a further assumption of the model (A2.1). A2.1 assumes that the user has some understanding of the distribution of terms with in documents. This is also required by Assumption Three because how the user considers the documents in the collection and will influence the query terms that they will choose. The documents need to reflect this understanding and encode this with in the document model. By doing so, it was posited that building better representations of the underlying document models with respect to the user's understanding of the document should obtain better retrieval performance[112].

3.2.3 Assumption Three - Discrimination

This assumption cast according to Ponte and Croft[113] asserts that the user will choose query terms that will sufficiently distinguish between relevant and non-relevant documents[112]. A similar assumption is made implicitly by Miller *et al.* [96] from their ideal document analogy. If the user can imagine an ideal document and the terms used within such a document, then they should be able to select query terms which will be likely to occur in this document. Presumably, these query terms will discriminate the relevant documents sufficiently to separate them from the non relevant documents. This assumption can be considered from two points of view:

A3.1 The user will issue query terms that are highly discriminative, i.e will identify relevant from non-relevant, or

A3.2 The user will issue query terms that are highly likely in relevant documents.

[4]Throughout the course of this thesis we may interchange between data model and document model depending on the context, but meaning is the same as in all cases the data are the documents.

For the user to be able to select terms that would satisfy either sub assumption, then it is required that A2 and A2.1 hold(i.e the document models reflect the user's understanding of the terms within documents). A3.1 assumes that the user is more intimate or familiar with the collection and able to select highly discriminative terms, while A3.2 assumes that the user is able to identify common and general terms but is not so familiar or intimate with the collection and hence submit terms that would be highly likely in the relevant document(s).

An implication of A3 is that the query must consist of key terms that are likely to have come from the relevant document(s) and not a description of the information need[112]. This is because there may be a mismatch in terms the author uses in describing the information and the terms used to describe the information need. It is further assumed that the user will in fact choose terms that are more likely to occur in relevant documents than non relevant documents[112]. Under this assumption there should be reasonable discrimination between the query likelihood of relevant documents and query likelihood of non relevant documents. If the queries are of such quality then we believe that this will produce a correlation between the query likelihood and the relevance of the document (i.e uphold A1).

A summary of the assumptions is provided in the Appendix A. In Chapter 5, we provide an analysis of these underlying assumptions to see if they hold in practice and to what extent. However, our core contribution focuses on Assumption Two, where we attempt to develop context based document models which reflect the user's understanding in Chapter 4 and Chapter 6.

3.3 Query Likelihood Approaches

Regardless of the proponent, the scoring is determined through computing the query likelihood $p(q|d)$. So far we have not discussed how the probability of a query given a document is actually determined by computing the probability of a query given the document model $p(q|\theta_d)$. The reasoning for this will be made apparent in Section 3.4.

The document model θ_d provides a representation of the underlying data in d and is defined as a *multinomial probability distribution* over the discrete sample space over the vocabulary $|T|$. The probability for each term given the document model is defined as $p(t|\theta_d)$. It is assumed, unless stated otherwise, that the query terms are drawn identically and independently (i.i.d) from the document model. While this is not the case in reality, as the meaning of a word is dependent on its context, it is a reasonable starting point and an acceptable approximation[96]. This independence assumption states that every possible order of the terms has the same probability, regardless of position[57].

In the sampling example, the query terms were randomly drawn. This implies that the query generation process is assumed to be a random process[113]. Again this is not strictly the case: however, from the IRSs perspective, the process appears random because it has knowledge of the query generation process. Similarly, the document language generation process is also treated as a random process.

Aside from the differences in interpretations[113, 57, 96], the next main difference when estimating the query likelihood is the treatment of the query. It is either treated as a set[112], a sequence[57] or a distribution[96]. Below, we describe each of the approaches' specific implementation of the query likelihood approach.

In Ponte and Croft[113], they consider the query as a binary vector of terms, where term t is either in the query q or not. Treating the query as such leads to a *multiple-Bernoulli* view of the document model. The query likelihood is therefore composed of two parts, the probability of the query terms occurring in the document, and the probability of the terms not occurring in the query also not occurring in the document.

$$p_{set}(q|\theta_d) = \prod_{t \in q} p(t|\theta_d) \prod_{t \notin q} (1 - p(t|\theta_d)) \tag{3.14}$$

The idea is that if a document discusses lots of issues that are unrelated to the query topic then the document is probably less relevant than a document that predominantly covers the query topic. This provides a normalization component to the ranking, but is computationally expensive as all terms in the vocabulary require to be scored per

document, not just the terms that appear in the query.

In the sequence based approaches[55, 96], the query is represented as a sequence of terms $q = \{q_1, \ldots, q_k\}$. The $p(q|\theta_d)$ is the joint probability of all the query terms occurring in the document model (See Equation 3.15). It is generally assumed that the terms are drawn identically and independently from the document model, resulting in the multiplication of the probability of each query term given the document model (See Equation 3.16).

$$p_{seq}(q|\theta_d) = p(q_1, \ldots, q_k|\theta_d) \tag{3.15}$$

$$= \prod_{i=1}^{k} p(q_i|\theta_d) \tag{3.16}$$

Treating the query in this way represents a *multinomial* view of the document model. The assumption of independence means that order is not considered, and so the query can be represented as an empirical distribution $n(t,q)$ which represents the number of times term t occurs in query q. The probability of a query given the document model can now be expressed as shown in Equation 3.17, which is equivalent to Equation 3.16.

$$p_(q|\theta_d) = \prod_{t \in q} p(t|\theta_d)^{n(t,q)} \tag{3.17}$$

This representation is the form used throughout the remainder of this thesis, unless otherwise stated. For computational issues and mathematical convenience the log of the query likelihood is usually taken. This does not affect the ranking as the log function is a monotonic transformation, but ensures that the multiplication of very small probabilities can be computed, through the summation of the log probability (See Equation 3.18).

$$\log p(q|\theta_d) = \sum_{t \in q} n(t,q) \log p(t|\theta_d) \tag{3.18}$$

A similar ranking functions proposed by Ng[105] used the ratio $\frac{p(q|\theta_d)}{p(q)}$ as a measure of the similarity between document and query. This normalizes the query likelihood and is referred to as the log likelihood ratio (LLR).

In the following section we describe the different document modeling techniques.

3.4 Document Modeling

Recapping, the basis behind the LM approach is that we infer a language model for each document (in the collection) and rank according to our estimate of generating the query from that model. An estimate of the probability of a query q given the language model of document θ_d, $p(q|\theta_d)$, is therefore required. In order to calculate this, the usual assumption that terms are independent is made. The calculation of the ranking for each document is obtained by taking the product over the query terms: $p(q|\theta_d) = \prod_{t \in q} p(t|d)^{n(t,q)}$ where the $p(t|d)$ is the maximum likelihood estimate of the term occurring in the document d. i.e. $p(t|d) = n(t,d)/n(d)$ where $n(t,d)$ is the number of times t occurs in d, and $n(d) = \sum_t n(t,d)$.

However, the empirical document model has a severe limitation[113]. If the document is missing one or more of the query terms then the document will be assigned a zero probability because of the multiplication of the probabilities. This extreme estimate is undesirable from a probabilistic viewpoint, because assigning the probability of a term given a document as zero is quite a radical assumption. Such an assignment would mean that the event would be impossible. To alleviate this problem it is often assumed that a term is no more likely to occur than the probability of drawing the term by chance, though this may introduce other problems. If a term that occurs in the majority of documents but is not a stop word and does not appear in the document then it is possible that it will have a significant impact on determining the document's relevance[113]. Nonetheless, creating a document model can resolve the Zero Probability Problem (ZPP) by smoothing the maximum likelihood estimates such that the $p(t|\theta_d) > 0$ for all $t \in T$. This represents a departure from the Frequentist view of probability, and in this case towards a Bayesian persuasion, where the document model is estimated according to a set of model parameters[5].

[5]However, there are other ways to deviate from the Frequentist view, which are not considered within this thesis.

In the following subsections, we review some of the smoothing techniques that have been applied to LM for IR. One of the aims of this thesis is to investigate smoothing techniques that use context to generate better representations of the underlying data. According to A2, building better representations should obtain better performance. We defer any discussion of these until Chapter 4, where we present a generic framework for context based document models.

3.4.1 Risk Based Smoothing

Ponte and Croft [113] estimated the $p(q|\theta_d)$ using risk based smoothing. Their approach attempts to smooth the document models in a robust fashion by minimizing the risk associated with adjusting the likelihood of a term occurring in a document. A geometric function was used that can be understood intuitively, if thought of as follows: as the term frequency deviates further from the mean, then the mean probability becomes riskier to use as an estimate. The risk function L is used as a mixing parameter in the estimation of the probability of a term given a document model as follows:

$$\begin{aligned} p(t|\theta_d) &= p(t|d)^{(1-L_{(t,d)})} \times p_{avg}(t)^{L_{(t,d)}} && \text{if } n(t,d) > 0 \\ &= p(t|\theta_C) && \text{else} \end{aligned} \tag{3.19}$$

Where the probability of a term $p_{avg}(t)$ and the risk function for a term in a document $L_{(t,d)}$ are defined as follows:

$$p_{avg}(t) = \frac{\sum_d p(t|d)}{df(t)} \tag{3.20}$$

$$L_{(t,d)} = \left\{ \frac{1}{1+tf_{avg}(t)} \right\} \times \left\{ \frac{tf_{avg}(t)}{1+tf_{avg}(t)} \right\}^{n(t,d)} \tag{3.21}$$

Where $tf_{avg}(t) = p_{avg}(t) \times n(d)$ is the average term frequency of the term t and the probability of a term in the collection model is defined by:

$$p(t|\theta_C) = \frac{\sum_d n(t,d)}{\sum_{d'} \sum_{t'} n(t',d')} \tag{3.22}$$

By smoothing in this way the probabilities are not normalized. So Ponte and Croft calculate the probability of a query given a document as follows:

$$p(q|\theta_d) = \prod_{t \in q} p(t|\theta_d) \times \prod_{t \notin q} (1 - p(t|\theta_d)) \tag{3.23}$$

This is why the query is considered a set of terms as opposed to a sequence of terms. The second part of the expression determines the probability of not producing the terms out of the query and restricts the possibility of modeling phrases within the local context such as those captured by bi-grams or tri-grams.

It was shown on TREC topics 51-100, 202-250 that this approach could outperform the cosine measure using TF.IDF. This was the first piece of evidence to demonstrate that the Language Modeling approach was effective in *ad hoc* IR, though today this model is generally considered obsolete in terms of both effectiveness and efficiency.

3.4.2 Laplace Smoothing

The simplest method in which to overcome the ZPP is by applying Laplace smoothing [88]. This approach adds a phantom count to each term in the document. The estimate for a term given a document model is:

$$p(t|\theta_d) = \frac{n(t,d) + \alpha}{n(d) + |T|\alpha} \tag{3.24}$$

where α is the size of the phantom count[6] (and thus model smoothing parameter) and $|T|$ is the total number of terms in the vocabulary. While this avoids ZPP, there is no reason to believe that each term should be assigned equal additional count(s). Doing so may actually violate the assumption that a user has an understanding of the distribution of terms used within documents, especially when document descriptions are very sparse. An extension of this form of correction is the absolute discounting method [135], which subtracts a small constant from each count and then redistributes the total subtracted count to the unseen terms. The performance is usually considered so poor that it is often not reported as is the case in [160].

[6] Strictly speaking Laplace Smoothing is when $\alpha = 1$. The Lidstone correction allows α to be any real positive number.

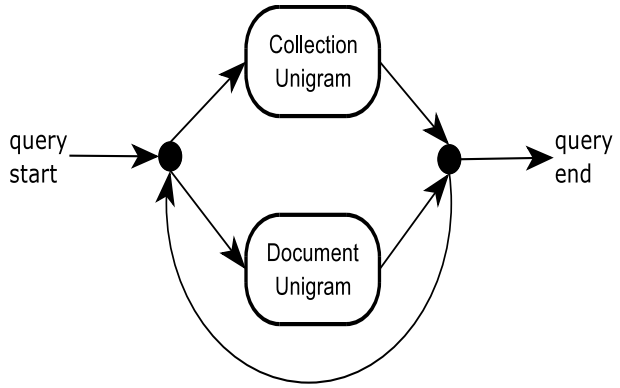

Figure 3.2: Hidden Markov Model for query production. The query is assumed to be generated from either one of two states; the collection or the document unigram.

3.4.3 Jelinek Mercer Smoothing

The Jelinek-Mercer smoothing method [68] is characterized by a sum of its components and is sometimes referred to as linear interpolation or mixture model. This form of smoothing was derived from a linguistic perspective by Hiemstra[55] and from a formal basis using the Hidden Markov Model (HMM) by Miller *et al.* [96]. We present the HMM approach first, and then show the similarity with the former.

In a HMM application the observed or seen data is assumed to be generated as a result of some unknown key being passed through a noisy channel. Miller *et al.*[96] assume the unknown key is the relevant (ideal) document that the user conceives. It passes through the user's mind (the noisy channel) and is emitted as the query (observed data). The HMM is defined by a set of output symbols (terms), a set of state transition probabilities and a probability distribution for each state. The terms are generated by:

1. Starting the process at some initial state
2. Moving from one state to another given the state transition probabilities
3. Sampling from the output distribution at the new state to produce an output symbol;
4. Steps 2 and 3 are repeated until the desired amount of data is generated.

Only the terms are seen by the observer and not the underlying sequence of states that generated them. Hence the name, *hidden* Markov Model[116].

In Miller *et al.*[96], they restrict the HMM to two states; (1) sample from the document, and (2) sample from the collection. The second state represents choosing a term that is commonly used in natural language but that is unrelated to the document. The transition to either state is represented by $(1-\lambda)$ and λ. That is, they assume that the next choice of where to select the term from is generated independently of the previous choices.

$$p(q|\theta_d) = \prod_{t \in q} ((1-\lambda)p(t|d) + \lambda p(t|\theta_c))^{n(t,q)} \tag{3.25}$$

However, in [57], the document and query are assumed to be a sequence of terms. The query is defined by a sequence of terms $q = (t_1, \ldots, t_k)$ that are produced from the document model (where the subscript denotes the position of the term, and refers to a particular term t in the vocabulary). The document model is produced by interpolation with the collection model.

$$\begin{aligned} p(q|\theta_d) &= p(t_1, \ldots, t_k|\theta_d) && (3.26) \\ &= \prod_{i=1}^{i=k} (p(t = t_i|\theta_d) && (3.27) \end{aligned}$$

The $p(t|\theta_d)$ is computed as a mixture model between the empirical probability and the probability of a term given the collection model $p(t|\theta_c)$, as shown in Equation 3.28.

$$p(t|\theta_d) = (1-\lambda)p(t|d) + \lambda p(t|\theta_c) \tag{3.28}$$

In [105], they estimate $p(t|\theta_c)$ using the Good Turing smoothing[42] as it can account for query terms that did not occur in the collection. However, since no documents will contain such terms, it will have little impact on ranking if these terms are excluded from the query.

Hiemstra[55] ran a pilot study on the benefits of using the collection frequency information to smooth the document models on the CRANFIELD Collection and found that

it was better than using a vector space model using term frequency inverse document frequency and document length normalization. Subsequent results have confirmed these findings on TREC collection[96] [162]. Representing the document model as a mixture between the document and the collection is the most popular type of language model, and is usually referred to as the standard language modeling approach.

3.4.3.1 Encoding Term Importance

In [58], Hiemstra argues that users are aware of the terms that they want to see in a document and that a user should have control over the retrieval system. This implicity assumes that the user has some understanding of the terms in the documents and that they can quantify how important they are.

For instance, in the case where the user's query is 'IT Magazine', the retrieval system will normally remove 'it' from the query as it is very common word (and typically a stop word). However, the user may explicitly request that the term 'IT' be present in the document retrieved and returned. The user is effectively specifying how important they perceive the term to be and so each query term should be given its own specific term weighting. This value can also be interpreted as the prior probability that a term is important, $(1-\lambda_t)$ or not important, λ_t.

$$p(t|\theta_d) = (1-\lambda_t)p(t|d) + \lambda_t p(t|\theta_c) \tag{3.29}$$

If the value of λ_t for a particular term is equal to zero, then t must exist in the document, otherwise the document will be assigned zero probability. Conversely, if the value of $\lambda_t = 1$ then the particular term is akin to a stop word and it does not matter whether it occurs in the document or not. Under this approach stop words and mandatory terms can be accommodated and the user has direct influence in the process. It can be shown that as the λ_t values approach zero for all terms in the query then coordination level ranking ensues [58]. When relevance feedback information becomes available then the importance of a term can be estimated (See Section 3.7).

3.4.3.2 Probabilistic Justification for TF.IDF

Under certain circumstances the Language Modeling approach can be formulated such that a relationship with the popular TF.IDF weighting can be obtained[56]. If we assume that the probability of a term given the collection is proportional to the document frequency of the term such:

$$p(t|\theta_C) = \frac{df(t)}{\sum_{t' \in T} df(t')} \tag{3.30}$$

And substitute Equation 3.30 into Equation 3.28 then the document model can be defined as shown below:

$$p(t|\theta d) = (1-\lambda)\frac{n(t,d)}{n(d)} + \lambda\frac{df(t)}{\sum_{t' \in T} df(t')} \tag{3.31}$$

For a particular term in a document, the weighting assigned to it can be transformed without affecting the final ranking by dividing Equation 3.31 through by $\lambda\frac{df(t)}{\sum_{t'} df(t')}$, such that we obtain:

$$p(t|\theta_d) \quad \propto \quad 1 + \frac{n(t,d)}{df(t)} \cdot \frac{1}{n(d)} \cdot \frac{(1-\lambda)\sum_{t'} df(t')}{\lambda}$$

Each component can be interpreted as follows:

- $\frac{n(t,d)}{df(t)}$ is the term frequency inverse document frequency weighting of the term in the document,
- $\frac{1}{n(d)}$ is the inverse length of the document, and
- $\frac{(1-\lambda)\sum_{t'} df(t')}{\lambda}$ is the constant for any term in the document.

It should be noted that by smoothing with the document frequency instead of the collection frequency that two different distributions are being used (term frequency and document frequency). While similar performance may be obtained using the document frequency, the information will not be the same as $df(t) \neq \sum_d n(t,d)$ unless the document is represented as binary vector such that $n(t,d) = (0,1)$. In a similar derivation

with the frequency based $p(t|\theta_C)$, Zhai and Lafferty show that the term weighting is proportional to the term frequency and inverse collection frequency[162].

3.4.4 Bayes Smoothing

The Bayes Smoothing method has been hailed as the best smoothing technique for *ad hoc* Information Retrieval[163] and is sometimes referred to as Dirichlet Smoothing[92]. Bayes smoothing gives the *maximum posterior* estimate of the document model, which is an approximation of the predictive distribution of the full Bayesian inference model [160]. Simply, the method creates the document model by adding a proportion of the probability that the term occurs in the collection to the number of times the term occurs in the document, and then normalized as shown in Equation 3.32 where β is the Dirichlet prior and model parameter.

$$p(t|\theta_d) = \frac{n(t,d) + \beta p(t|\theta_c)}{n(d) + \beta} \tag{3.32}$$

The amount of smoothing applied to each document will be proportional to the document length. This intuitively makes sense as longer documents with a richer description, through having more terms to describe it, will require less smoothing. Shorter documents will attract more smoothing because it is a less reliable sample to base our estimate on. Bayes smoothing can be expressed as Jelinek Mercer smoothing where $\lambda = \frac{\beta}{n(d)+\beta}$ and $(1-\lambda) = \frac{n(d)}{n(d)+\beta}$. Further, the Laplace smoothing method is a specialized case of Bayes Smoothing, where $\beta = |T|$ and $p(t|\theta_C) = \frac{1}{|T|}$.

Results on several TREC collections in [162] showed that Bayesian Smoothing (Equation 3.32) consistently outperformed Jelinek Mercer Smoothing (Equation 3.25).

In Zaragoza *et al.*[160], they derive an analytical form for the *predictive* distribution, instead of using the *maximum posterior* estimate. They employ a standard Bayesian technique of accounting for uncertainty by integrating out unknown model parameters. So instead of using a single point estimate (i.e β), a distribution over β, θ_β, is obtained by combining a prior distribution over the model parameters $p(\theta_\beta)$ with the likelihood

of observing the document defined by $p(d|\theta_\beta)$. Applying Bayes' theorem, we can estimate the posterior $p(\theta_\beta|d)$ as shown in Equation 3.33.

$$p(\theta_\beta|d) = \frac{p(d|\theta_\beta)p(\theta_\beta)}{p(d)} \tag{3.33}$$

The uncertainty is reflected in the values of θ_β, such that if document d is long then the posterior should reflect this by being relatively narrow, whereas if document d is short the the posterior would be broader. The predictive distribution is then defined by computing the probability of query q by accounting for the uncertainty within the posterior through the integral defined in Equation 3.34.

$$p(q|d) = \int_{\theta_\beta} p(q|\theta_\beta)p(\theta_\beta|d)d\theta_\beta \tag{3.34}$$

$$= \frac{1}{p(d)} \int_{\theta_\beta} p(q|\theta_\beta)p(d|\theta_\beta)p(\theta_\beta)d\theta_\beta \tag{3.35}$$

Hence, the query likelihood is obtained by taking the average probability of q over all possible parameter values[7]. The results obtained from using this approach showed that an improvement over the simple Bayes Smoothing could be achieved, though the Jelinek-Mercer Smoothing method outperformed the predictive Bayes on TREC-6 and TREC-8 Collections. These results are contrary to the findings of previous work where Bayes Smoothing was able to outperform Jelinek Mercer smoothing[162]. Hence, it is unclear which method should be applied in order to obtain the best performance.

3.4.5 Other Smoothing Methods

The Statistical Language Modeling literature provides many different smoothing techniques that could be applied to the document modeling process. For example, Katz Smoothing, Absolute Discounting, Leave-one-out discounting, Witten-Bell smoothing [42, 71, 15]. Further more sophisticated latent variable models such as a aggregate and mixed Markov chain[129] or aspect models[62]) could also be applied to the document modeling process. These models can provide a contextual representation of the

[7]Full details of the estimation of this integral can be found in [160].

document. In chapter 4, we employ aspect models in the process of building such representations.

3.5 Variants and Extensions

A number of variants and extensions to the language modeling approach have been developed since its conception. These include, but are not limited to: (1) models which cater for other languages or synonyms in the same language; (2) models which encode higher order term dependencies; (3) models which estimate the document and the query within the Risk Minimization Framework and; (4) an alternate approach which models relevance instead of the document. We provide an overview of these models in the following subsections.

3.5.1 Translation model

Berger and Lafferty[11] approach the problem of generating a query from a document in a different manner by harnessing the advances in statistical translation for IR. Instead of viewing the document as a 'bag of words' and sampling query terms from it, Berger and Lafferty suggest an information theoretic perspective. They assume that when a user has an information need, the user has an ideal document in mind as is the case in [96]. From this ideal document, the user submits key terms as a succinct query. They posit that the process is akin to a translation or distillation of the user's ideal document to a query. As such, it can be viewed that the information need is a signal which becomes corrupted during the process and this corruption is witnessed as a query. The retrieval system is given this corrupted information need and must attempt to retrieve the documents that are most likely to satisfy this need. This is determined by the probability of a query given a document, the difference is in its inclusion of a translation construct, i.e the state translation matrix. A state translation matrix is defined as $p(t|w)$, the probability of term t being translated from term w. For example, in the problem of cross lingual Information Retrieval where the document is in French and

the query is in English, the translation matrix will define the probabilities of a French term w being translated to an English query term t.

When used for the same language, the simplest implementation would be when a term can only translate to itself (self transition). However, the translation probabilities could be computed using thesaurus relationships that spread the probability mass among synonyms. Under the statistical translation model, the query likelihood is obtained by summing over all possible translations of w to t, such that:

$$p(q|\theta_d) = \prod_{t \in q} \sum_{w} p(t|w) p(w|d) \tag{3.36}$$

By using statistical translation methods the model can address the important issues of synonymy and polysemy (the vocabulary problem) which is not possible by simply smoothing the document model. By employing this smoothing strategy we are effectively generating a semantically smoothed document representation [77].

However, the method as implemented in Berger and Lafferty[11] suffered from some drawbacks [163]: (1) The state transition matrix $p(t|w)$ was created by using synthetic training examples as collections of queries and relevance judgements large enough to accurately estimate the translation probabilities was not available, and; (2) the translation probabilities are context independent and are unable to directly utilize the word sense into the model. These limitations spurred on further research that investigated query translation models. In [163], Zhai and Laffery proposed a general probabilistic framework based on risk minimization which was based on Bayesian decision theory (See Section 3.5.3). Other work by Jin *et al.* [69] estimated the translation matrix by creating a training set which paired the title of a document to a document. The title was assumed to be one possible example query translation, in the sense that it succinctly represented the information contained in the document (i.e distilled or translated from document to title).

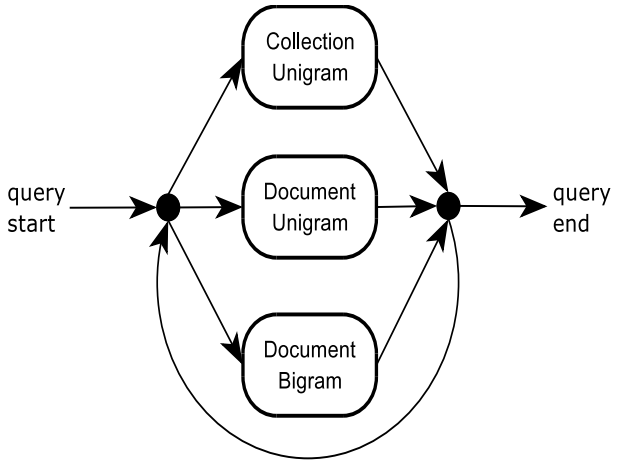

Figure 3.3: Hidden Markov Model for query production. In this case, the query is produced from one of three states; the collection, the document unigram or the document bigram.

3.5.2 Term Dependence Models

Song and Croft[137] propose a language model that combines a range of data smoothing techniques which can be easily extended to incorporate probabilities of bi-grams and tri-grams. They view the query as a sequence of terms, where the probability of a query term is dependent on the past query terms being produced from the document model. The joint probability of seeing the sequence of query terms is approximated using n-gram models:

- Unigram: $p(t_1,\ldots,t_n|\theta_d) = p(t_1|\theta_d)p(t_2|\theta_d)\ldots p(t_n|\theta_d)$
- Bigram: $p(t_1,\ldots,t_n|\theta_d) = p(t_1|\theta_d)p(t_2|t_1,\theta_d)\ldots p(t_n|t_{n-1},\theta_d)$
- Trigram: $p(t_1,\ldots,t_n|\theta_d) = p(t_1|\theta_d)p(t_2|t_1,\theta_d)p(t_3|t_2,t_1,\theta_d)\ldots p(t_n|t_{n-2},t_{n-1},\theta_d)$

The unigram model assumes that terms are drawn identically and independently (i.i.d) and is the basis of the standard language modeling approach, whereas the bi-gram and tri-gram models take the local context into consideration. They estimated the probabilities with the Good Turing estimate [42] and curve fitting to build a smoothed document model consisting of unigrams, bigrams and trigrams. In the unigram case,

using the Good Turing method, the document model is estimated by:

$$p(t|\theta_d) = \frac{n(t,d)^*}{n(d)} \tag{3.37}$$

The raw term frequency scores $n(t,d)$ within a document are adjusted to $n(t,d)^*$, where $N_{(n(t,d)+1)}$ is the number of terms with frequency $n(t,d)$ in a document and $E(N_{n(t,d)})$ is the expected value of the $N_{(n(t,d)+1)}$.

$$n(t,d)^* = (n(t,d)+1) \times \frac{E(N_{(n(t,d)+1)})}{E(N_{n(t,d)})} \tag{3.38}$$

Another smoothing function then is used to calculate the expectation $E(N_{n(t,d)})$[8]. The updated term frequency $n(t,d)^*$ ensures that a non zero probability is assigned.

Evaluation on the Wall Street Journal Collection and TREC4 Collection, showed improvements over the original language model [113] and the INQUERY system by using the combination of uni-gram and bi-gram estimates. Miller *et al.*[96] also implemented a bi-gram model by extending their Markov model to include bi-grams (see Figure 3.3 where the bigram is another state that can be visited). They also confirm that improvements over the baseline unigram model are possible. Other research instead of assuming query dependencies based on query term order, has focused on extracting the meaningful dependencies and attempting to predict that structure from the documents[142, 104, 40].

3.5.3 Risk minimization framework

In an empirical study on smoothing techniques Zhai and Lafferty [162] examined the effect that query length plays in the role of smoothing. Using TREC collections and a range of different smoothing techniques (including Bayes Smoothing and Jelinek Mercer Smoothing) they examined the influence of the smoothing parameter given short and long queries. Short queries consisted of a few terms taken from the title of the TREC Topics. Long queries consisted of the title and the description of the TREC

[8]For further details on how to compute this expectation see [137, 42].

Topics about 50-60 terms in length. They showed that the performance of the system was very sensitive to smoothing parameter value, and that longer queries required more smoothing while shorter ones did not.

They suggested that smoothing plays a dual role in the query likelihood method. One role is to obtain an accurate representation of the document and avoid the zero probability problem, called document estimation. The other is to account for common or non-informative terms in the query, and is referred to as query modeling. To account for the dual role of smoothing they developed a two stage language model[163]. For the document smoothing role they suggest that the Dirichlet Prior method is the best as it caters for the document length. whilst for the second stage they use Jelinek-Mercer smoothing for the query smoothing role where the influence of query model is adjusted depending on the length of query. The model can be mathematically expressed as:

$$p(t|\theta_d) = \lambda \left(\frac{n(t,d) + \beta p(t|\theta_C)}{n(d) + \beta} \right) + (1-\lambda) p(t|\theta_C) \tag{3.39}$$

Where the parameter β is held constant and the λ parameter is adjusted for different queries. The suggested benefits of this model are that for short queries the query smoothing role is insignificant and best performance is obtained from the document smoothing role. For long queries and queries containing unimportant terms however the query smoothing role is required.

The Risk Minimization framework[77], which is based on Bayesian decision theory, formed the basis of their approach which caters for the query, document and relevance. Under this approach, documents are ranked on the risk function **R** shown in Equation 3.40. The query q and documents d are modelled using statistical language models, θ_q and θ_d, respectively. The user's preferences are encoded through a loss function L and relevance is denoted as the binary variable r.

$$\begin{aligned}
\mathbf{R}(d;q) &= \sum_{r\in(R,N)} \int_{\theta_q} \int_{\theta_d} L(\theta_q,\theta_d,r)p(\theta_q|q)p(\theta_d|d,S)p(r|\theta_q,\theta_d)d\theta_d d\theta_q \\
&\approx -p(\hat{\theta}_d|q) \qquad (3.40) \\
&\propto -p(q|\hat{\theta}_d)p(\hat{\theta}_d) \qquad (3.41)
\end{aligned}$$

A series of assumptions are made to derive Equation 3.40, before Bayes' Theorem is applied to obtain the language modeling approach. The probability of a term given the document language model, $p(t|\hat{\theta}_d)$ is then equivalent to Equation 3.39. A full derivation of the two stage model is presented in [164].

In [161], Zhai and Laffery they show how different retrieval models can be derived using different choices of loss functions. For instance, the Risk Minimization framework can be shown to rank documents according to Kullback-Leibler Divergence. The Kullback-Leibler Divergence[75] function is a measure of the cross entropy between two probability density functions, $p(x)$ and $p'(x)$. The KL divergence between p and p' is denoted as $D(p||p')$ and defined as shown in Equation 3.42.

$$D(p||p') = \sum_x p(x) log \frac{p(x)}{p'(x)} \qquad (3.42)$$

The Divergence D is non negative and is zero when $p = p'$. While the measure is not symmetric, it is still intuitive to think of it as the distance between distributions where zero means they are the same. As a measure of similarity between the document model and the query model, it is assumed that the query q is obtained as sampled from the query model θ_q and the document is obtained as a sample from the document model θ_d. If the document and query language models are estimated then the KL divergence of d with respect to q can be measured by Equation 3.43.

$$D(\theta_q||\theta_d) = -\sum_t p(t|\theta_q) log p(t|\theta_d) + \sum_t p(t|\theta_q) log p(t|\theta_q) \qquad (3.43)$$

The query log likelihood function is a special case of the KL divergence model where the query model is estimated as an empirical distribution, as in Equation 3.17.

3.5.4 Parameter Estimation

So far we have discussed some of the different smoothing models that have been applied, but we have not mentioned how the parameters of these models are determined. This is a significant issue in LM and any other IR model. Typically, empirical tuning of the free parameters is performed to find the setting which gives the optimal IR performance. This requires that a set of queries and the corresponding relevance judgments are known beforehand. The challenge is to find the optimal model settings in automatic and unsupervised manner, without recourse to relevance assessments.

One such attempt has been offered by Zhai and Lafferty[164]. Within the risk minimization framework, because both the document and the query are modelled, they can employ statistical estimation techniques to automatically estimate the parameters. They argue that the document model should be estimated based on its underlying data, whilst the query model should be estimated with respect to the individual query to cater for the different types of queries. The parameters for the two stage model shown in Equation 3.39, β and λ, can be estimated as described in the following subsections.

3.5.4.1 Estimating β

When modeling the document our goal is to obtain the best representation of the underlying data possible in accordance with A2. A useful objective function for measuring the representational capabilities of a document model is the 'leave one out' likelihood function. This is, the sum of the log likelihood of each term in the document computed from a model that is constructed on the document with the target term excluded (i.e left out). This process is derived from cross validation and provides a criterion for selecting the best data model available when we take the parameter β that maximizes the $\ell_{-1}(\beta|C)$. The leave one out likelihood of the model for the collection given β is shown in Equation 3.44.

$$\ell_{-1}(\beta, C) = \sum_{d} \sum_{t \in d} n(t,d) \log \left(\frac{n(t,d) - 1 + \beta p(t|\theta_C)}{n(d) - 1 + \beta} \right) \tag{3.44}$$

The estimate of β is shown in Equation 3.45 and can be computed using Newton's method[9].

$$\hat{\beta} = \arg\max_{\beta} \ell_{-1}(\beta, C) \tag{3.45}$$

3.5.4.2 Estimating λ

Estimating the λ parameter is more difficult as what we would like to do is maximize the probability of the query being drawn from the relevant documents and minimize the probability of the query being drawn from non relevant documents. This is not possible as the notion of relevance is not explicitly defined in the query likelihood approach. Nonetheless, an approximation of the λ parameter can be obtained through the application of the Expectation Maximization algorithm, where we estimate the probability of a query given λ.

$$p(q|\lambda) = \sum_{d} \pi_d \prod_{1}^{n} ((1-\lambda)p(t|\theta_d) + \lambda p(t|\theta_C)) \tag{3.46}$$

It is assumed that the query is generated from a mixture of document models with an unknown weight π_d. The parameters λ and π_d can be estimated using the EM algorithm:

$$\pi_d^{(k+1)} = \frac{\pi_d^{(k)} \prod_{t=1}^{n} ((1-\lambda^{(k)})p(t|\theta_d) + \lambda^{(k)} p(t|\theta_C))}{\sum_d' \pi_{d'}^{(k)} \prod_{t=1}^{n} ((1-\lambda^{(k)})p(t|\theta_{d'}) + \lambda^{(k)} p(t|\theta_C))} \tag{3.47}$$

$$\lambda^{(k+1)} = \frac{1}{|D|} \sum_{d} \pi_d^{(k+1)} \sum_{t=1}^{n} \frac{\lambda^k p(t)}{(1-\lambda^{(k)})p(t|\theta_d) + \lambda^k p(t)} \tag{3.48}$$

The parameter π_d is a free parameter introduced because we do not want to maximize the query being generated from all documents, only those that are relevant. Estimating the parameters is performed by running the EM algorithm until convergence. However, this will result in $\pi_d = 1$ for one document d, and all other documents d' will have π_d'

[9] See [164] for further details

assigned zero. This is because one document in the collection will be the most likely to produced that query (this is the ideal relevant document, so to speak). This leads to a poor estimate, so it is recommended that only about ten EM steps be performed to avoid this.

When evaluated on several TREC collections results reported from using this two stage language model show marginal improvement in terms precision (average, initial, and at 5 documents) over various types of queries (long, short, verbose and keyword). However, they suggest that better performance could be obtained by adjusting smoothing parameters, or estimating the query language model from other sources. When parameter estimation was used on the two stage model it obtained results comparable to either the Bayes smoothed document model or the Jelinek Mercer document model. However, the two stage estimated model did not achieve the better performance than an empirically set two stage model. It is interesting to note that the estimation technique did not obtain the optimal performance. In Chapter 5, this observation is investigated through the analysis of the assumptions of the Language Model.

3.5.5 Relevance Model

Instead of using the query likelihood to rank documents, a document likelihood approach was suggested[82]. This approach is more akin to that of the traditional models. The probability of a document given the model of relevance is used to rank documents. The relevance based language modeling approach views relevance as a generative process, which can be modelled as a multinomial term unigram distribution. It is assumed that this model (the relevance model) is where the relevant documents are generated. Documents are ranked according to how likely they are to have been generated from the relevance model, $p(d|\theta_R)$. The Odds Ratio is employed to normalize the score, and the denominator considers the likelihood of the document being produced from the non-relevance model $p(d|\theta_N)$.

$$O(r|d) = \frac{p(d|R)}{p(d|N)} = \frac{p(d|\theta_R)}{p(d|\theta_N)} \tag{3.49}$$

The model was originally proposed by Kalt [70] in the form of a text classifier but formalized for *ad hoc* retrieval by Lavrenko and Croft [82]. Under their approach the PRP is upheld, unlike in the standard language modeling approach.

$$\frac{p(d|\theta_R)}{p(d|\theta_N)} \approx \prod_{t \in d} \frac{p(t|\theta_R)^{n(t,d)}}{p(t|\theta_N)^{n(t,d)}} \tag{3.50}$$

As with the Binary Independence Model[140], the estimation problem exists, rephrased here, 'what is the probability of a term occurring given the relevant set of documents?'. Kalt[70] suggested an IF approach, where the relevance model is developed from a set of example relevant documents. These provided the training examples to estimate $p(t|\theta_R)$. However, in *ad hoc* IR only a sparse query is supplied, and is believed to be insufficient to reliably estimate a relevance model. Lavrenko and Croft [82] suggested a novel approach for the estimation of $p(t|\theta_R)$ that applied Statistical Language modeling techniques in a different manner. The approach assumes that the set of relevant documents and the query have been generated from an underlying relevance model, where as other language modeling approaches assumes that the query is a sample of a specific document model. This notion of relevance is therefore different to that of conventional probabilistic models.

It is assumed that there is a relevance model, described as a uni-gram distribution, from which the query and the relevant documents are generated. To rank documents, the likelihood of the document being generated from the relevance model versus the likelihood of the document being generated from the non-relevance model is computed (see Equation 3.50). Though the relevance model may not necessarily be restricted to a unigram, it may actually be more appropriate for it to be described as a bi-gram or higher order distribution instead. However, since only a small amount of data is available to estimate the sufficient statistics of the relevance model, the unigram distribution (i.e no dependencies) is again a reasonable starting point. In the absence of relevance information regarding the underlying distribution of terms given relevance, Lavrenko and Croft[82] suggest that the top m documents returned from a query submitted to a conventional probabilistic retrieval system, can be used to estimate the relevance

model for that particular query. These top m documents define a set of documents, $d \in D_R$ used for pseudo relevance feedback. They then assume that the probability of a term being generated from the relevance model can be approximated by the conditional probability of observing the term given that the query terms, $q_1, \ldots, q_k$, have just been observed as shown in the following Equation:

$$p(t|\theta_R) \approx p(t|q_1, \ldots, q_k) \tag{3.51}$$

$$= \frac{p(t, q_1, \ldots, q_k)}{p(q_1, \ldots, q_k)} \tag{3.52}$$

They propose two methods to estimate the joint probability $p(t, q_1, \ldots, q_k)$ by;

1. assuming that words from relevant documents are identically and independently sampled from the uni-gram distribution of relevance (Equation 3.53),

$$p(t, q_1, \ldots, q_k) = \sum_{d \in D_R} p(d) p(t|d) \prod_{i=1}^{k} p(q_i|d) \tag{3.53}$$

   and;

2. assuming that query terms are independent of each other, but keep their dependence on the term t (Equation 3.54).

$$p(t, q_1, \ldots, q_k) = p(t) \prod_{i=1}^{k} \left\{ \sum_{d \in D_R} p(q_i|d) p(d|t) \right\} \tag{3.54}$$

Where K is the set of top ranked documents, the probability of the query can be determined by marginalizing $p(q_1, \ldots, q_k) = \sum_t p(t, q_1, \ldots, q_k)$, the probability of a term is $p(t) = \sum_{d \in K} p(t|d) p(d)$ and the probability of a document given a term is $p(d|t) = p(t|d) p(d) / p(t)$ it is assumed that the probability of document is a uniform constant.

The ranking function using the relevance model uses the odds-ratio which means that it satisfies the probability ranking principle. The non-relevance model is estimated as the probability of a term given the collection i.e. $p(t|\theta_N) = p(t|\theta_C)$.

$$O(r|d) = \frac{p(d|\theta_R)}{p(d|\theta_N)} \tag{3.55}$$

$$= \frac{\prod_{t \in d} p(t|\theta_R)^{n(t,d)}}{\prod_{t \in d} p(t|\theta_N)^{n(t,d)}} \tag{3.56}$$

Over a number of TREC collections, the relevance model approach has out performed both INQUERY (TF.IDF) and standard language modeling (Jelinek-Mercer smoothing with collection frequency) significantly and consistently [83]. However the comparison is not fair as the relevance model relies upon the first pass of retrieval to provide it with pseudo relevance feedback information from which to estimate the relevance model.

Nonetheless, relevance models provide a novel combination of statistical language modeling techniques whilst still maintaining the principles behind the traditional probabilistic models by ranking according to the log Odds Ratio of relevance given a document.

3.6 Document Prior

The document prior may be considered with or without being conditioned on relevance depending on the approach used. If considered as the unconditional prior probability of a document $p(d)$, then it represents the probability of sampling the document from the collection. Typically, the document prior is assumed to be uniform, and so does not influence the ranking. This is perhaps the most sensible approach, because *a priori* no document is more likely than another. If we were to assign the document prior according to some feature (i.e document popularity score), then we are assuming that a document is more likely because it is more popular. What we are really assuming this document will be more likely to fulfil any future information needs, regardless of the information need. This is quite a generalization. However, under the interpretation that the prior probability of a document is conditioned on the notion of relevance, $p(d|r)$. Then we are able to inject contextual evidence directly into the model by using features which are correlated to some degree with the document being relevant to determine the prior probability. Under this interpretation it provides an interesting avenue for contextual information retrieval research[122]. Assigning the document priors with respect to the information need (relevance), would allow the user to ingrain their pre-defined bias of what would make a document more likely to be relevant. Such

a mechanism could be performed automatically or implemented as part of the retrieval interface, where the query is no longer just a textual sequence of terms, but also uses a bias component.

Some of the features used to generate document priors have been: (1) document length[96]; (2) average word length[96]; (3) links[73, 52] and (4) time[87, 31], where the intuitions behind the prior is that document is more likely because (1) it contains more terms; (2) has longer words; (3) has attracted more links or (4) is more recent.

Two commonly used approaches are (1) and (3):

- the document length prior where the probability of a document is proportional to the length of the document[96][73]. Such that: $p(d) = \frac{n(d)}{\sum_{d'} n(d')}$. The rationale for using such a prior is that longer documents tend to contain more information, and hence are more likely to be relevant. The results from using such are prior are mixed, increasing and decreasing performance depending on the collection used.

- link structure analysis techniques have been used to derive document priors, instead of use such internal document based features. The intuition is that more popular or well cited documents will tend be more relevant. The simplest is where the document prior is proportional to the number of in links or references that a web page or document has received, i.e. $p(d) = \frac{n(l,d)}{\sum_{d'} n(l,d')}$. More sophisticated approaches have attempted[73] link structure analysis such as pHits[20], PageRank[109] or Scale Free Network[52]. However, success has been limited and increases in performance are often gained by simply re-ranking the documents by the prior, instead of computing the joint probability $p(q,d)$. Where re-ranking is when the top n documents returned according to the query likelihood are then ranked (i.e. re-ranked) according to the document prior.

The feature chosen to bias the ranking needs to be chosen with respect to the information need, reflecting the user's context. Hence, the document prior provides a novel means for extending of the language modeling approach to included contextual evidence. However, one major problem with using the document prior is that it usually

overwhelms the query likelihood which is usually much smaller. This can be to the point where regardless of how likely the query is form the document, another document with a greater document prior will be ranked higher. In practise the document prior and the query likelihood are often interpolated, or normalized to compensate for the imbalance.

3.7 Feedback

Using relevance feedback in the LM framework is performed through query expansion and term re-weighting, as oppose to updating a model of relevance. This is because there is no explicit definition of relevance within the LM[139]. In contrast, the Relevance Models can directly encode any relevance feedback by re-estimating the probability of a term given relevance. Also, the BIM can use relevance feedback to re-estimate the probabilities of relevance as defined by Equation 2.12 or to expand the query, but usually both allowing control over the definition of relevance and the query being issued, something neither Language Models or Relevance models can claim.

For the LM approach, several feedback techniques have been proposed [58, 112, 161], though they are considered as being fundamentally heuristical and at worst *ad hoc* in nature because the modification of the query modifies the intent of the need.

- However, Ponte[112] justifies the use of query expansion as a natural progression within the Language Modeling framework. He argues that because of Assumption Two and Three the user can select query terms that are likely to occur in documents that are of interest to them. As a consequence, terms for expansion can be selected from the documents which the user has expressed interest in (i.e the set of relevant documents d_R) , by selecting the terms with the highest average log-odds ratio. The odds of a term being interesting is determined by Equation 3.57 and is used to rank terms for the expansion.

$$O(t) = \sum_{d \in d_R} \log \left\{ \frac{p(t|\theta_d)}{p(t|\theta_c)} \right\} \tag{3.57}$$

An additional k query terms are selected and appended to the original query. The expanded query is then used to re-rank the documents.

- Using the Language Model which encodes term importance in Section 3.4.3.1 then relevance feedback information can be incorporated into the model as described in [58]: Given a set of relevant documents, D_R the weights λ_t for each term in the query can be estimated via the EM algorithm. The algorithm will iteratively maximize the probability of the query given the relevant documents. The Expectation (E) step at time p and then Maximization (M) step at time $p+1$ are defined as follows:

$$m_t^p = \sum_{d \in d_R} \frac{(1-\lambda_t^p).p(t|d)}{\lambda_t^p.p(t) + (1-\lambda_t^p).p(t|d)} \tag{3.58}$$

$$\lambda_t^{p+1} = \frac{m_t}{|d \in D_R|} \tag{3.59}$$

  where $|d \in D_R|$ is the number of relevant documents.

- Zhai and Lafferty [161] also proposed a method of feedback. They exploit the benefit of using the KL Divergence function in Equation 3.43 where there is separate document and query model. Relevance feedback information is used to update the query model. They suggest a simple interpolation of the original query model and the average of the relevant documents. The resulting uni-gram distribution is assumed to better represent the user's underlying information need. This method exploits the benefits of two language models, one to represent the document and one to represent the user's need.

3.8 Challenges for LM

There are some issues and general challenges that the Language Modeling approach needs to address. A non exhaustive list of some of these are:

- **Relevance** The issue of relevance within the language modeling framework requires clarification. The different proponents offer different opinions. However, the implicit definition of relevance in the standard language modeling approach

is troublesome because it is unclear whether the approach will uphold the PRP and how relevance feedback information should be applied.

- **Assumptions** The assumptions of language modeling have not been validated to determine whether they actually hold in practice, and to what extent. If the assumptions hold, then we should be able to use them to extend the model to deliver improved retrieval performance, or estimate the parameters of the model in an unsupervised fashion[164] such that the best retrieval performance is obtained. This thesis attempts to validate the assumptions of Language Modeling in Chapter 5.

- **Context** Incorporating contextual evidence within the retrieval process is a current research challenge amongst the members of the IR community generally. How to encode and use contextual evidence for the benefit of retrieval is a difficult problem, often the context works in specific instances only. Contextual Information Retrieval requires more than just a specific model that caters for only one situation, but more customizable solutions, which are flexible and can adapt to the desires and needs of the user. Within the Language Modeling framework there are many avenues for embedding context, in Chapter 6 we outline our attempt at do so, before providing experimental results of employing different forms of context in Chapter 6.

- **Multimedia** In using Language modeling for other forms of data the challenge lies in the representation of non-textual parts of documents. Of course, textual descriptions can be attached to such parts, but this avoids the problems as opposed to directly handling the various media, perhaps as the potential for integrating multi modal data already exists in speech recognition models as they combine evidence from a language model and an acoustic model (text and sound). However the problems of dealing with multi-modal data apply not only to language modeling but to all IR models.

- **User Interaction** The integration of user activity and interaction into the framework, and not just in terms of a principled approach to relevance feedback for the LM approach, is a current challenge for LM. The Language Modeling approach

provides excellent avenues by which interaction can be incorporated. For example, Hiemstra [58] suggests term specific weighting as a simple solution to better capture user needs, whilst Zhai and Laffery [163] suggest a user language model to model their query generation procedure. Whether the inclusion of information such as user profiles, analysis of browsing activity, structured/weighted queries or relevance feedback information can make a significant improvement in terms of IR performance still requires further investigation. And such investigation requires user testing, not just simulated tests on test corpora.

3.9 Other Applications in IR

Over the last seven years, the application of Language Modeling techniques to Information Retrieval has been studied and applied to a variety of other tasks besides ad hoc retrieval. Amongst others, these include: (1) in a distributed retrieval setting the selection of a resource or database is determined by the probability of a query the resource or database[155]; (2) the tracking and detection of topics[72] where the likelihood ratio[105] is used to determine the odds of the topic being produced from the new incoming document; (3) the summarization of text, where we wish to produce a summary of the document, terms are sampled from the document in much the same way that query terms are drawn[99], sentences with the highest likelihood are selected as possible summarizations; (4) translation models[11] have facilitated cross lingual retrieval[156]; (5) structured document retrieval by modeling the structure of a document as different weighted components such as title, abstract and paragraph[108] and (6) more recently a proposal to use language modeling techniques at each stage of the retrieval process for indexing, retrieval and feedback with parsimonious language models [59]. The success of the LM approach can be attributed to the explanatory nature of generative modeling techniques coupled with its simplicity and effectiveness.

3.10 Historical Notes

The statistical modeling of language began in the early 20th century when Markov attempted to model letter sequences in Russian literature [94]. The most notable application was by Shannon, who modelled sequences of letters and words to illustrate the implications of coding and information theory [132]. This spurred on attempts to use statistical language modeling for natural language processing [93]. Areas of marked success where such techniques have been applied are automatic speech recognition [117], parts of speech tagging [27], named entity identification [14], topic segmentation and event tracking [43, 159]. The emphasis in these models has been on predicting the next term given the preceding sequence of terms. Information Retrieval researchers have recently adopted statistical language modeling as a means of ranking [112, 57]. While they do not necessarily require knowledge of the next term given a sequence, statistical language modeling provides a mechanism for a smoothed estimate of the probability of a term given the document. Since 1998, and drawing on the wealth of literature from statistical language processing and machine translation, many models for IR have been proposed [11, 57, 112, 69, 77, 137, 161, 163]. While the current stream of research has been quite prolific, the idea of modeling language for IR purposes is not new. An early attempt focused on using the term frequency as the probability of a word appearing in a document. The renowned implementation of such a view was the Poisson model by Bookstein and Swanson [13] and Harter [50, 51]. Their conscious treatment of term frequency as the basis of the underlying probability distribution, as opposed to the normalized term frequency as an estimate of the probability itself, that is the distinguishing feature between earlier work and the later work on language modeling [44].

3.11 Summary

In this chapter we have presented the language modeling approach and the major developments of the model. First, we described the different approaches to Language Modeling applied to *ad hoc* Information Retrieval. This included our summary of the

common assumptions that underpins the Language Modeling approach. The different document modeling techniques were then described along with some of the further developments and extensions of the model. The chapter was concluded with an overview of the history of modeling language, the alternative applications of Language Modeling and the current research directions and challenges for Language Modeling.

Through the course of the chapter we identified several areas where contextual information could be incorporated within the Language Modeling framework. These include the dependencies between terms modelled as bi-grams or tri-grams, the different word senses modelled through a translation matrix, the inclusion of external document knowledge using a document prior, or generating better document representations.

In this thesis, we investigate the latter, and use the context associated with a document to achieve better representations of the documents. This is motivated by the underlying assumptions of the LM approach, where it is posited that building a better representation of the document should lead to improvements in performance. In the next chapter, we present our framework for 'context based' document models.

Chapter 4

Context Based Document Models

In Chapter 1, we proposed the context hypothesis. In this chapter we offer an implementation of this hypothesis within the Language Modeling approach to *ad hoc* Information Retrieval. We argue that the assumptions of the LM approach justify encoding context within the document modeling process. This is because document models should be constructed with respect to the user's understanding as implied by Assumption Two. We use the semantic associations between documents to quantify the user's understanding of the documents in the collection (i.e. their context). This context is represented as a distribution over the vocabulary and encoded within the document model to form a context based document model. We provide a generic framework for building context based document models, which includes the estimation of model parameters.

4.1 Introduction

Central to this thesis is the notion of context. In Chapter 1, we hypothesized that, *Semantically associated documents tend to be relevant to the same request*. To explore this hypothesis, we need to define what we mean by context, how we are going to represent this context, and how we can encode this context within the LM approach. We focus on the latter, where we impose two constraints:

- it must be consistent with probability theory to ensure that the model is sound, and
- it must be consistent with the underlying assumptions of the LM approach.

As we have already mentioned in this thesis, we restrict our view of context to the semantic associations between documents. We assume that these associations form part of the user's understanding of the documents in the collection. i.e how the documents relate to each other. However, we best qualify what we mean when we refer to the user's understanding.

In this chapter, we consider the User's understanding in a collaborative sense, as a shared or common understanding of the collection held by the users of that collection. We feel that this is a reasonable assumption to make as there are many such instances where a shared view of the collection is held. A classic example is the Dewey classification scheme used within libraries (See Figure 4.1). Users of a library are 'forced' to conform to the particular arrangement of books, etc, according to the hierarchical topic structure defining the categories and so forth.

Hence associations may be pre-defined ontologies, classification schemes, or if we consider time and interaction, structures developed through collaborative interactions with the collection. Alternatively, the associations may arise from some form of structure within the collection or even document. For example, in passage/element retrieval, an element is associated with other elements as defined by the structural layout of the document.

Or associations may be dynamic relationships, changing over time, formed as a result of interaction with the collection. For example, the detection and tracking of a particular topic in the Wall Street Journal, web links placed between web pages or citations between scientific papers. With citations and links, it is important to note that the direction will have an influence on its semantic meaning. For example with web links, in links (links coming from other pages) and out links (links going to other pages) will invoke a different context. Hence, the context (or semantic association) should be chosen appropriately and with respect to the user's information need.

00-099 **Generalities**
004-006 - Computer Science
020-029 - Library and information sciences
070-079 - Journalism and publishing

100-199 **Philosophy and related disciplines**
150-159 -Psychology

200-299 **Religion**
220 - Bible
230-289 - Christianity
290 - Other religions

300-399 **Social sciences**
301-319 - Sociology
310-319 - Statistics of the social sciences
320-329 - Political science

Figure 4.1: Extract of the Dewey Classification obtained from the Birkbeck Library (University of London) website. Quite quickly users of the library learn which section they are likely to find books containing relevant information (though not the book itself).

These are just some of the different types of semantic associations available within the IR domain. There are potentially many other associations that could be used, such as co-activation, time of publication, location of information, etc, the correct context will depend on the information need. Implicitly, the context of the document will affect the user's understanding of the document (i.e how the document is perceived or understood, that is, in context) and subsequently the model of that document needs to reflect this. Once we formalize the understanding through such semantic associations, we can then use it to define a context based document model. This is exactly what is prescribed by the assumptions of the language modeling approach. The Second assumption of the LM approach provides an avenue for encoding such contextual evidence within the LM framework. It implies that document models should be built in accordance with the user's understanding. By developing such contextual based document models, we hope to offer an implementation of the context hypothesis and assess its validity.

As we have already noted in Chapter 3, the assumptions of LM are interrelated. Restated, the user has an understanding of the distribution of the terms within the documents in the collection (A2.1). When formulating the query they are able to select terms that would discriminate relevant documents from non-relevant documents, or at least identify terms that are likely to occur in relevant documents (A3). These assumptions require that the document models are representative of the generating process, which consider the user's understanding of the collection (A2, A2.1). The user's perception of the document is important because what they associate with that document will affect what terms they will consider useful when issuing a query. Indeed, Ponte[112] claims that by providing a better representation that considers the user, improved retrieval performance is possible. The Assumptions, A2 and A3, emphasize the user and their perception of the documents in the collection, which are represented as the distribution of the terms within documents. The challenge, therefore, lies in constructing document models that reflect the user's understanding, yet still remain statistically accurate representations.

4.2 Modeling the Context

Before we can construct the document model, the associations, whether they are topical associations, citations between documents, etc, will reflect how the user views and understands the particular document. Therefore, we need to model the context associated with each document. We do so through the content of the associated documents represented as a multinomial term distribution. This will denote the user's understanding of the context for that document. We shall refer to this as the context background model (XLM) (where this is conditioned on a particular document, however many documents may share the same XLM depending on the semantic association.) In the previous chapter the majority of language models proposed relied on the collection background language model (CLM) to smooth the documents.

Intuitively, we can think of the context as helping to imbue the document with topically related terms. Consider an article about the latest sports car. The context of the article is the car magazine. However, this magazine is just one of many in that genre, and there are also many genres which define the background collection. Intuitively, we would expect that the context will provide a better indication of the terms that would be used in the document than just resorting to the background collection. For instance, we would expect the terms like, 'hot', 'machine', 'motor', 'fast', 'speed', would occur more frequently in the car magazine context , than in the collection background model.

The collection background model represents the most naive context background model where no context or bias is used in the document modeling process. Context, defined by the similarity of documents is a content based approach, and is an instantiation of the *Cluster Hypothesis*[67], whereas context defined by semantic associations is a context based approach and is an instantiations of the *Context Hypothesis*.

The following section defines how we formally represent the context of a document within the language modeling approach by providing a process or framework for building context based document language models.

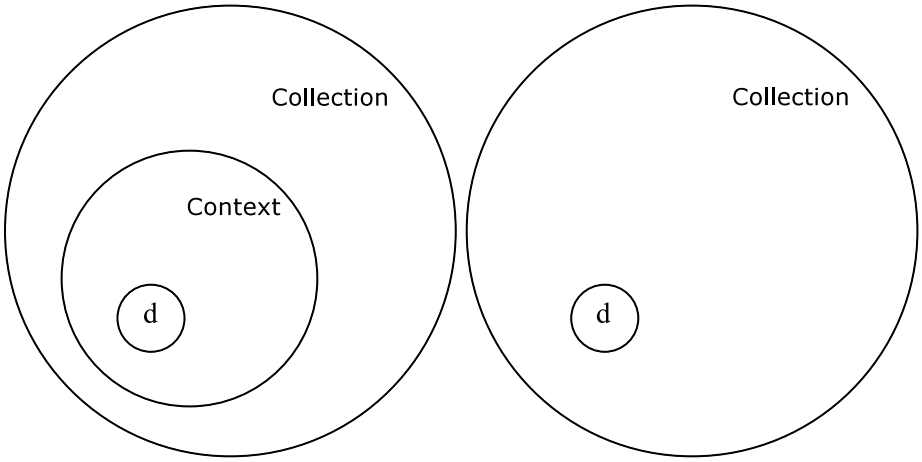

Figure 4.2: Left: The context associated with the document in the collection, where the context is defined by the set of documents in the collection which are related to d. Right: Without any context associated with the model.

4.3 Framework

So far we have discussed in general terms the intuition behind generating documents with respect to the user's understanding. The framework for incorporating this contextual evidence in the process of retrieval to generate context based document models is as follows:

1. Select the semantic association X that is appropriate to the user's information need.
2. Define the context through a set of parameters Θ_X. This set includes the definition of probability matrices that instantiate the semantic associations amongst documents.
3. For each document d,
   (a) The context for a particular document d given Θ_X is defined by the probability distribution $p_d(t|\Theta_X)$.
   (b) Since the context associated with a document may be sparse, further smoothing is required. Therefore, a context background model for each document,

$p_d(t|\theta_X)$ is constructed by smoothing the $p_d(t|\Theta_X)$ with the collection background model $p(t|\theta_C)$. This ensures there are no zero probabilities.

(c) Finally, a context based document model, $p(t|\theta_d^X)$ is constructed using the document's context background model, $p_d(t|\theta_X)$.

4.3.1 Semantic Associations

The choice of semantic associations is first limited to what information is available, and second which is appropriate to the information need. Typically, the semantic associations that are available in the IR domain include, but are not limited to, the following:

- ontology, structure, classification hierarchy
- citations, links, references
- semantic clustering (clusters formed by some human interaction or user defined relationship)

The set of context parameters is defined by $\Theta_X = \{p(t|x), p(x|d)\}$ where x is a context from the set of contexts available under X such that $x \in X$, $p(t|x)$ is the probability of a term given a context x and $p(x|d)$ is the probability of a context x given the document d. Each document is associated with one or more contexts as defined by the conditional probability $p(x|d)$. The context for the document is obtained by the marginalizing out x as shown in Equation 4.1. This is sometimes referred to as an Aspect model[129, 60].

$$p_d(t|\Theta_X) \underset{=}{\text{def}} \sum_{x \in X} p(t|x)p(x|d) \tag{4.1}$$

Using this decomposition allows for different types of associations to be encoded. A specific case is when the context is defined by a document to document relationship, for instance when citation information is used to define the context. Using this semantic association the number of contexts is equal to the number of the documents and there is a one to one correspondence between a document and a context, in which case the the context parameters can be defined as $\Theta_{X=links} = \{p(t|l)p(l|d)\}$ where $p(l|d)$ is the probability of document l given document d. A semantic association between the two

documents when $p(l|d) > 0$, otherwise there is no association in which case $p(l|d)$ will equal zero.

Later, in this thesis, we try several different types of associations on various test collections.

4.3.2 Context Background Model

Once the semantic associations have been defined and an estimate of the context for each document established, the context may still be relatively sparse, depending on the type of association. Hence, the ZPP will still may need to be addressed. Inevitably, we must now rely on the collection background model to resolve this problem.

A document's context background model $p_d(t|\theta_X)$ is defined as a two part mixture model of its context and the collection model.

$$p_d(t|\theta_X) = (1-\pi)p_d(t|\Theta_X) + \pi p(t|\theta_C) \tag{4.2}$$

If π is set to one, then no contextual information is used. However, when π is less than one, the user's understanding as quantified by the semantic association will be included in the document model.

4.3.3 Context Based Document Model

We propose two variants of context based document models, the standard approach using Jelinek Mercer Smoothing as shown in Equation 4.3 and the other using Bayes smoothing as shown in Equation 4.4. The context based document models models are a mixture between the maximum likelihood estimate of the document and its context background model.

$$p(t|\theta_d^X) = (1-\lambda)\frac{n(t,d)}{n(d)} + \lambda p_d(t|\theta_X) \tag{4.3}$$

The Bayes smoothing form is obtained by setting $\lambda = \frac{\beta}{n(d)+\beta}$ as shown in Equation 4.4. This can be considered as a *hierarchical* Bayes model, because of the levels of smoothing.

$$p(t|\theta_d^X) = \frac{n(t,d) + \beta p_d(t|\theta_X)}{n(d) + \beta} \tag{4.4}$$

Defining the context based document model as described above has the advantage that further contextual information may be incorporated into the model at a later stage. For instance the addition of a second stage of smoothing would allow model based feedback[161] or encoding query term importance [58].

4.3.4 Parameter Estimation

When building context based document models, there are several free parameters that need to be estimated. We first outline the measure that we shall use to measure the quality of the models used, then describe calculations to estimate the context background model, and then the context based document models.

Under the assumptions of Language Modeling, the data model and the retrieval model are considered the same and the data model influences the retrieval performance. Since we are obliged to develop the best possible representation of the documents (data model) we need to measure the quality of the representations generated. The statistical measure of goodness of fit is the log likelihood of the document on a held out sample of data, known as the *predictive likelihood*[29]. We can use the predictive likelihood as a measure of the quality of the document models. In speech recognition[123], the preferred measure is perplexity, which is directly related to the predictive likelihood. Perplexity is the exponential of the negative normalized predictive likelihood under the model. This gives an indication of the word error rate, which is used to evaluate such language models. The use of predictive likelihood in some form has been previously used in many studies on text retrieval[164, 3, 62, 61], where it has been generally believed that a better predictive likelihood will achieve better retrieval performance. This intuition matches the assumptions of Language Modeling already outlined. It should

be stressed that parameter estimation in this manner is conceptually different from that performed in the past. Previously, parameter's of Language Models (and other types of retrievals) have been tuned against the actual retrieval performance (typically on the query set for which the performance is reported). Clearly, tuning the model parameters under these circumstances requires *a priori* knowledge which in not available in a realistic environment. Hence, past results have tended to show the most optimistic retrieval performance given set of possible smoothing parameters.

In Section 3.5.4.1, we described the 'leave one out' log likelihood method which was used to measure the quality of the document models. We apply the same method for estimating the predictive likelihood for estimating π for the context background model, and estimating λ or β for the context based document model. Under this approach, we can select the smoothing parameters of the Language models which 'should' according to the second assumption also maximize the retrieval performance, without recourse to any *a priori* knowledge of the queries and relevance judgments.

4.3.4.1 Estimating π

When, each context $x \in X$ is defined by a set of documents, $x = \{d_1, \ldots, d_{|x|}\}$, the predictive likelihood for the context x given π is defined as follows.

$$\ell_{-1}(\pi, x) = \sum_{t \in x} n(t,x) \log \left((1-\pi) \frac{n(t,x)-1}{\sum'_t n(t',x)-1} + \pi p(t|\theta_C) \right) \tag{4.5}$$

where

$$n(t,x) = \sum_{d \in x} n(t,d)$$

The mean predictive likelihood for the set of contexts $\ell_{-1}^{avg}(\pi, X)$ is the average $\ell_{-1}(\pi, x)$ over all contexts, as shown in Equation 4.6, where $|X|$ is the number of contexts in X.

$$\ell_{-1}^{avg}(\pi, X) = \frac{1}{|X|} \sum_{x \in X} \ell_{-1}(\pi, x) \tag{4.6}$$

An estimate $\hat{\pi}$ is then assigned according to the π value which gives the highest mean predictive likelihood, such that:

$$\hat{\pi} = \underset{\pi}{\operatorname{argmax}}\ \ell_{-1}^{avg}(\pi, X) \tag{4.7}$$

The estimated $\hat{\pi}$ is then used to instantiate the context background models.

4.3.4.2 Estimating λ and β

Again, we use the same criterion of the predictive likelihood as defined by the leave one out log likelihood to estimate our context based document model parameters. The mean predictive likelihood (mPL) is used as an indication of the quality of the document models. The predictive likelihood for the context based document models using Jelinek Mercer smoothing is shown in Equation 4.8, while in Equation 4.9 is the equivalent for Bayes Smoothing.

$$\ell_{-1}(\lambda, d) = \sum_{t \in d} n(t,d) \log \left((1-\lambda) \frac{n(t,d)-1}{\sum_t' n(t',d)-1} + \lambda p_d(t|\theta_X) \right) \tag{4.8}$$

$$\ell_{-1}(\beta, d) = \sum_{t \in d} n(t,d) \log \left(\frac{n(t,d)-1+\beta p_d(t|\theta_X)}{\sum_t' n(t',d)-1+\beta} \right) \tag{4.9}$$

The mean predictive likelihood for the documents in the collection is the average of the predictive likelihood $\ell_{-1}^{avg}(\gamma, C)$, where γ represents the model parameter λ or β.

$$\ell_{-1}^{avg}(\gamma, C) = \frac{1}{|D|} \sum_{d \in C} \ell_{-1}(\gamma, d) \tag{4.10}$$

Again, we select the estimate $\hat{\gamma}$ which maximizes the mean predictive likelihood of the context based document model, as shown in Equation 4.11.

$$\hat{\gamma} = \underset{\gamma}{\operatorname{argmax}}\ \ell_{-1}^{avg}(\gamma, C) \tag{4.11}$$

By maximizing mPL our estimated parameters should be the best representation of the documents given the data from which they were estimated. With respect to the assumptions of Language Modeling, this is the objective function for obtaining the best retrieval performance. So, if a better representation of the underlying data is achieved using context based document models than better retrieval performance should also be possible. However, it remains to be seen whether this holds true, and this is examined in Chapter 5.

4.4 Unsupervised Learning

In this section we present two techniques which can automatically induce associations between documents. The first is the Naive Bayes mixture model, and the second is Probabilistic Latent Semantic Analysis. The former is a standard probabilistic clustering algorithm, while the latter induces more semantic based associations through co-occurrence data.

4.4.1 Naive Bayes

The Naive Bayes mixture model is not strictly context based, however, we include it for completeness and demonstrate how it can be applied within our framework to obtain 'cluster based' document models.

Under a generative Naive Bayes mixture model, each document d is generated from one (and only one) latent class g, given the set of classes G, such that $g \in G$. Given the latent class g, terms are drawn from that class to build the document of length $n(d)$. The terms are assumed to be independently and identically sampled from the class.

Under the Naive Bayes Model, the likelihood of a document is defined as follows:

$$\ell_d = \sum_g p(g)p(d|g) \tag{4.12}$$

where:

$$p(d|g) = \prod_{t \in d} p(t|g)^{n(t,d)} \tag{4.13}$$

To generate a document the sampling process is as follows:

1. we select the class g with probability $p(g)$
2. From g sample a term t
3. repeat step two until we have the desired sample

The probability of a document can be calculated by:

$$\begin{aligned} p(d) &= \sum_g p(d|g)p(g) \\ &= \sum_g p(g) \prod_{t \in d} p(t|g) \end{aligned}$$

The model is estimated using the standard procedure for maximum likelihood estimation of models with latent variables, the Expectation Maximization[30] algorithm. As mentioned earlier the procedure guarantees that a local maxima of the likelihood of all documents will be found. The EM algorithm alternates between the two steps: Expectation (E) Step and the Maximization (M) Step and is repeated until convergence (i.e the change in the likelihood of the model is zero or a very small improvement).

The E-step computes the posterior probabilities.

$$p(g|d) = \frac{p(g) \prod_{t \in d} p(t|g)}{\sum_{g'} p(g') \prod_{t' \in d} p(t'|g')} \tag{4.14}$$

The M-Step updates the parameters given the posterior probabilities.

$$\begin{aligned} p(t|g) &= \frac{1 + \sum_d n(t,d)p(g|d)}{|T| + \sum_{t'} \sum_{d'} n(t',d')p(g|d')} \\ p(g) &= \frac{1 + \sum_d p(g|d)}{|C| + |D|} \end{aligned} \tag{4.15}$$

Where $|T|, |D|$ and $|C|$ are the number of terms, documents and classes respectively. The count $n(t,d)$ is supplemented by Laplace smoothing which adds a count of one to avoid the assignment of any non-zero probabilities.

The probability of a term given a document can be determined from Equation 4.14. The estimation process may not converge completely to identify one class for a particular document. In such cases, the class with the highest probability $p(c|d)$ is selected as the class the document was drawn from and the appropriate class-specific uni-gram used.

4.4.1.1 Naive Model for *ad hoc* IR

Using the context based document model described in Equation 4.3, and setting the context parameters so that $\Theta_{X=G} = \{p(t|g), p(g|d)\}$, where association matrices are defined through the Naive Bayes estimation procedure, the final estimation details of the Naive Bayes context based document model is:

$$p(t|\theta_d^G) = (1-\lambda)p(t|d) + \lambda((1-\pi)p_d(t|\theta_G) + \pi p(t|\theta_C)) \tag{4.16}$$

If the number of classes is one, then the probability of a term given the document from the naive Bayes mixture will be equivalent to the probability of term given the collection. i.e. one class, the entire collection. Where further classes are added, and a class structure exists, the probability of an unseen term in a document will be more accurate. If the Cluster Hypothesis holds, then by expanding the document model using class information should increase the retrieval performance as well.

To implement cluster based retrieval given our context based model requires that we take the Jelinek Mercer variant proposed in Equation 4.3 and enforce certain conditions. First, we must set λ to zero, hence ranking is completely reliant on the cluster based model. We must also assume that a document may only be drawn from one cluster g, then the ranking of clusters is ascertained by the likelihood of drawing the query from the cluster of documents $p(q|g)$.

However, in practice ranking purely by clusters is rarely performed and often the rank-

ing based on the cluster and the document are combined in some fashion. Recently within the language modeling framework a similar model has been proposed by Liu and Croft[89]. They derive their model directly from the Cluster Hypothesis[67]. The resulting formulation of their cluster based document models, is essentially Equation 4.16. However, they induce clusters using k-means clustering as opposed to the Naive Bayes mixture model. The clusters formed are converted into probability distributions, $p(t|g)$ and $p(g|d)$. Using the cluster based document models they show that small gains in performance are possible on specific TREC collections 1, 2 and 3.

The Naive Bayes model (as the name implies) is rather simplistic and assumes that one document belongs to one class, hence multi-topic documents can not be adequately represented. Further, the model has been shown to suffer from over-fitting [12]. However, one benefit of the model is that it is relatively easy to implement and more efficient than more sophisticated estimation techniques such as Probabilistic Latent Semantic Analysis.

4.4.2 Probabilistic Latent Semantic Analysis

The Naive Bayes model makes the assumption that a document is generated from one and only one class. Under Probabilistic Latent Analysis (PLSA) it is assumed that the document is drawn from a number of classes. The model was first introduced by Hoffmann and Puzicha[63] for clustering, and developed further by Hofmann[62, 61, 60] for *ad hoc* Information Retrieval known as Probabilistic Latent Semantic Indexing (PLSI) or Probabilistic Latent Semantic Analysis (PLSA). The approach is a probabilistic alternative to the linear algebraic approach called Latent Semantic Indexing / Analysis (LSI/LSA) [28]. However, PLSA has two advantages: (1) PLSA defines a generative model of the document collection, and (2) instead of the Gaussian densities used in LSA, PLSA utilisers multinomial densities which are more suited to the term distribution (see [61] for a detailed discussion on the similarities of PLSA to LSA). In experiments by Hofmann significant increases in the performance were obtained over the VSM and LSA.

Formally, Probabilistic Latent Semantic Analysis is a latent variable model for co-occurrence data which associates an unobserved class variable z with each occurrence of a term t in a document d. Hence, the co-occurrence data is the term co-occurring with the document, through the latent variables. This enables a contextually smoothed version of the document to be generated through the following generative process:

1. A document is selected with probability $p(d)$
2. A latent class is selected with probability $p(z|d)$
3. A term t is generated with probability $p(t|z)$
4. go to step 2 and repeat sampling until done

The latent class variable z is marginalized and the result is the probability of a term and a document, i.e. the observed variables. The aspect model is mathematically expressed as follows:

$$\begin{aligned} p(t,d) &= p(t|d)p(d) \\ p(t|d) &= \sum_z p(t|z)p(z|d) \end{aligned} \tag{4.17}$$

Under the aspect model, the same term independence assumption is made. A further assumption is also asserted; conditional independence, based on the latent class, so that terms are generated independently of the document. By using an aspect model, a document can be a combination of a number of latent classes (unlike the Naive Bayes Mixture model). This generates a unique document representation for each document, and is considered to be a contextually smoothed representation of the document.

As with the Naive Bayes model, we can determine the probabilities $p(d)$,$p(z|d)$ and $p(t|z)$ by maximizing the likelihood function:

$$\ell_d = \sum_d \sum_{t \in d} n(t,d) log p(t,d) \tag{4.18}$$

Using the EM algorithm[30] the parameters for the model can be estimated to obtain a

local maximum for ℓ_d. The E-step computes the posterior probabilities.

$$p(z|d,t) = \frac{p(t|z)p(z|d)}{\sum_{z'} p(t|z')p(z'|d)}$$

The M-Step updates the parameters given the posterior probabilities.

$$p(t|z) = \frac{\sum_d n(t,d)p(z|d,t)}{\sum_{t'}\sum_{d'} n(t',d')p(z|d',t')}$$

$$p(z|d) = \frac{\sum_t n(t,d)p(z|d,t)}{p(d)}$$

The trivial estimate of $p(d) \propto n(d)$ is assumed. In order to estimate a better model of the underlying data tempering is suggested in [62]. Tempered EM is implemented as follows:

1. Hold out a portion of the documents contents (for instance 10% of the terms that occur in the document).
2. Set $\gamma = 1$ and perform EM until the performance on the held out data deteriorates.
3. Decrease γ by a small factor, such that $\gamma_{new} = \nu \times \gamma_{old}$, where $\nu < 1$.
4. Repeat TEM iterations at this new value of γ_{new} as long as the ℓ_d continues to improve.
5. Stop when decreasing the value of γ does not yield any further improvements to ℓ_d, otherwise goto to step 2.
6. Perform some final iterations using both training and held out data

However, the PLSA model has been criticized for not being a truly generative model and consequently a theoretically underpinned approach was proposed by Blei *et al.*[12]. They proposed Latent Dirichlet Allocation (LDA) as a fully generative alternative. Under such a model, the parameters are estimated using a *variational* approximation. However, it has been shown that PLSA is in fact the *maximum a posteriori* estimate of the LDA model [41]. The LDA model has not yet been used for *ad hoc* Information

Retrieval, though experiments on clustering tasks have shown it to outperform both the Naive Bayes and PLSA models. The PLSA model, however, has been applied to the text retrieval task but incorporated within a VSM.

The advantage of using PLSA is that each document is composed of a combination of latent classes in various proportions. This has a natural interpretation in that a document may be composed of several different latent variables. Sometimes the latent variables will be referred to as topics, but this is not strictly the case. We need to infer some kind of topicality from the distribution $p(t|z)$ such that it means something semantically before referring to it as a topic[12]. From the analysis performed this appears to be possible[12, 61, 60].

4.4.2.1 PLSI

Retrieval using PLSA is called Probabilistic Latent Semantic Indexing (PLSI)[60]. Hofmann presents two algorithms for ranking; one that used the conditional probability of a term given a document $p(t|d)$ and the other used the posterior probability of a latent class given a document $p(z|d)$. The former represents matching in the term space whilst the later attempts to match in the latent space (see [28, 60] for further details about matching in the latent space).

First, we describe the term space matching using PLSA and the VSM and then we compare this approach to our fully probabilsitic framework. The model proposed by Hofmann to match in the term space is PLSI-U [60]. PLSI-U uses a contextually smoothed representation of the document formed by linear combination of the empirical probability estimates of a term in a document with the PLSA estimates of a term in a document. This is then weighted by the Inverse Document Frequency to produce the document representation shown in Equation 4.19, where $0 \leq \lambda \leq 1$ and $p_{plsa}(t|d)$ is defined by Equation 4.17.

$$\hat{r}(t|d) = \left(\lambda p(t|d) + (1-\lambda) \left\{ \sum_z p(t|z)p(z|d) \right\} \right) .idf(t) \tag{4.19}$$

$$s(d,q) = \frac{\sum_t \hat{r}(t|d)r(t|q)}{\sqrt{\sum_t \hat{r}(t|d)^2}\sqrt{\sum_t r(t|q)^2}} \tag{4.20}$$

Equation 4.20 is derived from the Vector Space Model[127] where the angle between the document vector and query vector is computed to determine the similarity between them (Section 2.4.2). The combination is a fairly intuitive way of encoding the context within the document. It is unclear how to determine the combination parameter in such a way that would obtain the best performance, hence it needs to be empirically set. That is, the model parameters needed to be manipulated until the best retrieval performance was found. The inverse document frequency is also used. Essentially, the process is ad-hoc in nature, pieced together to give the best results. At this time Language Modeling for *ad hoc* retrieval had just been proposed accounting for such piecemeal combination strategy. On the other hand, the context based document modeling approach seamlessly integrates PLSA within a principled framework. The context parameters $\Theta_{X=Z} = \{p(t|z), p(z|d)\}$, where $p(t|z)$ and $p(z|d)$ are estimated using PLSA. Under this approach, indexing can be viewed as a principled alternative to the indexing performed in [60, 62], which is consistent with the Langauge Modeling paradigm.

Another related approach which uses an aspect model was proposed in [76]. To score documents they use a linear combination of the $p(q|\theta_d)$ based on the standard language model approach and $p(q|\theta_d^Z)$ based on an aspect model. This is conceptually different from our model because the we attempt to develop a more accurate representation of the underlying data, whilst this method relies upon a combination of evidence to manipulate the ranking of documents. Further, their method is similarity based, documents are associated with other documents using the KL divergence, where the closest m documents (those with the smallest KL divergence) are assumed to be related, i.e. $p(d'|d) = 1/m$ for m closest documents and $p(d'|d) = 0$ for the rest in the simplest case.

4.5 Summary

We have presented our framework for developing context based document models. Building such document models, should result in a more accurate representation of the underlying data, therefore abiding by the second assumption of language modeling. The context based document models that we have proposed are consistent with the probability theory that they are derived. This allows the incorporation of further forms of context (such as document prior, second stage of smoothing, etc) without compromising the integrity of the theory. In essence, this should result in an extensible approach for incorporating context that should, under the assumptions of Language Modeling deliver superior IR performance. Of course, this still relies on how well we can estimate the context based document models and to the extent to which the assumptions hold. In Chapter 6, we implement some context based models, with the following questions in mind:

- Can a better representation be obtained under context based document models?
- Will this translate in significantly better retrieval performance?
- Alternatively, does Assumption Two hold?

Chapter 5

Assumptions of Language Modeling

In this chapter, we undertake an analysis of the underlying assumptions of the Language Modeling approach for *ad hoc* text Retrieval. The assumptions are recast as hypotheses so that we can evaluate their validity through empirically based experiments. The experiments conducted within this chapter seek to deepen our understanding of the Language Modeling approach and provide a novel insight into the behavior, performance and utility of the approach.

5.1 Introduction

As we have previously mentioned in Chapter 3, the Assumptions of the Language Modeling approach can be stated as follows:

A1 **Correlation** The probability of a query given a document is *correlated* with the probability of a document being relevant[112, 57]. Or stated more firmly, that the probability of a query given a document is *proportional* to the probability of the document being relevant[78].

A2 **Unification** Relevance is subsumed by the document modeling process as shown in Equation 3.8, which is approximated with the $p(q|\theta_d)$. By ranking according to $p(q|\theta_d)$, the data model and retrieval function are one and the same. i.e

Unification[113].

A3 **Discrimination** The query terms that a user poses to the system are good at discriminating relevant documents from non relevant documents[113].

These assumptions have not been fully tested, nor empirically verified. Indeed, the considerable empirical success (in terms of IR performance) of the Language Modeling approach has to some extent been justification enough (resulting in widespread adoption). However, it is imperative that assumptions be tested for a number of reasons:

- to validate the approach,
- to develop a deeper understanding of the model,
- to revise/extend the assumptions/model accordingly
- and to determine its current limitations.

Each assumption has different effects on the application, usage and invariably its success of the Language Modeling approach; some examples of which are discussed below.

Assumption One is theoretically oriented with the implication that if it is not upheld then the ranking can not be guaranteed as optimal under the PRP. This is because the LM approach ignores relevance, which leads to further problems. When relevance feedback becomes available, then without relevance explicitly define it is unclear how to re-estimate the parameters. This implies that multiple relevant documents are to be retrieved. However, under the ideal document analogy, it may not be sensible to consider the retrieval of multiple relevant documents under the LM approach[139]. On the other hand, it is claimed that the query likelihood is actually proportional to the Odds Ratio[78]. It remains to be seen whether this, or the correlation do hold in practice.

The second assumption implies that a better representation of the document model should achieve better retrieval performance[112]. This is because the retrieval performance is inextricably linked to the document language models $p(t|\theta_d)$ as determining

the likelihood of the query terms being produced from the document model is precisely the mechanism by which documents are ranked. Ponte and Croft [113] claimed that this *unified the data model with the retrieval model* [1] because separate sets of inferences for indexing and for retrieval are not required. Hence the estimation of the document language model is not only vital to overcome the Zero Probability Problem but is paramount in obtaining better IR performance. Under this interpretation, our objective is clear. We should generate a better representation of the data (data model) to achieve better retrieval performance (retrieval model). We can use this principle then to guide in the unsupervised estimation of the model parameters by optimizing the data model, which should result in the optimal retrieval performance.

The third assumption posits that a user will submit a query consisting of keywords as opposed to a description of their information need. Therefore, issuing an entire TREC TOPIC, which includes a narrative and description of the information need, as a query may be construed as a misuse or even abuse of the model given the underlying assumptions. This is because the terms used to describe what is wanted may not neccessarily be the same as the terms in the document (vocabulary effect). Also, when users issue queries they need to imagine an ideal document and select terms from this document that would be likely to occur, requiring that the user must have some knowledge of the distribution of terms within documents and an ability to distinguish relevant documents from non relevant documents.

The validity of each of the three assumptions remains largely unexplored. In this chapter, we provide empirically based hypotheses to determine whether the assumptions hold or not, and to what extent. The remainder of this chapter is as follows: Section 5.2 restates the assumptions as hypotheses and Section 5.3 details the experimental methodology employed to test these hypotheses. In Section 5.4 we report the results of the experiments conducted along with a discussion of these findings.

[1] Our emphasis not the original authors.

5.2 Hypotheses

To test the assumptions of Language Modeling for text retrieval we have restated the assumptions as the following set of corresponding hypotheses:

HA1 The probability of a query being generated from a document $p(q|\theta_d)$ is correlated with the odds of a document being relevant $O(d|R,q)$.

HA2 The document language models which gives the 'best' representation of the underlying data will achieve the 'best' IR performance (and thus unification).

HA3 The query likelihood of relevant documents is greater than the query likelihood of non-relevant documents.

The transformation of each hypothesis to its assumptions required some interpretation which we explain. In (HA1) instead of comparing the query likelihood[2] directly against the relevance of a document $p(d|R,q)$ we have substituted the Odds Ratio instead. The Odds Ratio provides a measure of the relevance of a document which is comparable between documents as the probability of document given relevance is normalized by the probability of a document given non-relevance. Otherwise document's of different lengths would not be comparable, if we used $p(d|R,q)$ alone. Stating the assumption like this is exactly the decomposition offered by Lafferty and Zhai[78], (see Equation 3.9), who claim that the two measures are actually proportional, not just correlated.

In HA2, we restate the assumption in terms of its implication i.e. that the best data model will achieve the best IR performance and hence allows us to empirically determine whether this assumption holds. However, we need to quantify what we mean by 'best' with respect to the representation of data (data model) and 'best' with respect to the retrieval performance (retrieval model).

- The standard measures for IR were detailed in Chapter 2. We shall use the measures of mean Average Precision and Precision at 30 documents to quantify

[2]Note that the query likelihood measure does not need normalizing because the length of the query is fixed, and so the query likelihoods are comparable between documents.

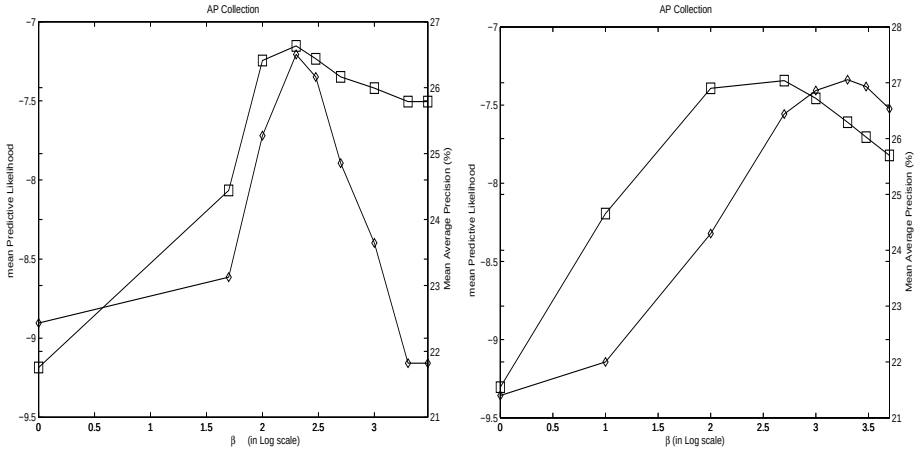

Figure 5.1: Left: An example of when the BRM and the BDM are unified, i.e both obtaining optimal performance given the parameter value. Right: An example of when the BRM and the BDM are not unified, and over fitting the data model results in obtaining the optimal retrieval performance. The performance of the data model (mPL) is denoted by the squares, and performance of the retrieval model (mAP) is denoted by the diamonds.

the retrieval performance. The 'best' retrieval model (BRM) is the one which obtains the highest precision values given the model parameters.

- For the data model, the statistical measure for the goodness of fit of the data model is measured by the log likelihood of the data model on a held out sample of data. We shall refer to this as the *predictive likelihood* of the document model. The mean predictive likelihood (mPL) is computed taking the mean of the predictive likelihood over all document models. Therefore, the 'best' data model (BDM) is the one which achieves the maximum mean predictive likelihood.

The hypothesis is tested by comparing the performance of the BDM against the performance of BRM, if they are equivalent (i.e. no significant difference) then we shall claim that the models are unified. See Figure 5.1 for a graphical representation of unification between the two models occurs and when it does not.

In HA3 we test to determine whether their is sufficient discrimination, or not, between relevant and non relevant documents given the respective query likelihood scores. To

quantify what is sufficient discrimination, we shall employ an appropriate statistical test, which allows us to quantify the discrimination at varying levels of significance

5.3 Experiments

This section details the experimental methodology employed for testing the stated hypotheses. We first outline both the data collections and document language models that we use through the course of the analysis. Then we describe the methodology undertaken to test each of the hypotheses.

5.3.1 Test Collections

Three distinct data collections were selected consisting of three Abstract collections, two full text collections and a Web collection. Each type has different characteristics providing a guide of the robustness of the model assumptions to differing conditions. The specific data collections used were: MedLine Abstracts (MED), Computer Abstracts from the ACM (CACM) , and CISI Abstracts(CISI); The full text document collections were taken from the TIPSTER/TREC Disks, Wall Street Journal Collection 1986-1992 (WSJ) and Applied Press Collection 1988-1989 (AP); and the web collection used was Web Track 2 Gigabytes collection (WT2g).

The standard query set associated with each abstract collection was used. On the full text collection, the TREC TOPICS 101-200 were used, where the title of the topic was treated as the query and on the web collection we used the TREC TOPICS 401-450, where the title of the topic was also used as the query.

The data preparation was standard and applied to both collections and queries; terms were stemmed using the Porter Stemming Algorithm[114], and standard stop words were removed. See Table 5.1 for an overview of the collection statistics, $\hat{n}(d)$ is the average document length, $\sum_d n(d)$ is the total number of term occurrences in the collection, $|Q|$ is the number of queries and finally the the total number of relevant documents aggregated over all queries is given.

Collection	$\|D\|$	$\hat{n}(d)$	$\|T\|$	$\sum_d n(d)$	$\|Q\|$	Total Rels
MED	1033	83	9380	86021	30	696
CACM	3204	91	13817	294478	52	796
CISI	1460	230	8418	336040	76	3114
AP	164597	243	196931	40111694	100	9738
WSJ	173252	247	175106	42862057	100	8469
WT2G	247491	611	1243186	151356491	50	2279

Table 5.1: Data Collection Statistics

Model	Param	Values
LP	α	0.0001, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1,10,100,1000
JM	λ	0.01,0.1,0.2,...,0.9,0.99
BS	β	1,10,50,100,200,300,500,1000,2000,3000,5000

Table 5.2: Document Language models and the Parameter values

5.3.2 Document Language Models and Parameter Space

We selected three different document language modeling approaches, detailed in Chapter 3, they were: Laplace Smoothing (LP) see Equation 3.24 , Jelinek-Mercer smoothing (JM) see Equation 3.25 and Bayes Smoothing (BS) see Equation 3.32. Table 5.2 shows their model parameter and the range of values that the model parameter was assigned. We selected these three models as opposed to any other aforementioned document modeling approaches for a couple of reasons. The LP model is the simplest approach which avoids the ZPP. From a statistical point of view it also makes the naivest assumption of prior knowledge of the term distribution (assumes terms are from a uniform distribution). The JM and BS are the more predominantly used LM with demonstrated empirical success. This is because JM and BS rely upon the background collection probabilities which provides a better estimate of the term occurring in the document and this gives a better representation of the document. The distinguishing feature between JM and BS methods is that the latter has an implicit length normalization component (see Section 3.4.4).

The experiments conducted were performed on all data collections and document language models. For each combination, the Assumptions were tested according to the procedures described in the following subsections.

5.3.3 Testing Assumption One

To test the first assumption we require the two measurements $O(d|r,q)$ and $p(q|\theta_d)$ for each document d, given the query q. To avoid computation problems where the multiplication of small probabilities results in zero instead of an extremely low probability value, because it is too small to be represented under the standard numerical types. Hence, we use the log of both measures, which simplifies the calculations to summations and the log transformation is monotonic so that the rank order is preserved. For each query, the log query likelihood $\log p(q|\theta_d)$ was computed for each document. The top 1000 documents were then selected and the corresponding log Odds ratio computed[3].

To estimate the Odds ratio, we used the generative relevance Modeling approach (described in Chapter 3.) An empirical relevance model was constructed from the set of documents that were relevant to the query, such that $p(t|R) = \sum_{d \in R} \frac{n(t,d)}{n(d)}$. The empirical relevance model was smoothed with the background model $p(t|\theta_C)$, such that the relevance model $p(t|\theta_R) = \frac{1}{2}p(t|R) + \frac{1}{2}p(t|\theta_C)$[4]. The non relevance model $p(t|\theta_N)$ was set to $p(t|\theta_C)$. The log Odds Ratio was then computed for each document $\log O(d|r,q) = \log \frac{p(d|\theta_R)}{p(d|\theta_N)}$.

To measure whether a correlation existed between the two measures, the set of points ($\log p(q|\theta_d)$,$\log O(d|r,q)$) for a given query were analyzed using the Spearman's rank test (ρ). The Spearman's rank test is a non-parametric test which we have opted for because the data points were not distributed normally. Had they been normally dis-

[3]We also considered the converse - ranking according to the log Odds Ratio and then computing the log query likelihood of the top 1000 documents, however the results were sufficiently similar that we only considered this point of view.

[4]The suggested amount of smoothing between the empirical estimate and the background collection model was between 0.4 and 0.8 in [82]. We conducted several experiments were we changed the amount of smoothing, however the rankings were rather invariant to the parameter change in the specified range.

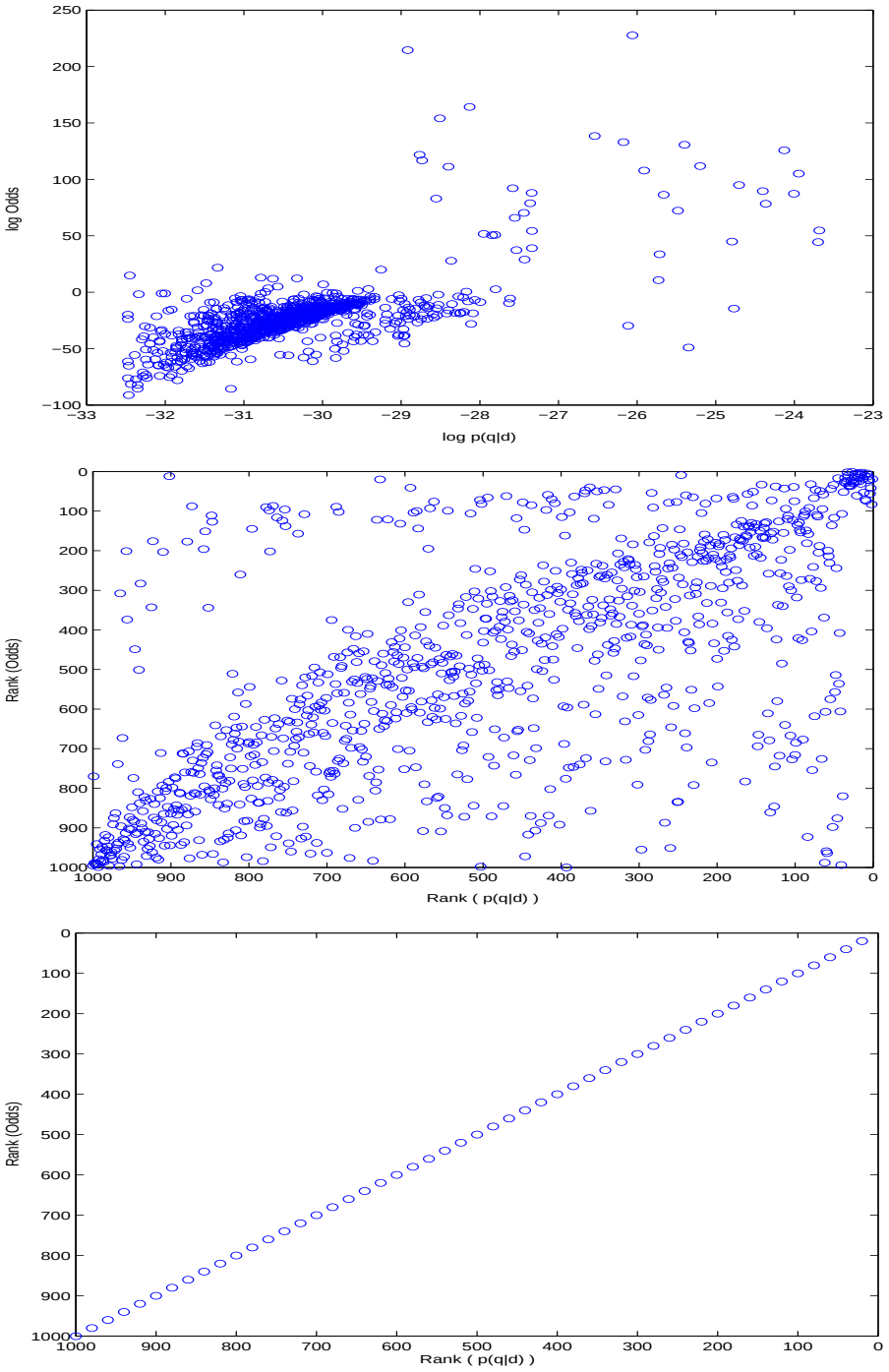

Figure 5.2: Top: Plot of the log query likelihood versus the log odds ratio. Middle: Plot of the log query likelihood versus log odds ratio using the ranks. Bottom: An example of when the correlation of the ranks between the query likelihood and odds ratio is one.

tributed we could have employed parametric tests such as Pearson's r co-efficient or considered linear regression. From Figure 5.2, we can see the actual values of $(\log p(q|\theta_d), \log O(d|r,q))$ plotted against each other in the top graph. The Spearman's rank test accounts for non normally distributed data, by resorting to the ranks of the scores, as shown in the middle graph. A positive correlation suggests that when one measure is high so is the other, whereas a negative correlation suggests that when one measure is high the other is low. In Figure 5.2, the bottom graph shows the former case of a perfect correlation. If the correlation co-efficient was statistically significant at 5% significance level for a particular query, then it was counted as being significantly correlated, the direction of the correlation was also noted.

The correlations were measured at three cut off points; at 30 documents, at 100 documents and at 1000 documents. The three points we examined provide different snapshots of the correlation, and also match up with standard evaluation metrics. i.e 1000 documents is the prescribed number of documents retrieved in the TREC evaluations; 30 documents represents the set of documents that are generally the only documents viewed or examined by the user for that query (Conveniently, it also provides a sufficient sample size for measuring the correlation).

It is worth pointing out that we are testing to determine whether there is a correlation between the two measures, with out considering whether those documents are relevant or not. This is considered by Assumption Three.

5.3.4 Testing Assumption Two

To test the second hypothesis, given the model parameters, we measured the corresponding data model and retrieval model. As previously mentioned the IR performance was measured by recording the mean Average Precision (mAP) and the precision at 30 documents (p@30docs). The data model was measured by recording the mean Predictive Likelihood. The mean Predictive Likelihood was computed using cross validation of the document model, to do so: One term from the document is held out, and then the document model is created given the model parameters. Then the term which was

held out is predicted by the model. This is repeated for every term in the document and the likelihood of the model is the product over each of the left out terms. This process is known as the 'leave one out likelihood'. This process is preformed for every document and the average taken to obtain the mean predictive likelihood. Maximizing the mean predictive likelihood will result in the best representation of the underlying data given the document language model. This is akin to the process described in Chapter 3 Section 3.5.4.

The predictive likelihood for each document language model, LP, JM and BS, can be computed as shown in the Equations 5.1, 5.2 and 5.3, respectively.

$$\ell_{-1}(\alpha, d) = \sum_{t \in d} n(t,d) \log \left(\frac{n(t,d) - 1 + \alpha}{n(d) - 1 + |T|\alpha} \right) \tag{5.1}$$

$$\ell_{-1}(\lambda, d) = \sum_{t \in d} n(t,d) \log \left((1-\lambda) \frac{n(t,d) - 1}{n(d) - 1} + \lambda p(t|\theta_C) \right) \tag{5.2}$$

$$\ell_{-1}(\beta, d) = \sum_{t \in d} n(t,d) \log \left(\frac{n(t,d) - 1 + \beta p(t|\theta_C)}{n(d) - 1 + \beta} \right) \tag{5.3}$$

The mean predictive likelihood is calculated as shown in Equation 4.10, where γ represents the model parameter either α, β or λ. The parameter value which gave the highest mean predictive likelihood according to Equation 4.10 was noted and will be referred to as the Best Data Model (BDM) given the fixed set of parameter values (see Table 5.2).

To obtain a better estimate of the data model given the set of parameters used, we took the average of the set of parameter values that maximized each document's predictive likelihood. See Figure 5.7, for a graphical representation of the distribution over the parameter values that obtained the maximum predictive likelihood. The distribution was then estimated by the mean of this distribution. Note, that the LP and BS document models used a parameter space that was not linear and simply taking the mean would have heavily biased the statistic towards the larger values. Instead, the parameters were first transformed in log space, took the mean, and reconstructed the actual parameter

value by raising the mean to the exponent. The models with the estimated parameter values will be referred to as the estimated Best Data Model $B\hat{D}M$. Whilst this does not ensure that the optimal estimate of the parameter is selected, it provides reasonable estimate.

We then selected the Best Retrieval Model (BRM), according to the model parameter that gave the highest mean Average Precision or Precision at 30 documents.

The hypothesis that the IR performance of the Best Data Model (BDM)/($B\hat{D}M$) would be equal to the IR performance of the Best Retrieval Model(BRM) was then tested on each collection, using the Sign Test at 5% significance level.

5.3.5 Testing Assumption Three

Under Assumption Three, we aim to determine whether there is a sufficient amount of discrimination between relevant and non relevant documents given the query likelihood scores. This presents an interesting challenge as the standard statistical tests are generally inappropriate. The use of a parametric test such as the point bi-serial correlation test, which is equivalent to the χ-Squared test, but for a continuous variable, in this case $(q|\theta_d)$ versus a discrete binary variable (which denotes R or N) would be inappropriate because the distribution of query likelihoods are not normally distributed for (non) relevant documents. This is similar to the approach taken by Swets[143] who assumed normal distributions for the scores of relevant and non relevant documents. Others have modelled the score of non-relevant documents using other distributions such as an exponential distribution[2, 165] or an empirical distribution[90], however, the scores of relevant documents has generally been assumed to be normally distributed[143, 90] or taking the form of a gamma distribution[2]. Such an assumption about the relevant documents is difficult to accept, because generally the number of points from which to estimate this distribution is limited (and usually not enough to form a reliable estimate of the parameters of a normal distribution).

Inspired by this idea of modeling the distribution of scores, we developed a non-parametric approach. Instead of comparing whether there was a difference between

the distributions of scores from relevant and non relevant documents, we first modelled the scores from non-relevant documents through an empirical distribution as in [90] (because we have numerous examples from which to estimate the form, we expect that we will obtain a reasonably good model of the distribution of non-relevant documents). Once we have approximated the distribution of non relevant documents, then we tested each relevant document to determine if the score of that document was significantly higher than non-relevant documents at various levels of significance using a one tailed test.

To quantify the meaning of 'sufficiently discriminative' we need to set two criteria:

1. the significance level (or rejection threshold), and
2. the proportion of relevant documents desired.

The first defines the level at which we reject the null hypothesis that the relevant document came from the non relevant distribution. Alternatively, this can be thought of as the proportion of non relevant documents willing to be seen during the course of a search session. The second defines the proportion of relevant documents that need to be returned given the significance level from (1) which would be deemed sufficient discrimination between relevant and non-relevant. For example, if one user is willing to see only 1% of the non relevant documents, but wants at least half of all the relevant documents, then this would quantify what they would term sufficient discrimination. However, others may disagree and be willing to see more non-relevant documents or are happy with a smaller proportion of relevant documents.

For these experiments we set the second criterion to half (as that seems to be a reasonable expectation), whilst the first criterion was varied, ranging from 0.1% up to 25% significance. The number of queries which exceeded the second criterion at each significance level was counted, and the overall proportion of relevant documents retrieved at each significance level was also recorded.

5.4 Results and Discussion

This section provides a summary of the pertinent results from our experimental analysis along with relevant discussion. We report the number of correlations in A1 and number of discriminations in A3, as proportions. This was to enable some comparison between data sets since they have differing numbers of queries.

5.4.1 Assumption One

Table 5.3 shows the results of the proportion of correlations (positive and negative) that were significant for each of the data collections at recall of thirty documents, one hundred documents and 1000 documents. (i.e the total number of queries that had a significant (positive/negative) correlation at 30/100/1000 documents divided by the total number of queries.)

Figure 5.3, the top graph shows each pair $(\log p(q|\theta_d), \log O(d|R,q))$ plotted for one of the queries on the AP collection using BS. The bottom graph shows the distribution of the query likelihoods for the non relevant documents. In the graphs the dashed lines indicate the different significance thresholds.

The results indicate that the correlation between the Odds of document being relevant and the query likelihood is infrequent/low when considering the top thirty document, but occurs often when considering the top 1000 documents.

At 1000 documents, the correlation between the Odds and the query likelihood was typically significant, and statistically so for a high proportion of queries. However, this may be an artifact of phenomena and the large sample size. Taking a correlation at 1000 documents swaps the comparison; the main bulk of documents are non-relevant which attract a low Odds Ratio and low query likelihood (hence, both are assigned low ranks). The swapping can be seen by examining the empirical distribution of the query likelihoods of non-relevant documents as shown in Figure 5.3, where the majority of documents attract a very low score and rank. Hence, the correlation appears to exist, because of these non-relevant documents, contributing to the correlation. This

Collection	Model	Pos @30	Pos @100	Pos @1000	Neg @30	Neg @100	Neg @1000
MED	LP	**0.5**	**0.7**	**0.9**	0	0	0
	JM	**0.567**	**0.867**	0	0	0	0.033
	BS	**0.567**	**0.867**	**1**	0	0	0
CACM	LP	0.288	0	0	0	0.269	**0.576**
	JM	0.25	**0.538**	0.442	0	0	0
	BS	0	0	0	0.25	0.346	**0.692**
CISI	LP	0.118	0.342	**0.973**	0	0	0
	JM	0.145	0.368	**0.894**	0	0	0
	BS	0.105	0.368	**1**	0	0	0
AP	LP	0	**0.58**	**0.88**	0.25	0	0
	JM	0.16	**0.55**	**0.98**	0	0	0
	BS	0.25	**0.64**	**0.93**	0	0	0
WSJ	LP	0.36	**0.67**	**0.83**	0	0	0
	JM	0.07	0.45	**0.93**	0	0	0
	BS	0.31	**0.63**	**0.91**	0	0	0
WT2g	LP	0	0.42	**0.52**	0.22	0	0
	JM	0.22	0.46	**0.92**	0	0	0
	BS	0.32	**0.52**	**0.84**	0	0	0

Table 5.3: The proportion of positive and negative correlations between the Odds Ratio and the Query Likelihood at Recall of 30, 100 and 1000 documents, for each of the Document Models and Collections. The values in bold are when over half of the queries were significantly correlated, which generally is the case for 1000 documents.

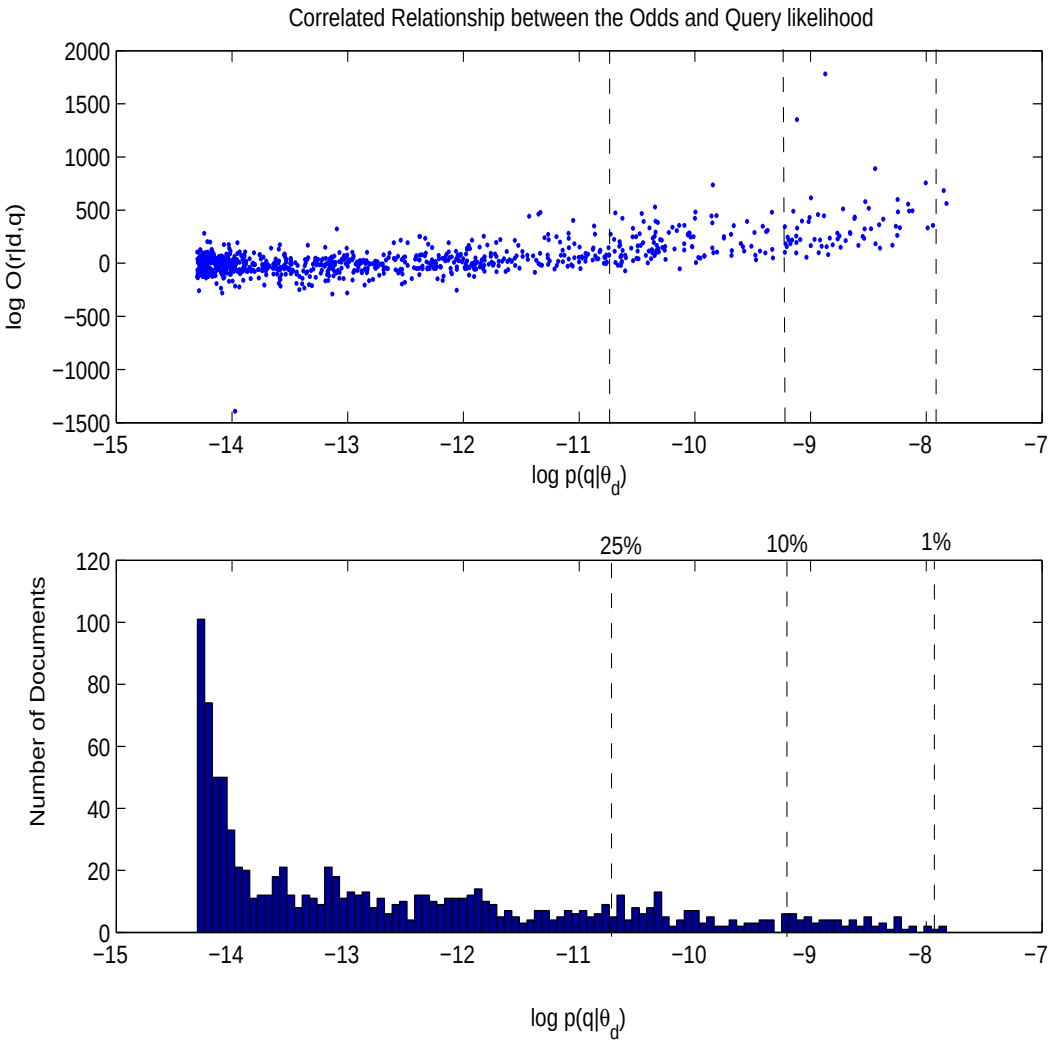

Figure 5.3: Positive Correlation on the AP Collection at 1000 documents using BS. Top: Odds Ratio vs Query Likelihood, Bottom: Distribution over the Query Likelihood. From the examples above, at 1% significance a relevant document would have need a log query likelihood greater than -7.9 to be rejected from the non-relevant distribution. At 10% a score greater than -9.2 will be enough to reject it, and so forth.

is not particularly useful, and indeed we can think of many measures that would have a similar correlation with the odds ratio, for instance the number of query terms a document has, or the size of a document.

Hence, we place more emphasis on the correlation at thirty documents instead. This is an appealing point to test A1 because it is typically the only part of the list that is actually inspected or examined by the user and it is sufficiently sized for statistical testing.

At thirty documents, the MED collection had the highest percentage of positive correlations where HA1 was successfully upheld approximately 50% of the time. However, on the other collections the proportion of queries where HA1 was upheld was approximately 20%. In some cases, negative correlations were witness on the CACM and WT2g collections. This suggests that the query likelihood measure may actually be giving a completely different ranking from the Odds Ratio.

At this early level of recall the evidence for the A1 hypothesis appears contradictory. However, when we examined the correlation at 100 documents, then the correlations were generally all positive for the majority of queries. The proportion of queries for which the two measures correlated is about 50% at 100 documents. Assuming that this is a reasonable point to capture whether Assumption One holds, then is difficult to ascertain whether the correlation of Assumption One actually holds or not. The intuition of the assumed correlation is certainly appealing, because of the model's empirical success (i.e the model returns lots of relevant documents at the top of the ranked list - hence there must be a correlation). However, saying that there is a correlation with relevance may just be a convenient way of ignoring any focus on relevance, whilst still obtaining operational effectiveness. For instance, the documents may not be ordered in decreasing likelihood of relevance, but by similarity (or some other relationship) to the query.

The times that A1 failed to hold, may be because A3 is violated. For instance, if the user submits poor query terms that do not appear in, or discriminate relevant documents from non relevant documents, then we would not expect that the measures would be correlated. We explore this further when analyzing Assumption Three in Subsection

5.4.3.

5.4.2 Assumption Two

Figures 5.4, 5.5 and 5.6 show the changes in the mean predictive likelihood and retrieval performance with respect to the parameter values. In each of these graphs, the square indicate the mean predictive likelihood of the document models with the corresponding value on the left axis, while the diamonds indicate the precision measures (either mAP or p@30docs) with the values shown on the right axis. The horizontal axis denotes the value of the smoothing parameter (shown in log scale for LP and BS document models).

The most striking result from these Figures is that each of the smoothing methods tend to produce a distinctive curve for their mean predictive likelihood. The LP method peaks at low values of α then steadily decreases (as α increases). The JM method tends to produce a dome shape that peaks around the $\lambda = 0.5$ (the standard value used in many papers), and the BS method increases dramatically before reaching a turning point and steadily subsides. On the other hand, in the somewhat erratic retrieval performance for each method, there were signs, of a systematic relationship with the mean predictive likelihood. This recurring pattern of behavior meant that the best retrieval performance was often obtained when more smoothing was applied, despite the fact that this degraded the mean predictive likelihood (this degradation in the data model is often referred to as over fitting. In Chapter 6 we provide an example of extreme over fitting in Section 6.1.3.1).

These findings appear to be quite consistent regardless of the retrieval performance measure, mAP and p@30docs. Since the graphs tended to be very similar between collections, we only show those for the different types of data collections instead of all of them. However, these are representative of the results that we obtained throughout the course of this study.

Tables 5.4, 5.5 and 5.6 show the statistics for the *BRM* and the *BDM*s and report any significant differences between the IR performance (mAP and p@30docs). In

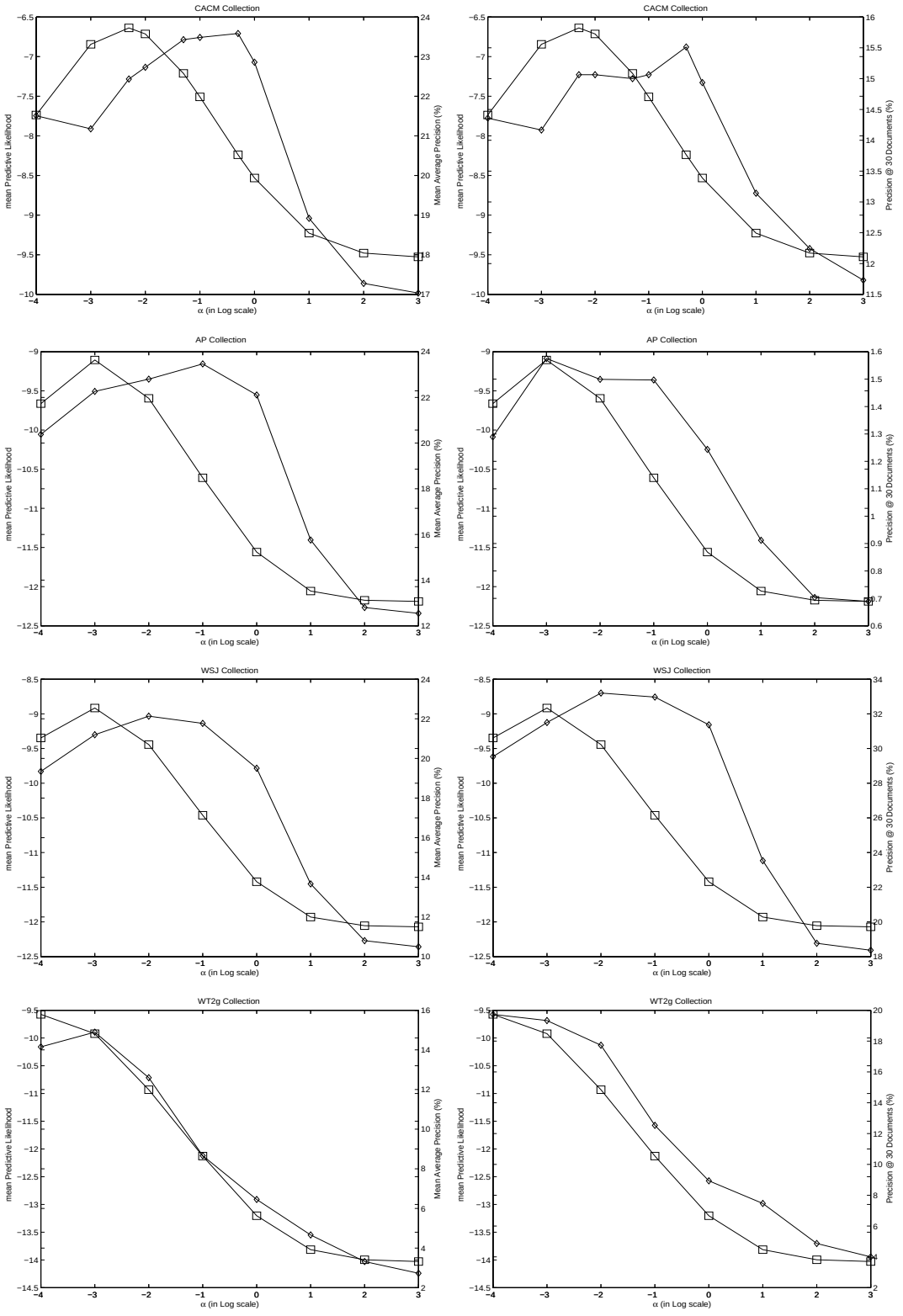

Figure 5.4: The change in measures for Laplace Smoothed Document Models. Top to Bottom: CACM, AP, WSJ and WT2g. Right: mPL vs mAP Left: mPL vs p@30docs. In most graphs there is a mismatch between the maximum mPL and maximum mAP, though for the AP collection measured with p@30docs exhibits unification of the BRM and BDM.

Collection	Point	α	mPL	mAP (p@30 Docs)
MED	BRM	0.05	-7.5908	48.69c (39.00)
	BDM	0.01 (0.05)	-7.2956 (-7.5908)	48.63 (39.00)
	$B\hat{D}M$	0.0087	-7.2613	48.44 (39.66ab)
CACM	BRM	0.5	-8.2385	23.58bc (15.51bc)
	BDM	0.001	-6.8446	21.18 (14.17)
	$B\hat{D}M$	0.0051	-6.6281	22.44b (15.06b)
CISI	BRM	0.5 (1)	-7.0211(-7.3527)	13.80bc (18.25bc)
	BDM	0.05	-6.0988	12.57c (16.62c)
	$B\hat{D}M$	0.0148	5.9521	11.04 (14.96)
AP	BRM	0.1	-10.6103	23.47bc (34.07bc)
	BDM	0.001	-9.1046	22.26c (32.90c)
	$B\hat{D}M$	0.0011	-9.0986	22.30 (32.93)
WSJ	BRM	0.01	-9.4434	22.13bc (33.20bc)
	BDM	0.001	-8.916	21.20c (31.5c)
	$B\hat{D}M$	0.00099	-8.8905	15.28 (25.13)
WT2g	BRM	0.001(0.0001)	-9.9212 (-9.5734)	14.90 (19.73)
	BDM	0.0001	-9.5734	14.1539 (19.73)
	$B\hat{D}M$	0.00020662	-9.5431	14.62(20.13)

Table 5.4: The statistics for the best data models and best retrieval models for each collection when employing the LP document models. The BRM gives significantly better retrieval performance on all collections except WT2g, where the a, b and c denotes the setting was significantly different to the BRM, BDM and $B\hat{D}M$, respectively.

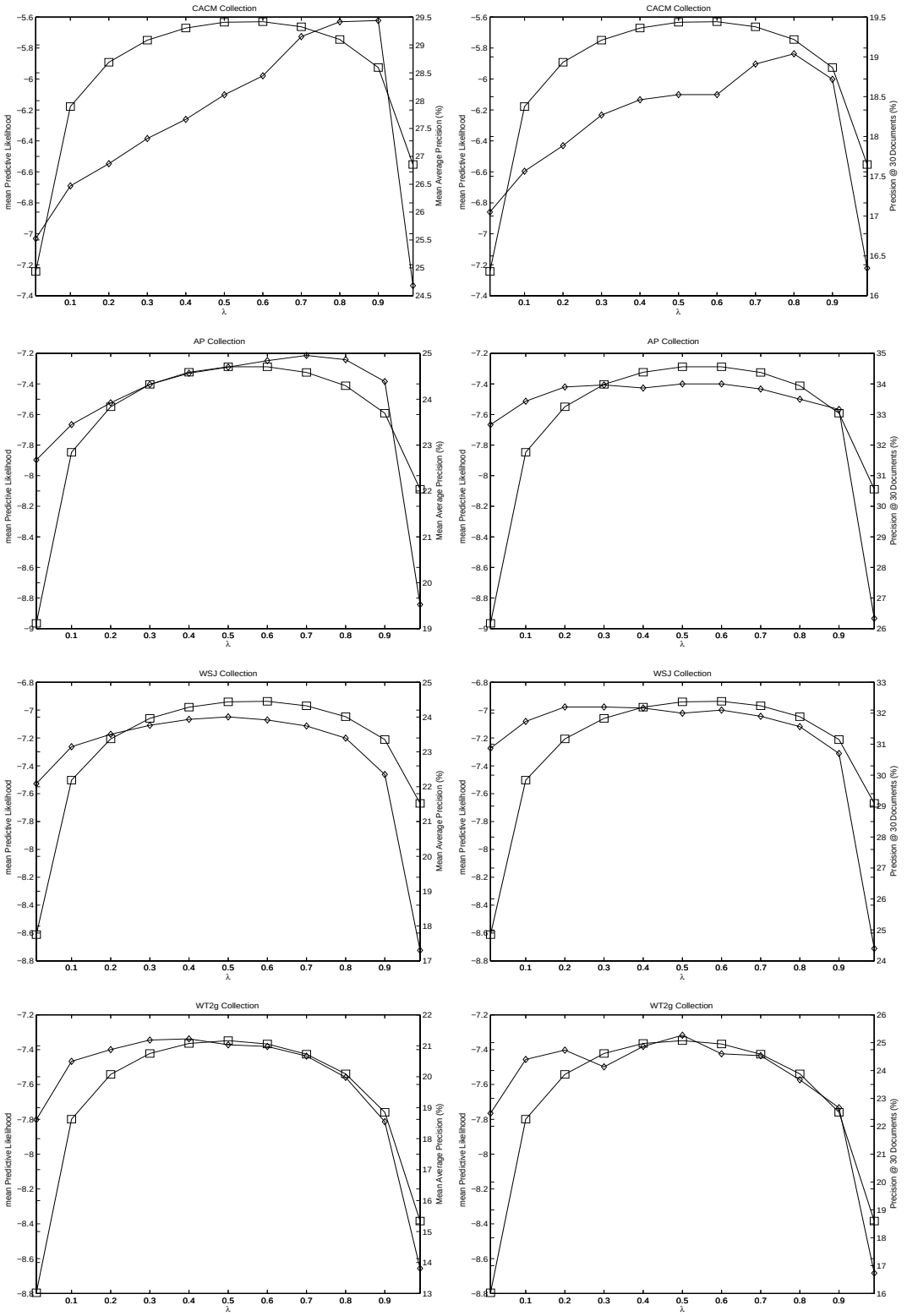

Figure 5.5: The change in measures for Jelinek Mercer Smoothed Document Models. Top to Bottom: CACM, AP, WSJ and WT2g. Right: mPL vs mAP Left: mPL vs p@30docs.

Collection	Point	λ	mPL	mAP (p@30 Docs)
MED	BRM	0.6 (0.8)	-6.7196 (-6.8354)	52.10 (42.44)
	BDM	0.6	-6.7196	52.14 (41.89)
	$B\hat{D}M$	0.5654	-6.6765	52.06 (41.78)
CACM	BRM	0.9 (0.8)	-5.926 (-5.7448)	29.44 (19.04)
	BDM	0.6	-5.6302	28.45 (18.53)
	$B\hat{D}M$	0.5529	-5.5973	28.32 (18.53)
CISI	BRM	0.1 (0.01)	-5.9927 (-6.93)	14.38 (17.85)
	BDM	0.5	-5.519	14.08 (17.76)
	$B\hat{D}M$	0.54199	-5.5094	14.06 (17.85)
AP	BRM	0.7 (0.6)	-7.3252 (-7.288)	24.9533 (34.00)
	BDM	0.5	-7.288	24.6993 (34.00)
	$B\hat{D}M$	0.54924	-7.2658	24.8302 (34.03)
WSJ	BRM	0.5 (0.3)	-6.9408 (-7.0595)	24.00 (32.20)
	BDM	0.6	-6.9372	23.92 (32.10)
	$B\hat{D}M$	0.556	-6.9065	24.02 (32.00)
WT2g	BRM	0.4 (0.5)	-7.3643(-7.3481)	21.22 (25.27)
	BDM	0.5	-7.3481	21.03 (25.27)
	$B\hat{D}M$	0.48907	-7.286	21.07 (25.27)

Table 5.5: The statistics for the best data models and best retrieval models for each collection when employing the JM document models. Notice that the estimated BDMs are not significantly different from the corresponding BRMs in terms of retrieval performance.

these Tables the values for p@30 docs are shown in brackets, with the corresponding parameter values if different from the those for the mean Average Precision. The letters after the mAP (and p@30docs) values indicate whether the value was significantly different from the other values (using the sign test at 5% significance), *a* denotes that it was better than *BRM*, *b* better than *BDM* and *c* better than $B\hat{D}M$.

The standard error of the predictive likelihood is not shown in the Tables, but for the smaller collections it varied from 0.01 to 0.02 and the larger TREC collections, from 0.001 to 0.002. This was regardless of the document language model, but proportion to the document collection size, as more samples (documents) will decrease the standard error. When we compared the mean predictive likelihood of each model, using a Wilcoxon Ranksum Test, we found that they were significantly different at 5% significance level, unless the model parameter was the same. Hence, the mean predictive likelihood of the BDMs were consistently and significantly different to mean predictive likelihood of the BRMs, for each collection and document model.

The results from our experiments, analyzing the relationship between the mean predictive Likelihood and the mean Average Precision (and p@30docs) indicate that for the LP and BS models that a mismatch between the the BDM and BRM existed. This mismatch translated into a significant difference in mean Average Precision and p@30docs between the BDM and BRM on most of the collections.

For JM the mismatch between the BRM and the BDM was not significant with the best parameter value $\lambda \approx 0.5$ obtaining the best performance. Figure 5.7 shows some examples of the empirical distribution of the λ_d parameter, where the distribution is clearly centred around the 0.5 value. This is a typical value suggested by previous research. However, the difference being we arrived at this conclusion from the assumptions of the Language Model, instead of requiring queries and corresponding relevance judgements.

So far only the JM model has performed as expected given the second assumption. The LP and BS models require over fitting of data model to obtain the best IR performance and hence a mismatch between the data and retrieval models. This suggests that the JM is a better document model to employ for *ad hoc* text retrieval.

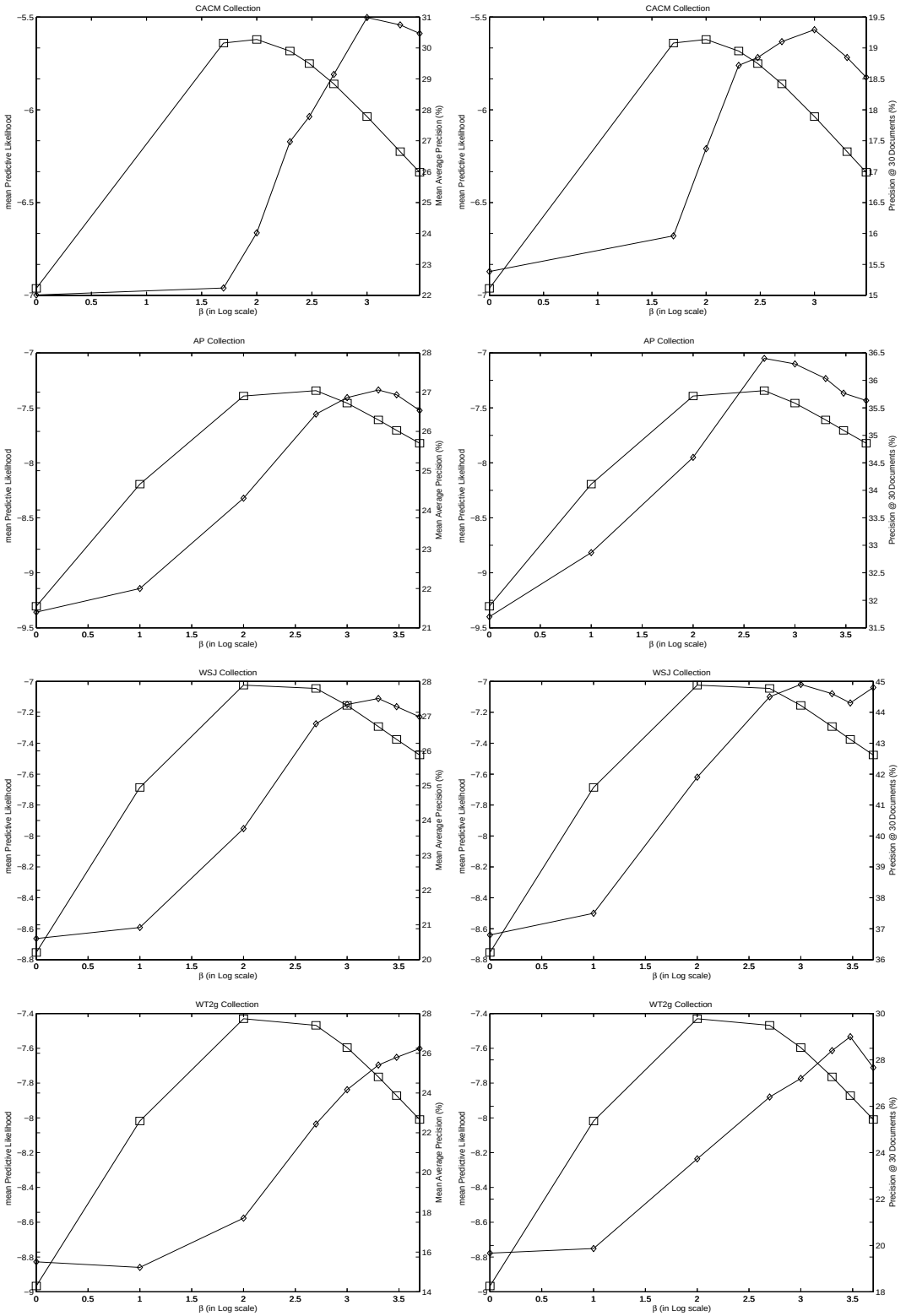

Figure 5.6: The change in measures for Bayes Smoothed document models. Top to Bottom: CACM, AP, WSJ and WT2g. Right: mPL vs mAP Left: mPL vs p@30docs. Notice the pronounced divergence between the best BRM and best BDM indicating a lack of unification.

Collection	Point	β	Log Likelihood	mAP(p@30 Docs)
MED	BRM	300 (200)	-6.8426 (-6.7766)	51.21 (41.89)
	BDM	100	-6.7235	49.81 (40.67)
	$B\hat{D}M$	103.38	-6.6844	49.95b (41.33b)
CACM	BRM	1000	-6.0363	30.98bc (19.30bc)
	BDM	100	-5.6208	24.02c (17.37c)
	$B\hat{D}M$	82.03	-5.6001	23.5164 (16.99)
CISI	BRM	3000 (2000)	-5.96 (-5.8609)	12.83bc (17.50bc)
	BDM	200	-5.5398	10.69 (15.00)
	$B\hat{D}M$	228.54	-5.5108	10.87b (15.16b)
AP	BRM	2000 (500)	-7.46 (-7.34)	27.06 (36.4c)
	BDM	500	-7.3436	26.44c (36.4c)
	$B\hat{D}M$	279.02	-7.29	25.85 (35.67)
WSJ	BRM	2000	-7.2923	27.51bc (37.73bc)
	BDM	100	-7.0244	23.77 (33.17)
	$B\hat{D}M$	371.6	-6.9245	26.37b (35.47b))
WT2g	BRM	5000 (3000)	-8.0093 (-7.8719)	26.24bc (29.00bc)
	BDM	100	-7.4301	17.71 (23.73)
	$B\hat{D}M$	219.55	-7.3111	20.70b(25.47b)

Table 5.6: The statistics for the best data models and best retrieval models for each collection when employing the BS document models. Notice the pronounced divergence between the BRM and BDM, and consequently lack of unification.

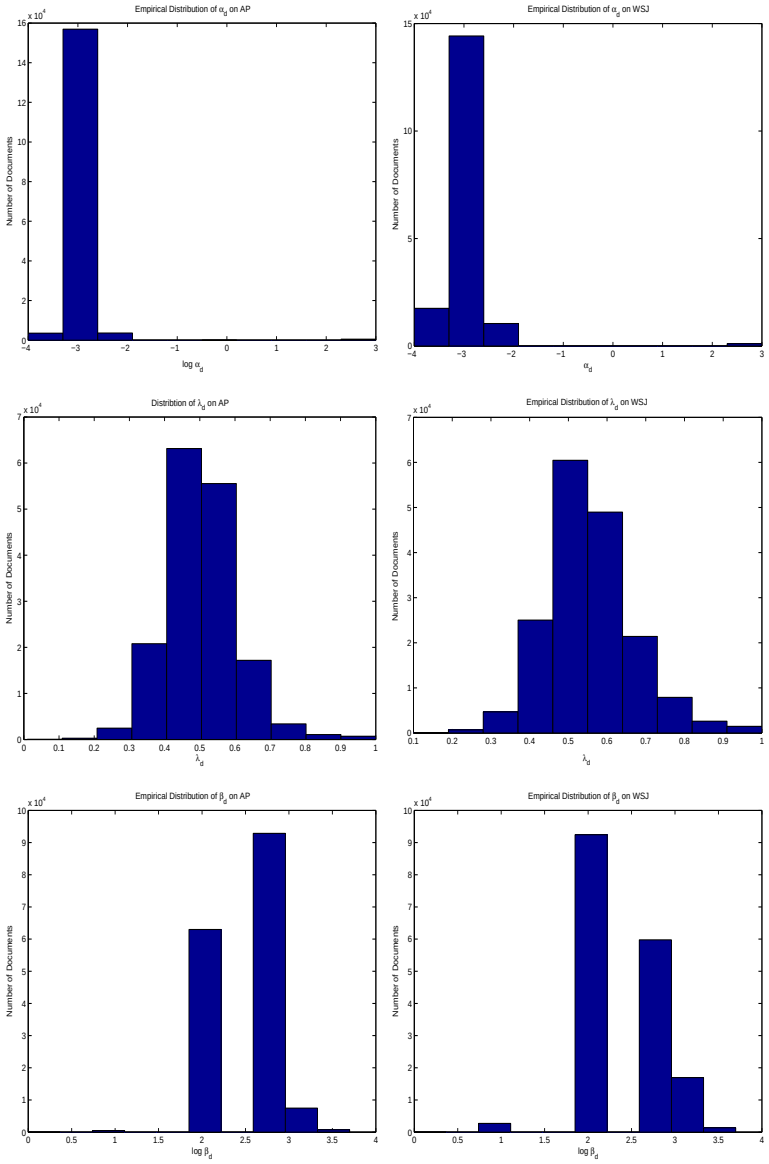

Figure 5.7: The distribution of model parameters values that maximized each document model's predictive likelihood. Left: AP Right: WSJ Top to Bottom: LP, JM and BS.

When we compared the BDMs of each document model against each other (See Table 5.7) the results show that the JM document models consistently outperform LP and BS in terms of model likelihood (and this was significantly different). But in terms of mAP the JM model was the best for the abstract collections and the web collection. The BS model, despite having a lower mean predictive likelihood than JM, outperformed the JM in terms of mAP. This is a very interesting result, and somewhat contrary to popular belief (i.e it is typically thought that Bayes Smoothing is a better estimator), and indeed the assumptions of the Language Modeling approach. Typically, it has been assumed that if a generative probabilistic model can increase the predictive likelihood of the data then the performance will improve. For instance, it is claimed that LDA increases the predictive likelihood over Naive Bayes and PLSA, but does that necessarily translate into better clustering or retrieval performance? Similar anomalies have appeared in speech recognition literature where the a higher model likelihood has not translated into lower word error rates[123]. This illustrates the problem with using a surrogate measure as an indicate of the actual performance.

Zhai and Lafferty[162] suggest that BS document model is the best document modeling technique. From a retrieval standpoint the BS model appears to be able to offer more in terms of IR performance as it contains an implicit document length normalization component. However, according to the model likelihood, this does not appear to be the case. Also, the best IR performance can only be obtained when the parameters are empirically set (i.e performing an exhaustive search over the parameter space, searching for the β value that gives the best mAP). From the graphs in Figure 5.6 we can see that over fitting the data model for the BS tends to achieve the best performance. These discrepancies could be a results of several factors, the need for document length normalization on the larger collections (with documents of varying sizes), variations in the query (length of query and importance of terms), or some other factor.

The two stage model was empirically motivated from discrepancies in performance when queries of different length were used[163]. To examine whether the mismatch was a result of not accounting for the query variations we performed some further experiments. First, we selected the collections where there was a significant difference in mAP between the BRM and $B\hat{D}M$ for the BS models. These were CACM, CISI,

Collection	Method	mPL	mAP	Recall
MED	JM	-6.6765*	52.06*	695/696
	BS	-6.6844	49.95	695/696
	LP	-7.2613	48.44	695/696
CACM	JM	-5.5973*	28.32*	720*/796
	BS	-5.6001	23.52	603/796
	LP	-6.6281	22.44	647/796
CISI	JM	-5.5094*	14.06*	2834*/3114
	BS	-5.5108	10.87	2536/3114
	LP	-5.9521	11.04	2635/3114
AP8889	JM	-7.2658*	24.83	6329/9738
	BS	-7.2872	25.85*	6319/9738
	LP	-9.0986	22.30	5887/9738
WSJ	JM	-6.9065*	24.02	5367/8469
	BS	-6.9245	26.37*	5420*/8469
	LP	-8.8905	15.28	4578/8469
WT2g	JM	-7.2860*	21.42*	1789*/2279
	BS	-7.3111	20.70	1709/2279
	LP	-9.5431	14.62	1361/2279

Table 5.7: The performance statistics for Assumption Two given the BDM of each document model, and for each data collection. The best result is denoted as significantly different to the others by an Asterisk after the value.

	Stage	mAP	p@30docs
CACM	BRM	27.78c	18.85c
	BRM+SS	30.72cd	18.78
	BDM	23.52	16.99
	BDM+SS	27.36c	18.78c
CISI	BRM	12.83cd	15.48
	BRM+SS	12.93cd	17.81
	BDM	10.87	15.18
	BDM+SS	12.13c	16.23ca
WSJ	BRM	27.51cd	37.03c
	BRM+SS	27.48cd	37.73cd
	BDM	26.37	35.50
	BDM+SS	26.35c	35.47
WT2g	BRM	26.24cd	29.00cd
	BRM+SS	26.34cd	28.07d
	BDM	20.70	25.47
	BDM+SS	21.66c	25.60

Table 5.8: Results of the Addition of the Second Stage to BS document models. Within each data collection *a*, *b*, *c* and *d* denotes whether this run was statistically significance over the first, second, third and fourth run, respectively.

WSJ and WT2g. The second stage of smoothing was applied (See Equation 3.39). The query model parameter was estimated using the EM algorithm defined in Chapter 3 Section 3.5.4 and the number of EM steps was set to the recommended 10 steps[163]. To denote the use of the second stage (+SS) is added to the base models (BRM and $B\hat{D}M$). Table 5.8 shows the results from applying the second stage.

While the second stage of smoothing appears to ameliorate the mAP of the $B\hat{D}M$, it still does not out perform the BRM+SS, or BRM. Interestingly, the influence of the second stage does not significantly increase performance for the BRM but does so consistently and significantly for the $B\hat{D}M$. It appears that over fitting the data model for BS (and so applying a greater amount document length normalization) is more

effective than trying to account for the query variation. However, a significant increase in mAP only occurs when the second stage is applied to the $B\hat{D}M$. If we consider that the $B\hat{D}M$ requires more smoothing to achieve the same results as the BRM, then the second stage accounts for some of this difference. The BRM, however, is smoothed sufficiently, and this second stage is not really that appropriate anymore as the extra stage smoothing may even begin to degrade the IR performance (as is the case with CACM and WT2g).

5.4.3 Assumption Three

Table 5.9 reports the proportion of relevant documents which are significantly different to the distribution of query likelihoods given non relevant documents. The different significance levels are 0.001 0.01 0.5 0.1 and 0.25 corresponding to 0.1%, 1%, 5%, 10% and 25% of the non-relevant documents in the top 1000, respectively. The values reported in brackets at each significance level is the number of queries where the proportion of relevant documents was greater than a half. Note that this proportion was calculated based on the total number of relevant documents with the top 1000 for each query and not on the total number of relevant documents for that query.

The number of queries which returned over half the proportion of relevant documents was generally very small at the lowest significance levels (0.1% and 1%). It is not until we are willing to see approximately 10% to 25% of the non relevant documents before we obtain half the relevant documents (in this case about 100 to 250 non relevant documents will be amongst the relevant documents. If we are willing to accept that the query is sufficiently discriminative if it returns more than half the relevant documents in the top 10 percent of non relevant documents, then Assumption Three holds for 30 to 40 percent of all queries. However, on many data collections this can be somewhat lower. For instance on the CISI collection it is not until we are willing to accept about 25% of the non relevant document that we obtain half the number of relevant documents.

As we have previously mentioned Assumption One and Assumption Three appear to

Collection	Model	0.001 (0.1%)	0.01 (1%)	0.05 (5%)	0.1 (10%)	0.25 (25%)
MED	LP	0.17 (1)	0.31 (5)	0.48 (14)	0.59 (20)	0.77 (26)
	JM	0.21 (3)	0.31 (9)	0.51 (15)	0.62 (20)	0.74 (26)
	BS	0.19 (2)	0.34 (10)	0.57 (18)	0.67 (23)	0.79 (28)
CACM	LP	0.11 (2)	0.19 (3)	0.35 (11)	0.49 (18)	0.67 (34)
	JM	0.10 (2)	0.23 (6)	0.46 (20)	0.59 (35)	0.75 (45)
	BS	0.13 (2)	0.25 (4)	0.44 (18)	0.54 (27)	0.69 (43)
CISI	LP	0.02 (0)	0.07 (0)	0.19 (3)	0.28 (9)	0.47 (28)
	JM	0.02 (0)	0.08 (0)	0.21 (3)	0.31 (12)	0.53 (47)
	BS	0.03 (0)	0.08 (1)	0.19 (4)	0.27 (6)	0.43 (22)
AP	LP	0.07 (2)	0.16 (8)	0.32 (24)	0.44 (36)	0.64 (72)
	JM	0.06 (2)	0.15 (7)	0.30 (27)	0.42 (44)	0.64 (82)
	BS	0.07 (3)	0.17 (8)	0.34 (31)	0.46 (47)	0.67 (79)
WSJ	LP	0.05 (0)	0.09 (0)	0.22 (8)	0.31 (16)	0.49 (44)
	JM	0.05 (0)	0.15 (4)	0.34 (25)	0.46 (40)	0.66 (78)
	BS	0.08 (0)	0.18 (7)	0.38 (30)	0.51 (52)	0.69 (82)
WT2g	LP	0.07 (0)	0.16 (3)	0.33 (9)	0.43 (13)	0.65 (39)
	JM	0.06 (0)	0.16 (2)	0.36 (11)	0.50 (27)	0.69 (41)
	BS	0.08 (0)	0.20 (5)	0.41 (19)	0.55 (30)	0.73 (45)

Table 5.9: A3: The proportion of queries which showed sufficient discrimination at the various levels of significance. As the level of significance decreases, the proportion of queries that sufficiently discriminate increases.

be dependent. (i.e, if the query terms used are good at discriminating relevant from non-relevant, then we would expect a strong correlation between the Odds Ratio and the query likelihood, and possibly vice versa.) To investigate this dependency we believed that issuing queries which were consistent with the Third Assumption, would illuminate the dependency with Assumption One.

5.4.3.1 Ideal Queries

To test the dependency, follow up experiments were undertaken where we used 'ideal' queries. 'Ideal' queries are queries which have been generated in accordance with the third assumption of the Language Modeling approach. Hence, the ideal query is generated from the ideal documents (the relevant documents). 'Ideal' queries were generated by taking all the relevant documents associated with the original query and assuming that this was the 'ideal' document (or set of) which the user has envisioned. We then employed two different methods for generating queries from the set of relevant documents, with respect to A3.2 (i.e. general terms) and A3.1 (i.e discriminative terms):

Q1 We selected the top ten most frequent terms in the set of relevant documents, and;

Q2 We selected the top ten terms which were the most discriminative terms (as defined by Equation 3.57). Terms that appeared less then 5 times in the set of relevant documents were excluded, so not to include terms that were too specific (and occurring in too few relevant documents).

Table 5.10 gives some examples of the ideal queries. The first method (Q1) does not account for how discriminative the terms are, just how popular they are. Remember, we have excluded stop words from our data collections, so we are not just issuing a set of stop words as a query, just very frequently occurring terms in relevant documents. The queries generated by Q1 bear much resemblance to the original queries (Q0) posed. The second method, assumes that the user has more knowledge of the collection and can select terms which can discriminate relevant from non relevant. These queries

appear less intuitive and more specific than Q1, as they contain specific entities (names of people, components, companies, places, and even part numbers) which were directly related to the information need (i.e the missiles, the terrorists, and the hackers.)

We generated queries for the AP, WSJ and WT2g collections and then re-assessed Assumption One and Three using the JM and BS document models. The parameter values for JM and BS were set to that which obtained the $B\hat{D}M$.

Table 5.12, Table 5.13 and Table 5.11 contain the results for Assumption One, Assumption Three and the corresponding performance scores, for each of the different query types (Q0, Q1 and Q2).

The most striking result is that the mAP and p@30 for Q1 queries is significantly higher than the original Q0 and ideal Q2 queries on the AP and WSJ. This was quite a surprising result because we would have expected that Q2 queries would have performed the best as they were generated in a manner consistent with the Third Assumption using more information about the terms to select. The Q1 queries seem more reasonable in that a user could possibly formulate such queries. Here less specific and intimate information about the relevant documents and the terms within them is assumed. Issuing such queries improves the correlation in Assumption One, but still only holding about half the time (and this is despite the sizable increase in IR performance).

On the other hand, for Q2 queries the early precision p@0% was the highest (almost 100 percent) suggesting that relevant documents were almost always returned at the top. From Table 5.13, we can see that Q2 has a lower recall of relevant documents, suggesting that the query terms were perhaps too specific, only in a few of the relevant documents. This meant that the terms used gave too much discrimination to return the majority of relevant documents, only those specific documents which contained those query terms. Nonetheless, issuing Q2 queries increases the correlation of Assumption One at 30 and 100 documents, holding for about three out of every four queries. It would appear that issuing queries which contain the query terms that discriminate relevant from non relevant tended to produce a ranking more correlated with the Odds Ratio. However, since the terms are very specific the recall is adversely affected and

Topic	Type	Query/Query Terms
101	Title	Design of the 'Star Wars' Anti-missile Defence System
	Q0	design star war anti missil defens system
	Q1	year war star defens missil billion test program space base
	Q2	pbfa npb r2p2 sbi schriber otten rocketdyn interceptor monahan abrahamson
119	Title	Action Against International Terrorists
	Q0	action intern terrorist
	Q1	state unit year terrorist offici nation report govern attack peopl
	Q2	vigneron khaidir overholt zehdi khadar zozad terwillig terzi sidra kikumura
190	Title	Instances of Fraud Involving the Use of a Computer
	Q0	instanc fraud involv comput
	Q1	comput year hacker charg govern state system compani million code
	Q2	dicicco mitnick landreth pfaelzer zinn doucett ingraham newsham huse flori

Table 5.10: Some examples of 'ideal' queries executed on the AP collection. Query terms are shown as their stemmed form. The Q1 queries appear more intuitive than the Q3 queries which would seem to require much more intimate knowledge of the terms in the collection.

Collection	Model	Query	mAP	p@0%	p@30docs	Recall
AP	BS	Q0	25.85	68.33	35.67	6271
		Q1	**39.49**	85.39	48.60	7499
		Q2	29.60	**97.51**	46.67	2830
	JM	Q0	24.84	67.68	34.03	6284
		Q1	**38.41**	80.06	46.60	7509
		Q2	32.85	98.23	50.07	3524
WSJ	JM	Q0	24.06	64.54	32.00	5330
		Q1	**29.94**	66.67	35.33	5363
		Q2	29.21	**95.89**	42.6	2468
	BS	Q0	26.37	68.90	35.47	6367
		Q1	**31.59**	74.50	38.00	6377
		Q2	25.16	**93.18**	37.23	1907
WT2g	JM	Q0	21.41	63.5	25.47	1789
		Q1	25.582	91.25	30.27	1426
		Q2	24.06	**98.79**	27.27	523
	BS	Q0	**20.70**	68.84	25.47	1709
		Q1	18.44	81.86	22.26	1054
		Q2	9.03	**83.5**	11.07	176

Table 5.11: The IR performance when using the different query types on AP8889 and WSJ for JM and BS document models. Notice that for Q1 queries high mAP and Recall is obtained, whilst for the Q3 queries a high p@0% is obtained but the very poor recall lowers the mAP.

Collection	Model	Type	Pos @30	Pos @100	Pos @1000	Neg @30	Neg @100	Neg @1000
AP	JM	Q0	0.16	0.55	0.98	0	0	0
		Q1	0.21	0.68	0.98	0	0	0
		Q2	0.63	0.79	0.46	0	0	0
	BS	Q0	0.25	0.64	0.93	0	0	0
		Q1	0.46	0.86	0.99	0	0	0
		Q2	0.61	0.89	0.97	0	0	0
WSJ	JM	Q0	0.07	0.45	0.93	0	0	0
		Q1	0.16	0.50	0.96	0	0	0
		Q2	0.62	0.77	0.38	0	0	0
	BS	Q0	0.31	0.63	0.91	0	0	0
		Q1	0.28	0.79	0.98	0	0	0
		Q2	0.53	0.92	0.91	0	0	0
WT2g	JM	Q0	0.22	0.46	0.92	0	0	0
		Q1	0.56	0.88	0.98	0	0	0
		Q2	0.34	0.22	0.02	0	0	0
	BS	Q0	0.32	0.52	0.84	0	0	0
		Q1	0.60	0.92	0.98	0	0	0
		Q2	0.16	0.08	0	0	0	0

Table 5.12: The proportion of positive and negative correlations between the Odds Ratio and the Query Likelihood at Recall of 30, 100 and 1000 documents, for each TREC collection using original and ideal queries. Notice the increasing proportion of correlations as the number of documents increased, except for the Q2 queries which degrades.

Col.	Model	Type	0.001 (0.1%)	0.01(1%)	0.05 (5%)	0.1 (10%)	0.25 (25%)
AP	JM	Q0	0.06 (2)	0.15 (7)	0.30 (27)	0.42 (44)	0.64 (82)
		Q1	0.08 (3)	0.19 (22)	0.41 (50)	0.54 (65)	0.72 (94)
		Q2	0.23 (22)	0.28 (29)	0.41 (44)	0.47 (59)	0.63 (84)
	BS	Q0	0.07 (3)	0.17 (8)	0.34 (31)	0.46 (47)	0.67 (79)
		Q1	0.09 (5)	0.21 (22)	0.43 (51)	0.56 (68)	0.74 (94)
		Q2	0.25 (21)	0.30 (28)	0.43 (47)	0.52 (61)	0.70 (85)
WSJ	JM	Q0	0.05 (0)	0.15 (4)	0.34 (25)	0.46 (40)	0.66 (78)
		Q1	0.07 (3)	0.19 (16)	0.40 (41)	0.52 (62)	0.70 (82)
		Q2	0.26 (24)	0.32 (31)	0.46 (52)	0.52 (65)	0.65 (84)
	BS	Q0	0.08 (0)	0.18 (7)	0.38 (30)	0.51 (52)	0.69 (82)
		Q1	0.09 (3)	0.21 (16)	0.42 (41)	0.55 (67)	0.72 (85)
		Q2	0.25 (16)	0.30 (22)	0.44 (48)	0.52 (57)	0.68 (83)
WT2g	JM	Q0	0.06 (0)	0.16 (2)	0.36 (11)	0.50 (27)	0.69 (41)
		Q1	0.10 (2)	0.24 (10)	0.42 (21)	0.53 (30)	0.66 (38)
		Q2	0.62 (33)	0.63 (34)	0.69 (36)	0.73 (42)	0.81 (46)
	BS	Q0	0.08 (0)	0.20 (5)	0.41 (19)	0.55 (30)	0.73 (45)
		Q1	0.11 (10)	0.24 (16)	0.42 (22)	0.53 (29)	0.63 (36)
		Q2	0.68 (38)	0.68 (38)	0.71 (39)	0.72 (40)	0.81 (42)

Table 5.13: A3 for perfect queries: The proportion of relevant documents and number of queries shown in brackets which showed sufficient discrimination at the various levels of significance.

the correlation at 1000 documents deteriorated.

These results suggest that there are two distinct querying approaches, a recall oriented strategy and a precision oriented strategy. The former requires that the user issue key terms that would frequently occur in the set of relevant documents. The latter requires the user to issue key terms which are distinct and highly discriminative, which will bring back a specific subset of relevant documents. This leads to an interesting insight; if the query terms are highly discriminative then the early documents are very likely to be relevant (and would make an ideal source for pseudo relevance feedback). However, to improve recall, the more frequent terms of pseudo relevant documents should be issued.

Another possibility is that ranking by the query likelihood actually provides a different ranking of documents than the Odds Ratio. While it still returns relevant document at early levels of recall, the ranking is not consistent with the Odds Ratio, hence the ordering is not going to adhere to the PRP. There are occasions (for specific queries) when a query biased view of the collection is presented, which does not correlate with the Odds Ratio, but still provides sufficient discrimination.

Table 5.14 shows the number of queries which uphold Assumption One and Assumption Three (or not). The correlation in Assumption One was computed at 100 documents and the sufficient discrimination in Assumption Three was quantified at significance level of 10% where at least half the relevant documents need to have been retrieved. The last four columns in the Table 5.14 represent when both assumptions hold (A1* and A3*), when only either assumptions holds, or when neither assumption holds.

When Q1 and Q2 were used the number of times both assumptions succeeded increased, Q2 type queries increasing more, while the number of complete failures (i.e both assumption One and Two failing) decreased. Interestingly, there were hardly any occasions when Assumption Three held, and Assumption One did not. This suggests that when Assumption Three holds, then the ranking is more likely to be equivalent to the Odds Ratio.

Collection	Model	Type	A1* A3*	A1* Only	A3* Only	Neither
AP	LP	Q0	18	40	6	36
	JM	Q0	17	38	10	35
		Q1	46 (+29)	22 (-16)	4 (-6)	28 (-7)
		Q2	35 (+18)	44 (+6)	9 (-1)	12 (-23)
	BS	Q0	24	40	7	29
		Q1	50 (+26)	36 (-4)	1 (-6)	13 (-16)
		Q2	46 (+22)	43 (+3)	1 (-6)	10 (-19)
WSJ	LP	Q0	8	59	0	33
	JM	Q0	20	25	5	50
		Q1	33 (+13)	17 (-8)	8 (+3)	42 (-8)
		Q2	37 (+17)	40 (+15)	15 (+10)	8 (-42)
	BS	Q0	23	40	7	30
		Q1	39 (+16)	40 (+0)	2 (-5)	19 (-11)
		Q2	47 (+24)	45 (+5)	1 (-6)	7 (-23)
WT2g	LP	Q0	7	14	2	27
	JM	Q0	6	17	5	22
		Q1	19 (+11)	3 (-14)	3 (-4)	25 (+3)
		Q2	3 (-3)	8 (-9)	33 (+28)	6 (-8)
	BS	Q0	12	14	7	17
		Q1	18 (+6)	8 (-6)	2 (-5)	22 (+5)
		Q2	2 (-10)	2 (-12)	37 (+30)	9 (-8)

Table 5.14: The relationship between A1 and A3. The ideal queries tend to increase the number of times when A1 holds and A3 hold and decrease the number of times when the neither A1 or A3 holds.

Overall the number of times Assumption One held for the ideal queries increased, regardless of whether Assumption Three held (i.e the addition of columns 4 and 5 in Table 5.14) over the initial queries. Hence, the ideal queries produced rankings that were more inline with the Odd's Ratio, however the discrimination between relevant and non-relevant was not as high as expected. This would depend on how the sufficient discrimination in A3 is quantified, though.

To a large extent the quality and style of the query affected whether the Assumptions hold. The impoverished initial queries (Q0) resulted in many more complete failures of A1 and A3 than the ideal queries. Suggesting that if better queries are formed, then better retrieval performance should follow.

5.5 Summary of Findings

Whilst our analysis of the underlying assumptions of Language Modeling raises more questions than it answers, it nonetheless provides some interesting and valuable insights into the retrieval model (and its basis).

A1 The Correlation between the query likelihood and relevance of a document appears to hold when the queries are well formed. That is, when consistent with the Third Assumption. Otherwise the query likelihood will not be correlated with the relevance of a document. The correlation tends to strengthen as the number of documents seen increases. However, this is of limited utility because the correlation is strengthened by the influx of non relevant documents with low query likelihoods and low relevance scores.

A2 The unification of the data and retrieval models holds reasonably well for JM, but breaks down for LP and BS. When the breakdown occurs on the LP and BS document models it usually results in a significant difference in terms of IR performance. Whilst the mean predictive likelihood provides an unsupervised and principled means to estimate the parameters of the document models, it may result in sub optimal retrieval. Possibly, if we used more sophisticated docu-

ment modeling techniques[160, 61, 12] or techniques that use context[5] are employed then perhaps better representations of the underlying data could be obtained which deliver better IR performance.

A3 The quality and type of queries issued will dictate the success of the language model. If a user issues terms that commonly occur in relevant documents then recall can be dramatically improved. However, if the terms are too common then the precision and recall will be seriously degraded. Conversely, if the terms issued are very discriminative then this will invariably lead to high early precision but substantially lower recall. This may be useful when performing pseudo-relevance feedback.

To summarize, the Language Modeling approach represents an appealing paradigm for Information Retrieval because of its intuitive behavior; that is, ranking documents according to how likely the query came from a document. However, this requires assumptions to be made about the notion of relevance, (because it is not explicitly modeled). The extent to which these assumptions hold appears to be dependent on the query and the data model.

In the next chapter, we implement our context based document models, which attempt to improve the representation of the underlying data, with respect to the user's understanding. This extends our work investigating Assumption Two. Does a better representation of a document model based on the user's context actually deliver better IR performance?

Chapter 6

Context Experiments

In this chapter, we present our empirical analysis of context based document models. The chapter is divided into three sections; In each section we employ a different type of semantic association to define a document's context according to the user's understanding of the documents in the collection. First, we use PLSA to build context based document models on the three abstract collections. Second, we use the topics that users have been following to define the context and compare this against cluster based document models on the two news collections. Third, we use the web links associated with a web document as the context of the document on the WT2g collection. The evaluations performed within this Chapter are presented with respect to the Second Assumption and its implication for retrieval performance, that is, a better generative document model will entail better retrieval performance.

6.1 Induced Associations

In this section, we present our work on context based document models that are defined by inducing a topical structure from the corpus of documents. This was achieved by using Probabilistic Latent Semantic Analysis[61] as defined in Chapter 4.4.2. While PLSA is actually a machine learning algorithm, it has been hailed for its ability to induce latent factors which are interpretable by humans. That is, the terms that are

highly associated with a particular latent factor are topically related; and resemble the terms that a user would associate with that topic. Each document is then described as a combination of these latent factors. We believe it is reasonable to accept that these latent factor can be substituted for topics that capture the user's understanding of the collection. Further, that these can be used to define the semantic association between documents, through their topical associations. As we have already mentioned PLSA has already been used for *ad hoc* Information Retrieval making similar such assumptions. Our evaluation is essentially a replication of this past work, but with several key differences which we shall detail later in this section.

Given the PLSA model defined in Equation 4.17 where a topic is represented as the latent variable z, then the context parameters for the context model can be defined as $\Theta_Z = \{p(t|z), p(z|d)\}$. Therefore, the context for a particular document is defined as:

$$p_d(t|\Theta_Z) = \sum_z p(t|z)p(z|d) \tag{6.1}$$

A context background model $p_d(t|\theta_Z)$ is constructed by substituting $p_d(t|\Theta_Z)$ into Equation 4.2 for $p_d(t|\Theta_Z)$, which applies a proportion π of smoothing with the background collection model.

The context based document model is then created by using the general form (Jelinek-Mercer) and Bayes smoothed document models shown in Equation 4.3 and Equation 4.4, respectively. Where the former shall be referred to as JM-PLSA and the later BS-PLSA. For the general form (Jelinek Mercer), the complete estimation of the document model is:

$$p_d(t|\theta_d^Z) = (1-\lambda)\frac{n(t,d)}{n(d)} + \lambda\left((1-\pi)\left(\sum_z p(t|z)p(z|d)\right) + \pi p(t|\theta_C)\right) \tag{6.2}$$

The parameters that are required to be estimated are λ,π, $p(t|z)$ and $p(z|d)$. The later two are estimated given the outlined procedure in Section 4.4.2, where the former are estimated according to the average Leave One Out Log Likelihood. To obtain the Bayes Smoothed form, λ is replaced by $\frac{\beta}{n(d)+\beta}$. The following subsections detail our

empirical analysis of the context based document models using the induced associations derived from the PLSA model.

6.1.1 Experimental Settings

Due to the computation expense involved in applying the estimation procedure for PLSA we restricted our analysis to the three smaller abstract collections (MED, CACM, CISI)[1]. The details of these collections can be found in Section 5.3.1 and collection statistics in Table 5.1.

The baseline document models used were Jelinek Mercer (JM) Smoothing (see Equation 3.25) and Bayes Smoothing (BS) (see Equation 3.32). These were then compared against the PLSA-JM and PLSA-BS document models. The set of parameter values for the corresponding parameters used are given in Table 5.2. For the PLSA document models, the set of parameter values used for π were $\{ 0.1, 0.2, 0.3, \ldots, 0.9\}$, and the number of topics $|Z|$ were $\{ 16, 32, 64, 128\}$. Whilst, the number of topics could be any positive integer value greater than or equal to one, these values were selected based on the prior research in [61]. In fact, it is still an open problem as to how to select the optimal number of topics that maximizes the retrieval performance. However, according to Assumption Two, the data model that provides the best representation of the underlying data should also provide the best retrieval performance. Hence, under our approach to context based document modeling, we should be able to select the number of latent topics according to mPL of the document models when generated using different numbers of latent topics (i.e select the number of latent topics that maximized the mPL).

6.1.1.1 Estimation Details

The procedure for estimating the PLSA context based document models is slightly different to that prescribed in our earlier chapter. We have used the PENN-ASPECT

[1]These data sets were used in the original studies[61, 60].

Implementation within the LEMUR Toolkit[2]. This implementation of PLSA estimates the $p(t|z)$ and $p(z|d)$ by maximizing the log likelihood of the PLSA model on a held out sample of data (see Section 4.4.2). This is not particularly different from the leave-one-out log likelihood (i.e. mPL) except instead of only leaving one out at a time, a percentage of the document is extracted and used as the held out sample. This approach drastically reduces the amount of computational expense required (i.e one PLSA model is estimated given the training data, and then tested, as opposed to one PLSA model is estimated per term in the document, and then each model is tested). However, it is generally considered more thorough to use the leave one out method, but computationally expensive.

The PLSA model was estimated using the Tempered EM algorithm defined in Section 4.4.2 where $\gamma = 0.95$, the maximum number of EM steps was step to 100, and the held out sample was ten percent of the document's contents. Since PLSA is a non-linear optimization up to ten different initializations were performed and the model which reported the highest average log likelihood was selected to build the context based document model for each $|Z|$.

The context obtained under PLSA is expressed as a multinomial probability distribution as a opposed to a distribution over the vocabulary of term frequencies which are then normalized to obtain a multinomial probability distribution for a topic. Hence, the π and λ or β parameters were then estimated by performing a two way exploration of the parameter space $|\pi| \times |\lambda|$ or $|\pi| \times |\beta|$ and computing the average leave one out log likelihood. The parameter values that gave the highest average leave one out log likelihood were selected. It was a concern that several local maxima may exist in the parameter space, however this did not appear to be the case. The contour plots, shown in Figure 6.1, suggest that there is only one maxima in the search space. For the PLSA-JM models this was when $\pi = 0.9$ and $\lambda = 0.5 - 0.6$, and for the PLSA-BS models this was approximately $\pi = 0.9$ and $\beta = 300 - 1000$. Interestingly, this was regardless of the given collections and the various number of topics used.

With respect to Assumption Two, these points should provide the best representation

[2]LEMUR: Language Modeling and Retrieval Toolkit available from www.lemurproject.org.

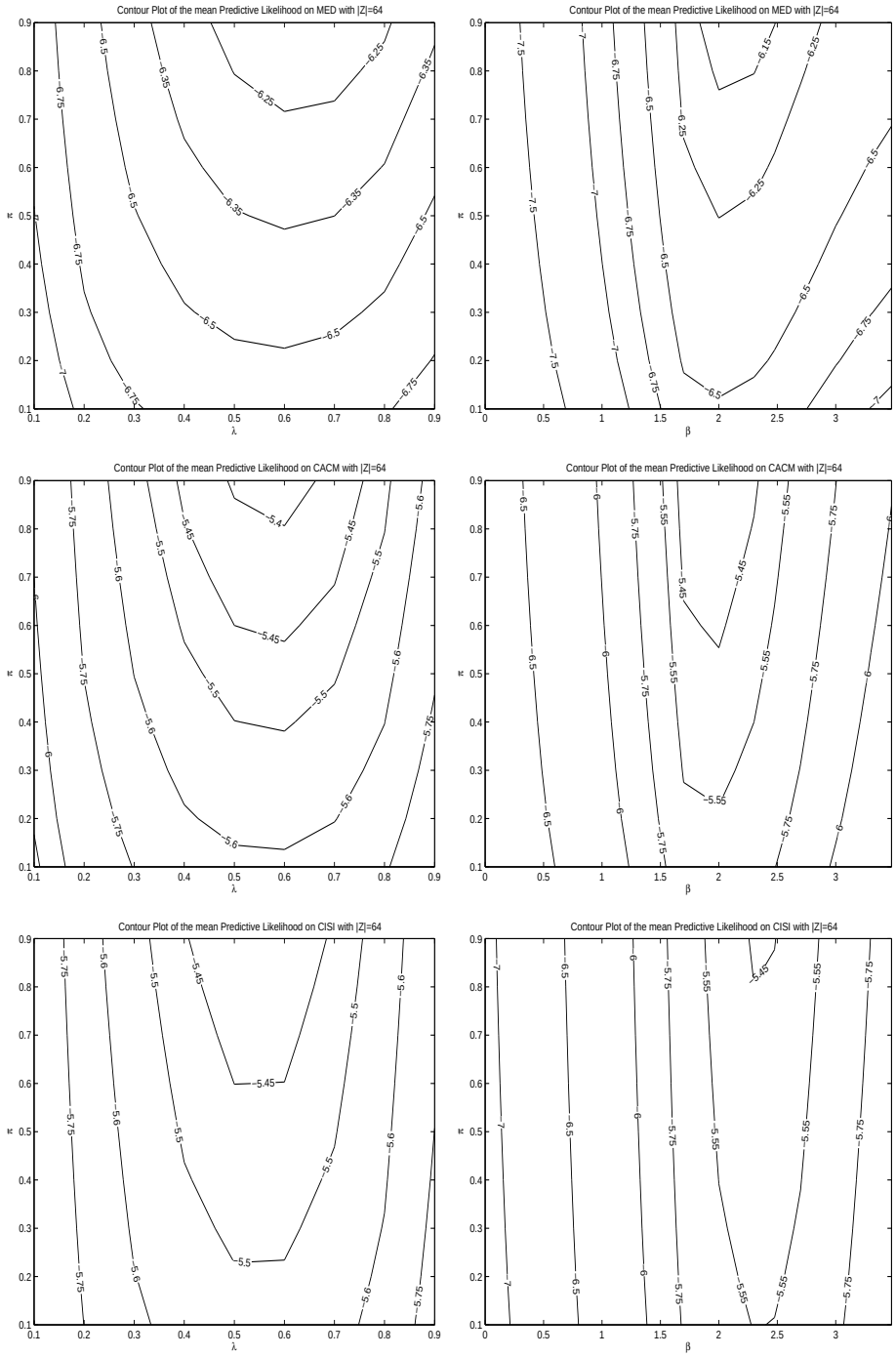

Figure 6.1: Contour plots of the average leave one out log likelihood where $|Z| = 64$. Left Side: PLSA-JM Right Side: PLSA-BS Top: MED Middle: CACM: Bottom: CISI. The small plateau indicates the region of highest mPL.

of the underlying data which takes into consideration the user's understanding. Hence, we first tested whether a better representation over the standard language models were obtained, and then compared the IR performance. Then we considered whether better retrieval performance could be obtained elsewhere in the parameter space.

6.1.2 Results

In Tables 6.1, 6.2 and 6.3 we report the results comparing the standard versus the context based models by providing statistics on the mPL, mAP, Precision at ten percent Recall (p@10%), the p@30 docs and the total number of relevant documents retrieved. The standard models (JM and BS) were selected according to the best data model. Whilst we report for the PLSA models, we show the performance for each number of topics, and when the context parameter π is set to 0.5 (i.e half from the document's context and half from the collection background model) or when the context parameter $\pi = 0.9$ (i.e gave the best data model). The Sign Rank test was again employed to determine whether there was any significant difference between the baseline and the PLSA models. Asterisks denote whether the values were statistically significant at 95% confidence.

From the Tables 6.1, 6.2 and 6.3, it can be seen that the mPL for each of the context based document models is higher than the corresponding standard model. The differences in mPL on each occasion was significantly different using a Sign rank test (at 5% significance). Hence, by encoding the user's understanding we were able to produce a significantly better representation of the underlying data according to the mPL. This was for all the PLSA models over the baseline models. However, the retrieval performance varied between collections and models.

On the MED collection a significant increase in performance was obtained by the estimated set of parameters using either PLSA-JM for all $|Z|$ and PLSA-BS for $|Z| = 64$ and $|Z| = 128$. However, the best overall IR performance in terms of mAP was obtained when we manually set the parameters for PLSA-JM ($|Z| = 32$,$\lambda = 0.6$ $\pi = 0.5$) and PLSA-BS ($|Z| = 64$,$\beta = \pi = 0.5$) which obtained mAP of 61.75% and 61.82% re-

Model	Parameters	mPL	mAP	p@10%	p@30 Docs	Recall
JM	$\lambda = 0.565$	-6.6765	52.06	94.36	41.78	695/696
PLSA-JM	$\pi = 0.5\ \lambda = 0.5$	-6.4791	58.52*	93.73	47.22*	695/696
$z = 16$	$\pi = 0.1\ \lambda = 0.6$	-6.3686	57.52*	84.59	45.78*	695/696
PLSA-JM	$\pi = 0.5\ \lambda = 0.6$	-6.3742	61.75*	84.73	50.00*	695/696
$z = 32$	$\pi = 0.1\ \lambda = 0.6$	-6.2354	59.41*	84.23	46.89*	695/696
PLSA-JM	$\pi = 0.5\ \lambda = 0.7$	-6.3361	61.61*	90.24	50.78*	695/696
$z = 64$	$\pi = 0.1\ \lambda = 0.6$	-6.1945	58.56*	84.13	46.56*	695/696
PLSA-JM	$\pi = 0.5\ \lambda = 0.6$	-6.2285	58.98*	82.39	47.89*	695/696
$z = 128$	$\pi = 0.1\ \lambda = 0.7$	-6.0628	57.16*	82.11	46.33*	695/696
BS	$\beta = 100$	-6.7235	49.95	94.35	41.33	695/696
PLSA-BS	$\pi = 0.5\ \beta = 200$	-6.4868	56.73*	92.29	46.22	695/696
$z = 16$	$\pi = 0.1\ \beta = 100$	-6.3779	53.30	79.75	44.00	695/696
PLSA-BS	$\pi = 0.5\ \beta = 200$	-6.2852	61.06*	89.94	50.11*	695/696
$z = 32$	$\pi = 0.1\ \beta = 100$	-6.1508	57.25	81.94	48.00	695/696
PLSA-BS	$\pi = 0.5\ \beta = 300$	-6.2478	61.82*	91.55	50.00*	695/696
$z = 64$	$\pi = 0.1\ \beta = 100$	-6.1106	59.55*	84.85	48.89*	695/696
PLSA-BS	$\pi = 0.5\ \beta = 100$	-6.2381	58.03*	91.02	47.78*	695/696
$z = 128$	$\pi = 0.1\ \beta = 100$	-6.0781	55.20*	80.97	47.44*	695/696

Table 6.1: The results for using PLSA-JM and PLSA-BS on MED. The asterisk indicates whether there was there was a significance different between the baseline and PLSA model.

Model	Parameters	mPL	mAP	p@10%	p@30 Docs	Recall
JM	$\lambda = 0.55$	-5.6302	28.315	66.6076	18.5256	720/796
PLSA-JM	$\pi = 0.1\ \lambda = 0.8$	-5.4573	29.04	69.89	18.65	718/796
$z = 16$	$\pi = 0.1\ \lambda = 0.6$	-5.3546	28.21	55.21	18.59	721/796
PLSA-JM	$\pi = 0.5\ \lambda = 0.9$	-5.5105	30.26	71.66	18.97	699/796
$z = 32$	$\pi = 0.1\ \lambda = 0.6$	-5.4301	28.96	54.74	18.85	723/796
PLSA-JM	$\pi = 0.1\ \lambda = 0.9$	-5.4662	29.91	70.82	18.85	727/796
$z = 64$	$\pi = 0.1\ \lambda = 0.6$	-5.3845	28.72	55.30	18.65	723/796
PLSA-JM	$\pi = 0.5\ \lambda = 0.9$	-5.3953	29.83	68.28	18.78	681*/796
$z = 128$	$\pi = 0.1\ \lambda = 0.6$	-5.3069	28.77	55.29	18.91	722/796
BS	$\beta = 82.03$	-5.6208	23.52	64.42	16.99	603/796
PLSA-BS	$\pi = 0.5\ \beta = 300$	-5.4516	26.43*	68.33	17.56	664*/796
$z = 16$	$\pi = 0.1\ \beta = 100$	-5.3498	23.81	63.92	16.22	648*/796
PLSA-BS	$\pi = 0.5\ \beta = 1000$	-5.5063	31.65*	74.75	20.00*	678*/796
$z = 32$	$\pi = 0.1\ \beta = 100$	-5.4274	27.24*	68.17	18.27*	643*/796
PLSA-BS	$\pi = 0.5\ \beta = 500$	-5.4628	31.00*	73.40	19.55*	676*/796
$z = 64$	$\pi = 0.1\ \beta = 100$	-5.3827	26.86*	68.02	18.01*	638*/796
PLSA-BS	$\pi = 0.5\ \beta = 500$	-5.3919	30.29*	71.61	19.55*	660*/796
$z = 128$	$\pi = 0.1\ \beta = 100$	-5.3047	27.16	67.11	17.76	635*/796

Table 6.2: The performance statistics for the PLSA context based document models on CACM collection.

spectively. Corresponding to an 18.6% and 23.8% percent increase over the respective baselines.

On the CACM collection significantly better mAP was obtained only when using the PLSA-BS document models. However, the estimated PLSA-BS document models only obtained significantly better mAP over the baseline BS performance, when $|Z| =$ 32 or 64. Though when we compared these results to the baseline JM performance they were not significantly different. Only by manually setting the parameters could the best mAP be obtained.

On the CISI Collection, the estimated parameter settings failed to deliver superior IR

Model	Parameters	mPL	mAP	p@10%	p@30 Docs	Recall
JM	$\lambda = 0.5$	-5.519	14.06	49.41	17.85	2834/3114
PLSA-JM	$\pi = 0.1$ $\lambda = 0.1$	-5.309	14.51*	51.57	17.72	2871*/3114
$z = 16$	$\pi = 0.1$ $\lambda = 0.6$	-5.223	14.09	29.33	17.85	2830/3114
PLSA-JM	$\pi = 0.1$ $\lambda = 0.1$	-5.2377	14.50*	51.72	17.81	2881*/3114
$z = 32$	$\pi = 0.1$ $\lambda = 0.5$	-5.1962	14.18	30.07	17.85	2865/3114
PLSA-JM	$\pi = 0.1$ $\lambda = 0.1$	-5.0613	14.47*	52.33	17.81	2880*/3114
$z = 64$	$\pi = 0.1$ $\lambda = 0.5$	-5.0418	14.13	29.96	17.94	2854/3114
PLSA-JM	$\pi = 0.1$ $\lambda = 0.1$	-5.1228	14.49*	52.30	17.85	2877*/3114
$z = 128$	$\pi = 0.1$ $\lambda = 0.6$	-5.0777	14.05	29.72	17.68	2843/3114
BS	$\beta = 200$	-5.5398	10.87	44.87	15.16	2536/3114
PLSA-BS	$\pi = 0.1$ $\beta = 500$	-5.3302	12.99	49.76*	15.53	2573/3114
$z = 16$	$\pi = 0.1$ $\beta = 200$	-5.245	11.50	44.96	14.91	2544/3114
PLSA-BS	$\pi = 0.5$ $\beta = 1000$	-5.2586	13.17*	50.16*	17.24*	2602*/3114
$z = 32$	$\pi = 0.1$ $\beta = 200$	-5.2169	11.78*	46.22*	15.53	2561/3114
PLSA-BS	$\pi = 0.5$ $\beta = 1000$	-5.182	12.59*	49.47*	16.80	2546/3114
$z = 64$	$\pi = 0.1$ $\beta = 200$	-5.0421	10.54	23.80	15.22	2521/3114
PLSA-BS	$\pi = 0.5$ $\beta = 1000$	-5.1431	12.19*	48.11*	16.45	2565*/3114
$z = 128$	$\pi = 0.1$ $\beta = 200$	-5.0977	10.41	23.65	14.74	2527/3114

Table 6.3: The performance statistics for the PLSA context based document models on CISI collection.

performance (only obtaining up to 14.18% mAP when $|Z| = 32$, $\lambda = 0.1$ and $\pi = 0.5$). However, manually setting the parameters obtained a statistically significant increases given any $|Z|$ and regardless of document model type (PLSA-JM or PLSA-BS). The best overall performance was obtained by setting the parameter of PLSA-JM to $\pi = 0.1$, $\lambda = 0.1$ resulted in a mAP of 14.51%, and setting the parameters of PLSA-BS to $\pi = 0.5$ $\beta = 1000$ obtained a mAP of 13.17%.

In the Figures 6.2 and 6.3, the graphs show the change in mAP and mPL given the smoothing parameter λ or β for different values of $|Z|$ on the MED and CACM collections. The divergence between the parameter value which obtained the BDM and BRM models using PLSA based document models was noticeably larger than for the baseline models examined earlier (See Figures 5.5 and 5.6 for comparison).

From a different point of view, the graphs in Figure 6.4 show the change in mPL vs mAP across the number of latent factors, $|Z|$. As the $|Z|$ increased, the mPL increased which tended to convergence. On the CACM and CISI collection there is a small drop at $|Z| = 128$ though we attribute this to not performing enough randomizations or tuning of the tempering parameters to enable a small increase in mPL. On the other hand the mAP reached a maximum around $|Z| = 16 - 32$, after which the mAP dropped. From the mPL we would select the higher order models, however this will not ensure the best retrieval model is actually selected. However, there does appear to be a systematic relationship between the mPL and mAP across the collections[3].

6.1.3 Discussion

Whilst we have been able to consistently generate context based document models that provide a better representation of the underlying than the base line document models, according to the mPL, we have met limited success in terms of retrieval performance. When we used the mPL criteria for model selection, the corresponding estimated parameter values, did not always deliver significantly better retrieval performance. It was only when we manually assessed the different parameter value combinations that

[3]We report similar findings in [3].

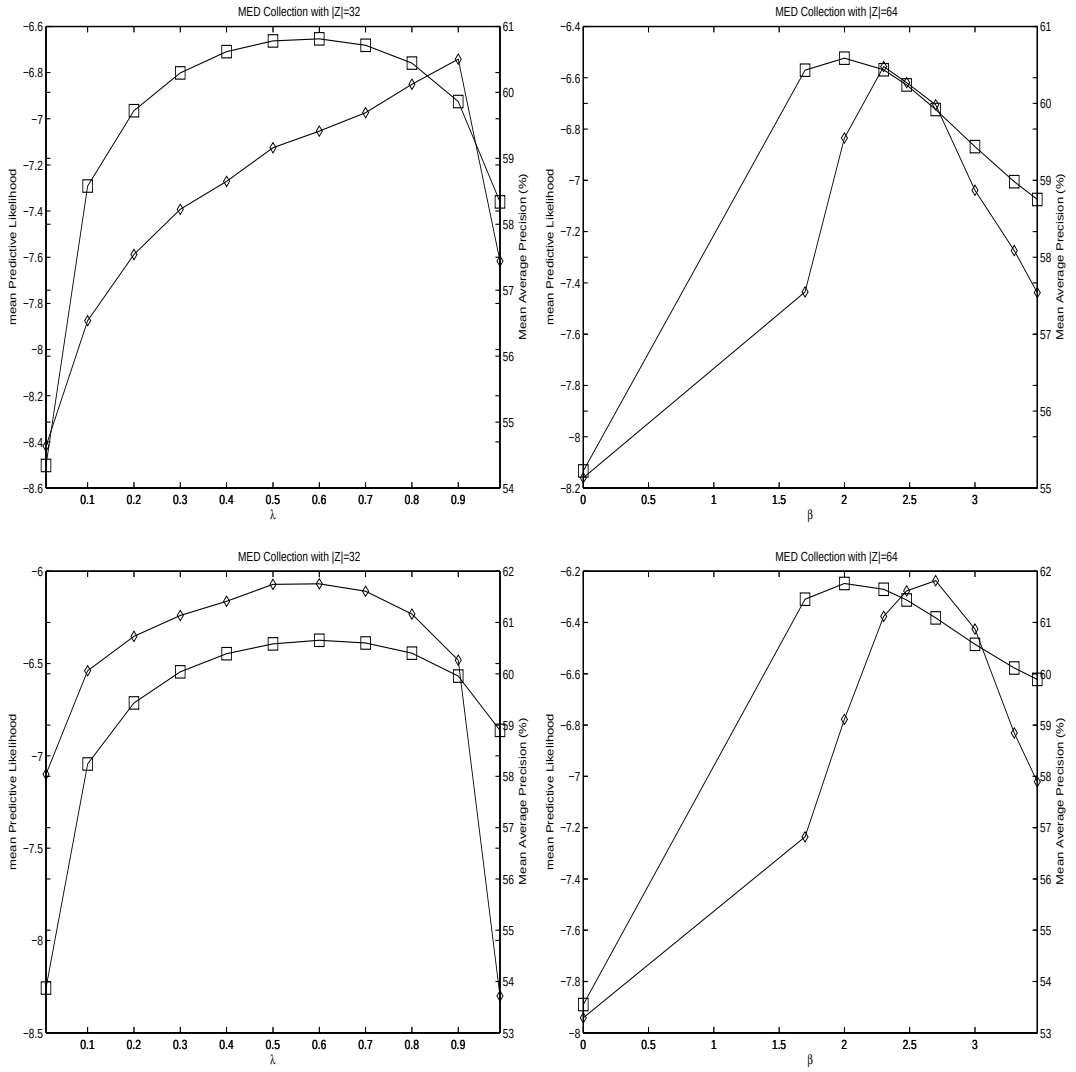

Figure 6.2: mPL vs mAP: MED Left: PLSA-JM given $|Z| = 32$ Right: PLSA-BS given $|Z| = 64$. Notice the divergence between the BRM and BDM under the PLSA models.

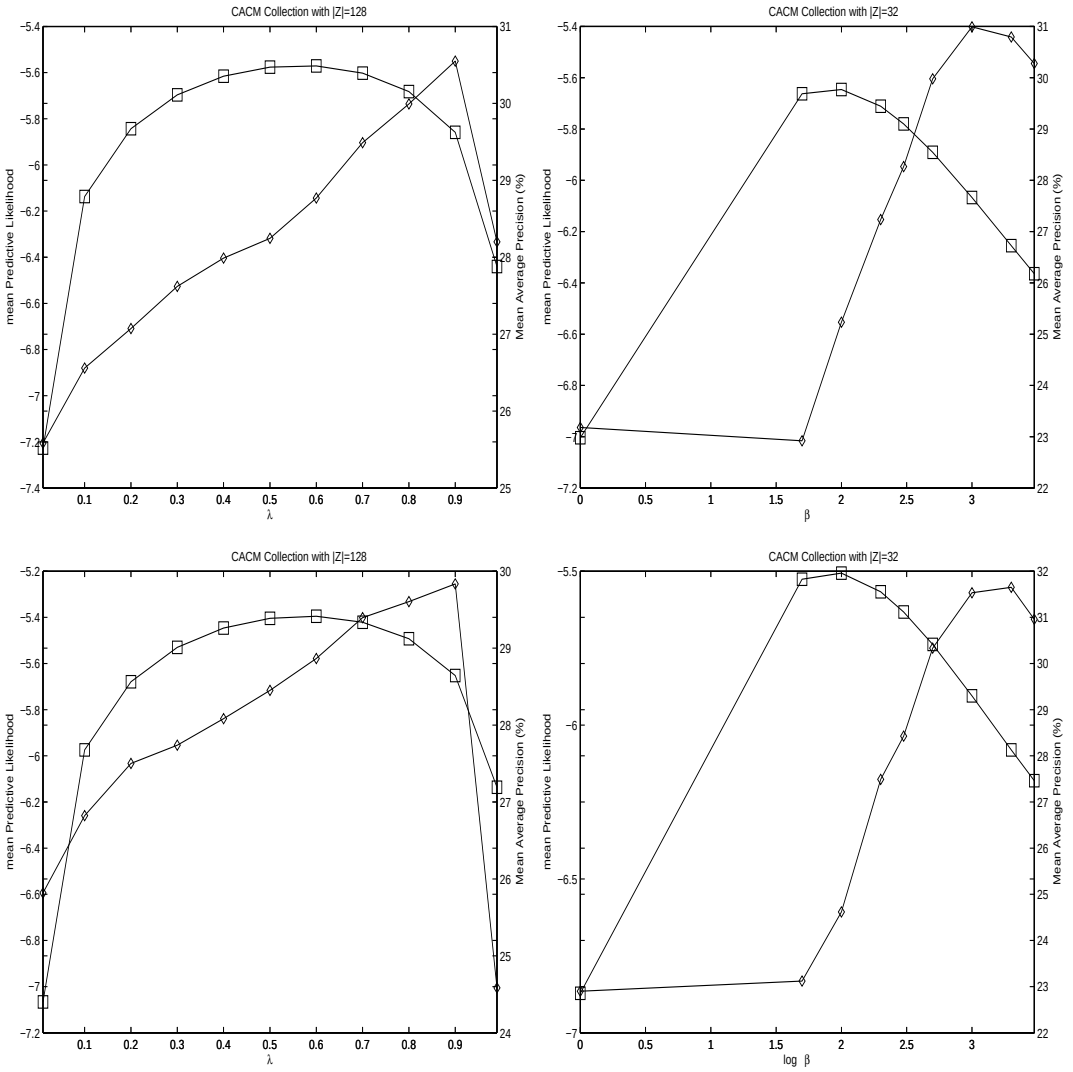

Figure 6.3: mPL vs mAP: CACM Left: PLSA-JM given $|Z| = 32$ Right: PLSA-BS given $|Z| = 128$. The divergence is even more pronounced.

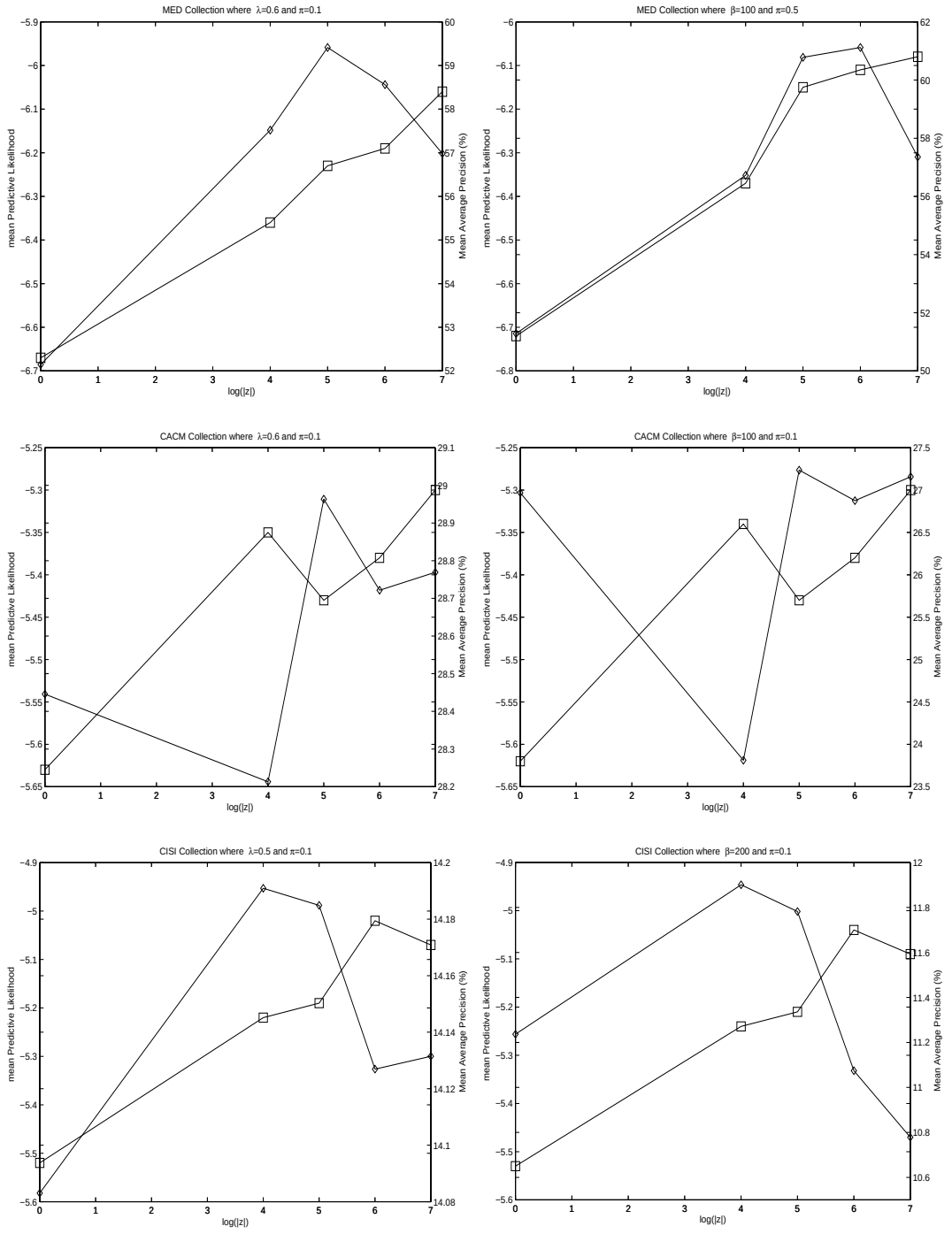

Figure 6.4: mPL vs mAP across the $|Z|$ space(shown in log scale). Top to Bottom: MED, CACM and CISI. Despite the increasing mPL, the mAP appears to reach a maximum point before decreasing.

we where able to find the parameter settings which gave significantly better retrieval performance. Success under such conditions is of limited utility in an operational environment because there is no way to identify these conditions, other than a brute force search. Whilst, it should be possible according to the assumptions of the Language Modeling framework, in practice this clearly is not the case, when using the PLSA context based document models. When we analyzed the change in mAP versus mPL over the range of latent variables, the selection of the best data model according to the mPL did not necessarily result in the selection of the best retrieval model. Again, using this criterion for identifying the number of latent variables did not provide a clear indication as to the model which will be optimal for retrieval performance. This evidence tends to suggest that Assumption Two does not appear to hold when we generated context based document models using PLSA on these collections.

6.1.3.1 Previous Work

In previous work, using PLSA[60, 61] for *ad hoc* retrieval, quite substantial and significant increases over the baseline (TF and TF.IDF using the VSM) performance was achieved on these collections. Indeed, this substantial increase in performance initially drew our attention to the possibility that using contextual evidence (i.e PLSA) *is* beneficial in the retrieval process. However, our results suggest that this form of contextual evidence *can* be beneficial in the retrieval process, but only under certain circumstances.

There are several key differences between our study and the former studies, these are:

- We have used PLSA within a wholly consistent language modeling framework and not within the Vector Space model.
- We did not employ any *ad hoc* weighting schemes such as idf, and
- We have reported the performance of the model which we could select through estimation, as opposed to selecting the model according to the one which gave the best retrieval performance, whether that be the parameter settings or PLSA

decompositions[4].

We believe the former two points while qualitatively different are not the main causes of the difference in performance. The LM approach is akin to the vector space approach in many respects, it makes similar assumptions (i.e that the similarity of a document and a query is correlated to relevance). They both match on query terms but essentially employ different weighting schemes. The VSM relies on *ad hoc* weightings such as idf to improve performance, whereas in the LM the weighting is implicit with in the document representation. The main difference here is that within the LM framework, under the assumptions, there is a clear objective as to how to maximise the retrieval performance, whereas in the VSM there is no such objective. This is an example of the benefits of performing such analysis within a principled framework. The latter point presents an obvious problem. Selecting the 'best' run according to the retrieval performance runs the risk that the PLSA model is tuned specifically to the set of queries. Given the sheer volume of parameters $|D| \times |Z| \times |T|$ then conceivably many local optima will exist within this parameter space, with respect to the retrieval performance. Some more generalizable to future queries than others. To illustrate, consider the following constructed example, we have taken the MED and CISI collection and taken the first half of the queries with which to used as a 'training' set. To find the 'best' settings we used this set of queries to evaluate the model's retrieval performance, select the parameters according to those that achieve the best retrieval performance. We then used the other half of the queries to validate the model's retrieval performance. Instead of a random initialization, we seeded the $p(z|d)$ matrix such that all the relevant documents associated with one of the initial queries in the first set were ten times more likely to be generated from a corresponding latent variable. (i.e. one latent variable for each query) The remaining documents were assigned such that they were ten times more likely to be generated from an addition latent factor. The $p(t|z)$ was randomly initialized, as before. Then the PLSA model was estimated according to the EM algorithm with Tempering. We selected the PLSA model and the parameters that gave the best retrieval performance for this initial set of queries, then

[4]In [4], we conducted a study which replicated this work we took the average of the retrieval performance over all the randomizations performed. Within the randomisations existed models that significantly out performed the baseline, but on average there was no significant difference.

computed the retrieval performance for the second set of queries. In Figure 6.5, the PR curve for the initial set of queries is denoted by boxes and the second set is denoted by diamonds. From these graphs it is clearly evident that the model is severely over fitted to the initial query set. Whilst this example has been quite contrived, it demonstrates the tailoring of the retrieval performance to the query set. Hence, selection of such models needs to be through a non-discriminative approach such as the mPL criterion. However, the mPL criterion failed to identify the optimal parameter settings, which questions the utility of applying PLSA to *ad hoc* retrieval (whether it be within the Language Modeling approach as done here, or within the VSM).

In summary, the context based document models which we derived from PLSA may provide significantly better retrieval performance when given the appropriate set of parameters. However, given the method of estimation and selection used, we were not able to consistently and reliably identify this set of parameters, such that significantly better retrieval performance was obtained. Furthermore, our analysis indicated that using PLSA based document models on these data collections that there was a break down in Assumption Two. Whist we were able to generate better representations, unification of the data and retrieval model did not transpire.

6.2 Topic Tracking Associations

In this section, we present a different form of semantic association as the context of the document based on the past interactions of the user. We employ Topic Tracking to define the semantic associations between documents, in a unsupervised manner given initial user input of the definition of a topic. Topic Tracking is essentially Information Filtering generalized to cater for multiple topics. For each topic, a number of initial example documents are provided as the definition of a topic that a user is interested in. Topic Tracking uses this information to decide whether the incoming (or remaining) documents are related to this topic or not[72]. Hence, the user is able to define their context through building topics. These topics represent areas of interest to the users and define their understanding of the documents in the collection. Unlike in the first

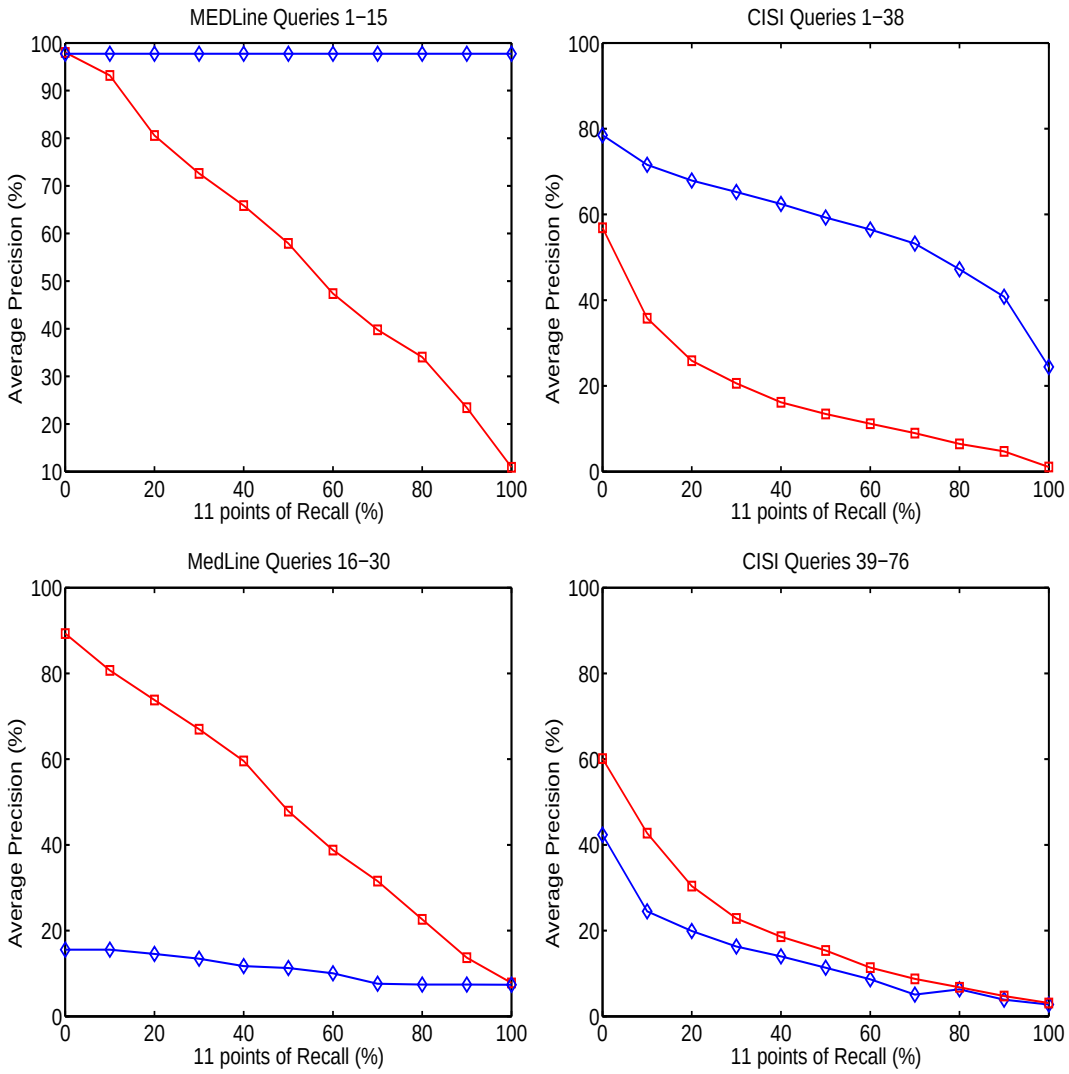

Figure 6.5: Top: The performance of PLSA models tuned on the initial set of queries shown by the diamonds and the performance of a baseline shown by the squares. Bottom: The performance of the PLSA and baselines models on the remaining queries. Left: MED Right: CISI. Notice the excellent performance by PLSA in the top graphs whilst in the bottom graphs their performance is very poor due to the over tuned PLSA model.

set of experiments where the context we used assumed that a document is composed of several topics, in this set of experiments we assume that a document may only belong to one topic. This assumption has made for two reasons; (1) to ease the computational burden such that it could be computed in a reasonable amount of time given the size of the collections, and; (2) to maintain a fair comparison between performance when using semantically defined topics against similarity based clusters.

The topic and cluster models that we employed in this section are defined as follows: The context parameters for topics are: $\Theta_{top} = \{p(t|k), p(k|d)\}$ and clusters: $\Theta_{clu} = \{p(t|g), p(g|d)\}$ where the k represents the topics and g represents the clusters.

The context for a particular document under the topic tracking semantic association is defined as:

$$p_d(t|\Theta_{top}) = \sum_k p(t|k)p(k|d) \tag{6.3}$$

where there exists a value of i such that $p(k = i|d) = 1$ and $p(k \neq i|d) = 0$. i.e. a document is drawn from one topic.

A context background model $p_d(t|\theta_{top})$ is constructed by substituting $p_d(t|\Theta_{top})$ into Equation 4.2 for $p_d(t|\Theta_X)$, which applies a proportion π of smoothing with the background collection model.

The context based document model is then generated by using the general form (Jelinek Mercer) and Bayes smoothed document models shown in Equation 4.3 and Equation 4.4, respectively. Where the former shall be referred to as TOP-JM and the latter TOP-BS. For the general form (Jelinek Mercer), the complete estimation of the document model is:

$$p(t|\theta_d^{top}) = (1-\lambda)\frac{n(t,d)}{n(d)} + \lambda\left((1-\pi)\left(\sum_k p(t|k)p(k|d)\right) + \pi p(t|\theta_C)\right) \tag{6.4}$$

In a similar fashion, the cluster based document model is generated using Θ_{clu}. The general form is referred to as CLU-JM and the Bayes smoothed approach CLU-BS, where the λ, π, $p(t|g)$ and $p(g|d)$. For the cluster based document model the parameters $p(t|g)$ and $p(g|d)$ are estimated according to a similar based clustering of the

collection as opposed to a semantic based association. This effectively defines the cluster based approach in [89] and serves another baseline for comparing the performance of the context based model (see Section 6.2.1). The following section details the actual process of topic assignment employed along with all other pertinent experimental details.

6.2.1 Experimental Settings

The empirical evaluation was performed on the Wall Street Journal (WSJ) and the Associated Press(AP) Collections. The details of these collections can be found in Section 5.3.1 and collection statistics in Table 5.1. The queries used were the titles of the TREC TOPICS 101-200 which were stemmed and stopped as was the collection. The baseline document models used were Jelinek Mercer (JM) Smoothing see Equation 3.25 and Bayes Smoothing (BS) see Equation 3.32. These were then compared against the context based document models (TOP-JM and TOP-BS) and against the cluster based document models (CLU-JM and CLU-BS). The number of topics and clusters used was one hundred, and another topic/cluster was added, such that $|K| = 101$ and $|G| = 101$. The additional topic/cluster was required to assign any documents that were not assigned to the first one hundred topics.

6.2.1.1 Context Estimation

As we have argued, we can impose the user's understanding of the collection through the previous interactions that the user has had with the collection. For this we use the TREC TOPICS 1-100 from the topic tracking task, to define such previous interaction with the collection. The set of relevant documents associated with each TREC TOPIC was used to form the definition of the topic and then the remainder of the collection was classified as being in a particular topic or not. We used the relevance modeling approach defined in Section 3.5.5 to score and classify documents. The procedure for constructing the context based on the topical associations was performed as follows:

- For each TREC TOPIC i a relevance model was construct for that topic from the

set of relevant documents associated with that topic. The relevance model for this topic $p(t|\theta_{R_i})$ was smoothed with background collection model at $\lambda = 0.5$. The non-relevance model was the background collection model (regardless of topic). If a document document appeared in several topics it was placed in the more recent topic.

- For each unassigned document d,
  - The document was scored against each of the topics i using the relevance models constructed from above, where the score was computed using the following formula (adapted from Equation 3.50):

$$O(d,R_i) = \prod_{t \in d} \frac{p(t|\theta_{R_i})^{n(t,d)}}{p(t|\theta_C)^{n(t,d)}} \tag{6.5}$$

  - The document was assigned to the topic i with the highest odds ratio $O(d,R_i)$. However, if $O(d,R_i)$ was less than one, i.e less likely to be generated from the topic i than the collection, the document was assigned to the topic 101.
  - The probability of a topic given a document is then defined $p(k|d)$, where $p(k=i|d) = 1$ when the document was assigned to i.
- The probability of a term given the topic k_i can then be estimated, such that: $p(t|k=i) = \frac{\sum_{d \in i} n(t,d)}{\sum_{t'} \sum_{d \in i} n(t',d)}$

On the other hand, we defined the cluster matrixes $p(t|c)$ and $p(c|d)$ using the trec4-kmeans-xu99 data set[5]. This data set contains clusters of TREC documents. The documents were clustered according to the cosine similarity metric with the k-means clustering algorithm[155]. This data set was previously used in a distributed retrieval setting, where the clusters represented different resources containing similar documents [155]. The cluster associations for the WSJ and AP collections were extracted from this data set and used to form the clusters used to build cluster based document models. Any document that was not assigned to a cluster from this data was classified in a process akin to that for the topics, this was required because in this data set only

[5]This data set is available on line from http://hartford.lti.cs.cmu.edu/callan/Data/.

the Associated Press documents for 1988 were clustered. Hence, the documents from 1989 needed to be assigned to a cluster. Similarly, for WSJ documents 1986-1989.

6.2.2 Results

Tables 6.4 and 6.5 contain the performance statistics of the baseline versus the topic based document models and cluster based document models on the WSJ and AP collections, respectively. Only the performance statistics giving the best estimated parameters are shown for comparison.

On the WSJ collection application of either the topic or cluster based document models failed to yield better retrieval performance in terms of mAP, p@10% and p@30 docs. However, the cluster based document models achieved significantly higher recall. The topic based models actually achieved a substantially larger total recall than both the baseline and the cluster based document models, however this was not significantly different. When using the topic based document model under Bayes Smoothing (TOP-BS) the mAP and p@10% were significantly worse, despite the large increase in total recall.

On the AP collection significantly better mAP was obtained using the topic models (TOP-JM) and (TOP-BS) over the respective baselines, while this was only the case for the cluster based document models when using Bayes Smoothing (CLU-BS). Again a substantial increase in recall was witnessed but this was not significantly different. Nor was the difference between the cluster based models and the topic based models.

In Figure 6.6, the graphs show the behavior of mAP and mPL as we manipulated the smoothing parameter for the topic and cluster based document models on the WSJ and AP collection, unlike the corresponding graphs for the baseline model BS shown in Figure 5.6 where there was a divergence between the data model and the retrieval performance. In these graphs using the context/cluster based document models the mAP and mPL are unified.

From the results reported in the above tables and graphs the performance of the topics and the cluster based document models appeared to be very similar. Significance

Model	Parameters	mPL	mAP	p@10%	p@30docs	Recall
JM	$\lambda = 0.556$	-6.9065	24.06	64.54	32.00	5420/8469
CLU-JM	$\pi = 0.1423$ $\lambda = 0.555$	-6.7949	23.85	61.18	31.53	5588*/8469
TOP-JM	$\pi = 0.139$ $\lambda = 0.55$	-6.7988	23.36	62.75	31.67	5618/8469
BS	$\beta = 371.6$	-6.9245	26.37	68.90	35.47	5420/8469
CLU-BS	$\pi = 0.1423$ $\beta = 251.41$	-6.8438	25.42	67.19	34.30	5505*/8469
TOP-BS	$\pi = 0.139$ $\beta = 257.16$	-6.8469	25.40*	69.19*	34.67	5572/8469

Table 6.4: The performance statistics for the CLU and TOP document models on WSJ collection.

Model	Parameters	mPL	mAP	p@10%	p@30docs	Recall
JM	$\lambda = 0.549$	-7.265	24.84	67.68	34.03	6329/9738
CLU-JM	$\pi = 0.1041$ $\lambda = 0.5489$	-7.3316	25.26	68.26*	33.57	6575/9738
TOP-JM	$\pi = 0.1319$ $\lambda = 0.5575$	-7.1156	25.49*	69.33*	33.4	6716/9738
BS	$\beta = 279.02$	-7.290	25.85	68.32	36.4	6319/9738
CLU-BS	$\pi = 0.1041$ $\beta = 276.86$	-7.3889	25.96*	67.35	35.17	6546/9738
TOP-BS	$\pi = 0.1319$ $\beta = 341.88$	-7.1346	26.43*	68.11	35.4	6682/9738

Table 6.5: The performance statistics for the CLU and TOP document models on AP collection.

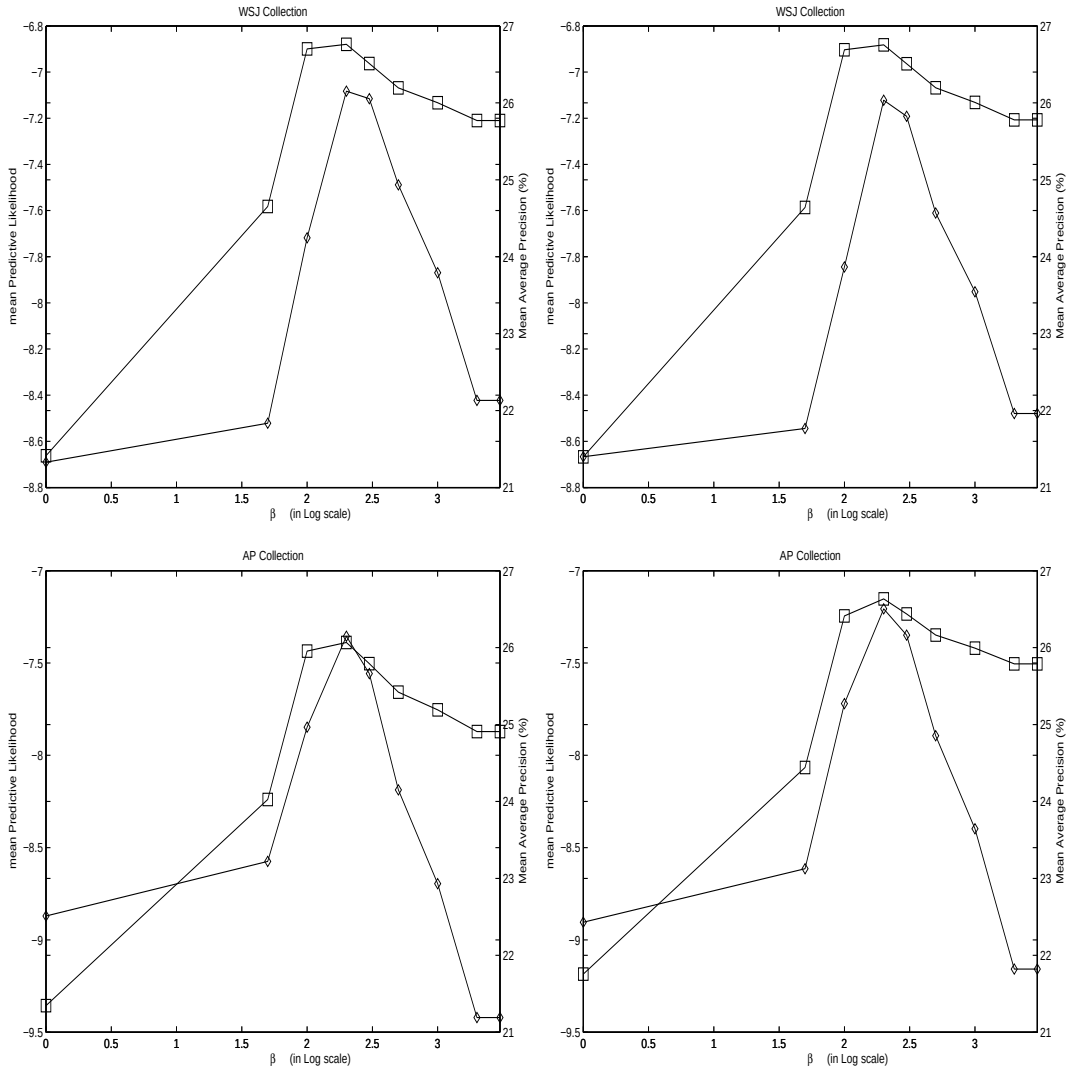

Figure 6.6: Change in mAp and mPL given the smoothing parameters for the cluster and context based document models. Top: WSJ Collection Bottom: AP Collection Left: CLU-BS Right: TOP-BS

testing revealed that these models were not significantly different from each other. We posited that the different types of evidence used (topics and clusters) would only be useful for certain queries. If the evidence contained within the topics and the clusters was similar (because of a similar mapping of documents to topics and document to clusters, or all the relevant documents for a particular query residing in one very distinguishable topic/cluster) then the benefits to a particular query would also be similar. On the other hand, if the mappings were quite disparate then the different forms of evidences may be useful for different queries. To discover whether this was case, or not, we plotted the difference in mAP between the topic and baseline model and the cluster and baseline model for each query (See Figure 6.7). From these graphs, we can see that when the topic based document model provides an increase in mAP, so too does the cluster based document model, and similarly for a decrease. We performed Pearson's correlation test[6] on this set of points and found that there was a significant correlation between the differences of each type of document model. The correlation coefficient r was around 0.7 and statistically significant for all collections at 95% confidence. So when the cluster model outperformed the baseline so too did the topic model, and when the cluster model did not outperform the baseline nor did the topic model. This suggests that the evidence used in either document model is quite similar in nature and effect. However, the topic based document model appears to show a greater variation, either obtaining a larger gain or larger loss in mAP over the baseline than the cluster based document models.

6.2.3 Discussion

In this study, we have considered both cluster based document models and topic based document models. Generally, the topic based document models tend to provide better retrieval performance than the cluster based document models. However, this difference was not significant. Against the baseline methods, improvements were only obtained on one of the collections (i.e. AP). The number of topics or clusters was limited

[6]We opted for Pearson's correlation test in this instance as the differences appeared to be normally distributed.

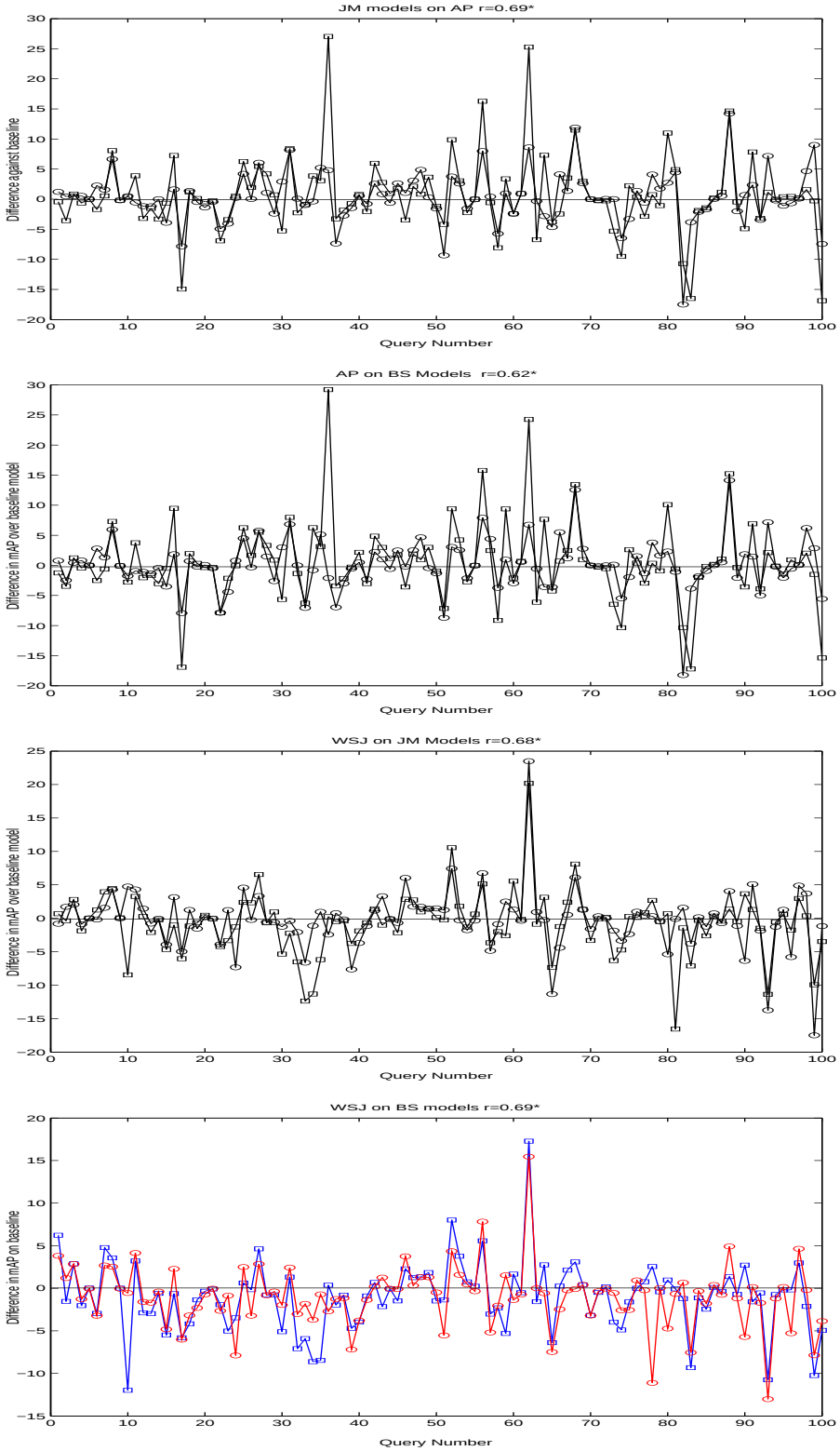

Figure 6.7: The difference in performance of the baseline against the topic and against the cluster models on WSJ and AP. The topics are shown by the circles and the clusters are shown by the squares. Notice the similar affect of the topics and clusters.

to one hundred plus one, because of the access to such clusters and a corresponding number of predefined topics. Had more or different forms of contextual associations been available we may have been able to consider the affects of varying the number of topics/clusters, as we did in the previous study by manipulating the number of latent variables.

The selection of the number of topics/clusters is an open problem because it is unknown as to how many topics represent the user's interests or how many documents sufficiently similar to each other define a cluster. We posit that an insufficient amount of topic-relevant document associations was available to define the context of the user's interactions within this collections. However, in an operational setting where access to query logs and interaction is available then it is possible to mine these logs to extract usage based associations. By selecting the most informative or interesting associations we could produce better context based document models. Further research could be directed in exploring this possibility.

The topics we used in our study to define the user's context may have been too specific to generalize to all users and further information needs. For instance some of the topics referred to particular government affairs or topical events. For example, 'the strategic defense initiative' or 'Black Monday the Stock Market Crash'. However, the context of these to the user may be somewhat different. For instance, the news web sites (and even news papers) organize content according to pre-existing categories (Current Affairs, Finance, Sport, Travel, etc). These categories represent paths of interaction that are so commonly associated that they represent a very strong semantic association. This association forms the context the user considers, so when thinking more generally about the two example topics, we would consider them to be in the 'military' and 'finance' sections respectively. In a related study[5], we defined domains according to such high level categories (domain) by grouping the topics according to their domain. We used them to define the domains and consequently the user's understanding of the collection at this higher level. i.e. documents about military, science, politics, and so forth. However, the results were generally no better than the baseline methods either (even when a search of the parameter space was undertaken to find the best retrieval performance).

In Chapter 4, we noted that the type of contextual evidence should be selected with respect to the query and the user's information need. From the difference plots in Figure 6.7, we noted that for some particular queries the context/cluster based document models were successful in surpassing the baseline models. Ideally, we would like to be able to select when the context/cluster based document models when they will be successful and when they will not, instead of just applying the model and assuming that it will work better in all circumstances. Contextual information retrieval requires that the IRSs tailor retrieval to the user and their information needs. Model selection is just one area where this could be achieved, where already research has shown that some improvements are possible[54].

A selection mechanism could be derived from the assumptions of Language Modeling. From A2, the goal is to obtain the best representation of the underlying data with respect to the user. This could be taken to imply that given different document models $\mathcal{M}$ (i.e. LP, JM, BS) which model is most likely to generate the query. Such that the probability of a query given $\mathcal{M}$ and the collection is determined by the sum over all documents in the query of the query likelihood (the average log query likelihood).

$$p(q|\mathcal{M},C) = \sum_{d} p(d) \prod_{t \in q} p(t|\theta_d, \mathcal{M}) \tag{6.6}$$

In this case, the question is, from which model $\mathcal{M}$ is the query most likely to be generated? The model $\mathcal{M}$ which maximizes the query likelihood is presumed to provide the better performance. The intuition is that if the representation is more likely to produce the query then the data models are more reflective of how the user views the collection. That is, they think these are good terms to use a query, and believe they are more probable in the desired documents. Whether such a criterion would work in practice remains undetermined, and is left to future research.

In work by Liu and Croft[89] they proposed and tested cluster based document models within the language modeling framework. Their, 'cluster based document model' are equivalent to our CLU-BS approach. In their study they found that the retrieval performance was sensitive to the clusters created. They used various types of clustering algo-

rithms, thresholds for such clustering algorithms, and smoothing parameters to empirically determine whether better retrieval performance *could* be obtained. They of course only report the combination which resulted in the best performance. Our approach differs in that we have automatically determined the smoothing parameters which resulted in the best data model. Even though they performed an exhaustive search of the parameter space, they only report small increases over the baseline method on the WSJ and AP 88-90 collections. Under our approach the optimal number of clusters would be set according to the number of clusters that maximized the mPL for the cluster based document models. Instead, they found through an extension search of the possible parameters that the best retrieval performance was attained when the number of clusters was set to 2000 for the AP and WSJ collection. It would have be interesting to try different sized clusters and topics to see the change in mPL. Unfortunately, we were not able to obtain access to the clustering software, nor clusters-document associations from their previous work. However, our results using just one hundred clusters shows that the cluster based document models can be estimated such that it gives comparable or significantly better mAP, whilst consistently returning more relevant documents.

In this section, we compared the baseline models against the topic based models which were defined according to the user's past interactions with the document collection. The use of these semantic associations provided significantly better retrieval performance on the AP collection, but this was not the case on the WSJ. Against the cluster based document models, the performance of the topic based document models was not significantly different. Both types of document models consistently retrieved more documents than the baseline models, but this was not always significantly different. These results provide some evidence towards the Context Hypothesis, and limited evidence towards the Cluster Hypothesis, respectively. The context/cluster based document models created better representations of the underlying data than the baseline models, however, this did not necessarily translate into (significantly) better retrieval performance. However, under the context or cluster based document models the A2 assumption appeared to be upheld, when examined over the change in smoothing parameter.

6.3 Web Link Associations

In this section, we use the links between web documents as the semantic association for defining the document's context. The user's understanding of the collection is partly expressed though the links they create, as these links represent a semantic association between two documents. The direction of these links will affect the meaning of the semantic association and how the document is understood and perceived. The two types of links we consider are out links and in links.

Out links are usually produced by a single user and represent that user's understanding of how this document relates to other documents within the collection. By using the out linked documents as context for a document d, we are assuming that d is about these out linked documents (in some way, or to some extent). Essentially, by generating the document model with out linked documents, we are creating a representation of a 'super' document. This super document is represented by itself and the context documents which are only one link away. This may not be entirely suitable for *ad hoc* IR, because we are after the references to the relevant documents, not references to references. However, it would seem more suited to the task of finding the best entry point, where the goal is to find documents which contain references to relevant documents.

In links are usually produced by many different users and so are probably more representative of the understanding amongst users of the collection. Each in linked document provides another way of referring to the document, and this contributes to the context of the document (i.e. the context in which the document is discussed, described and or referred). This could bridge the vocabulary effect, because the in linked documents may provide terms which are representative of document but not in that document. However, not all of the in linked document may actually be related, some parts of the in linked document maybe off topics. This will introduce some noise into the document model which could be detrimental to the performance of the system. Consequently, efforts of this nature, (i.e. using the in links to make a better representations of the documents), have usually focused on using the anchor text of the links, not the entire document[53, 111, 23].This makes sense, for the reason above, however

is not entirely consistent with the document modeling process. When anchor text is generated, the distribution of terms used changes significantly from normal text, with terms like 'home' and 'page' akin to stop words. Using the anchor text would need to be considered in some other way to remain consistent with the underlying approach. For this reason and to remain consistent in developing context based document models throughout the thesis, we use the in linked documents and not the anchor text.

The link based document models that we employ are defined as follows: The context parameters are: $\Theta_{out/in} = \{p(t|d'), p(d'|d)\}$, where $p(d'|d)$, the probability of a document d' given document d represents the distribution of links (in or out) associated with document d. The context for a particular document under the link based semantic association is defined as:

$$p_d(t|\Theta_{out/in}) = \sum_{d' \in D} p(t|d')p(d'|d) \tag{6.7}$$

where:

$$p(d'|d) = \frac{n(d',d)}{\sum_{d''} n(d'',d} \tag{6.8}$$

and $n(d',d)$ is the number of times d' links to d for in links, and is the number of times d' is linked to by d for out links. Potentially, other distributions could be developed to quantify the relationship between the documents more precisely, such as using PageRank[109] or pHITS[20].

The link based context background model $p_d(t|\theta_{out/in})$ is constructed by substituting $p_d(t|\Theta_{out/in})$ into Equation 4.2 for $p_d(t|\Theta_x)$, which applies a proportion π of smoothing with the background collection model. The link based context based document model is then generated by using the Jelinek Mercer and Bayes smoothed document models shown in Equation 4.3 and Equation 4.4, respectively. When using the in link associations the model shall be referred to as IN-JM and IN-BS, and for the models using out link associations, they shall be referred to as OUT-JM and IN-JM. For the general form (JM), the complete estimation of the document model is:

$$p(t|\theta_d^{out/in}) = (1-\lambda)\frac{n(t,d)}{n(d)} + \lambda\left((1-\pi)\left\{\sum_{d \in D} p(t|d')p(d'|d)\right\} + \pi p(t|\theta_C)\right) \tag{6.9}$$

Type	Total Links	Docs with Links	Avg Links	Max Links
IN	1041262	186912/247491	4.21	3201
OUT	1041262	212085/247491	4.21	1283

Table 6.6: Link statistics

6.3.1 Experimental Settings

The empirical evaluation was performed on Web Track 2 Gigabyte collection (WT2g) and the titles of the TREC TOPICS 401-450 were used as queries (see Table 5.1 for collection details) The baseline document models used were Jelinek Mercer (JM) Smoothing see Equation 3.25 and Bayes Smoothing (BS) see Equation 3.32. These were then compared against the context based document models which used in links (IN-JM and IN-BS) and out links (OUT-JM and OUT-BS).

The links for each document were extracted from the collection and used to define the context based models. The link statistics are shown in Table 6.6 where on average a document had 4.21 links. However, there were many documents without links. In these cases, the context background model was equivalent to the collection background model. For linked documents, the context based models were estimated as outlined in Section 4.3.4 where the parameter values that maximized the in and out links context based documents were $\pi = 0.353$ and $\pi = 0.372$, respectively.

6.3.2 Results

In Table 6.7 the performance statistics of the baselines (LP, JM and BS) versus the in link and out link context models is reported. The performance of the best data model and best retrieval model is reported. Note, however, that for the BS linked based models the best data model and retrieval model were one and the same. Consequently, we only report this once in the table. As before, the statistical significance of results is denoted by an asterisk and was obtained using the Sign Rank test at 5% significance level.

Model	Parameters	mPL	mAP	p@10%	p@30 Docs	Recall
LP	$\alpha = 0.0002$	-9.5431	14.62	59.13	20.13	1361/2279
JM	$\lambda = 0.489$	-7.286	21.07	63.50	25.27	1789/2279
OUT-JM	$\lambda = 0.557$ $\pi = 0.372$	-7.1508	15.05*	54.46*	18.87*	1644*/2279
OUT-JM	$\lambda = 0.1$ $\pi = 0.372$	-7.2508	19.29	63.32	21.60*	1650/2279
IN-JM	$\lambda = 0.552$ $\pi = 0.353$	-7.1440	16.46*	58.39*	20.27*	1658*/2279
IN-JM	$\lambda = 0.1$ $\pi = 0.353$	-7.3965	19.41*	64.14*	21.93*	1615*/2279
BS	$\beta = 219$	-7.3111	20.70	68.84	25.47	1709/2279
OUT-BS	$\beta = 258$ $\pi = 0.372$	-7.2167	17.86	62.26	21.06*	1708/2279
IN-BS	$\beta = 323$ $\pi = 0.353$	-7.2747	17.49*	60.52*	21.80*	1650/2279

Table 6.7: The results for using context based smoothing on using in links and out links as the context.

From Table 6.7, we can see that most of the results from the link based document models performed significantly worse than the baseline models (JM and BS). However, the link based document models still provided a better representation of the underlying data.

6.3.3 Discussion

From the results reported it is clear that using the in and out links as context for a document does not provide any benefit to retrieval performance, despite the fact that better document representations were generated. In these graphs the best retrieval performance was obtained when λ was 0.1. This is quite a small amount of smoothing (comparatively), where the main contribution to the query likelihood was from the

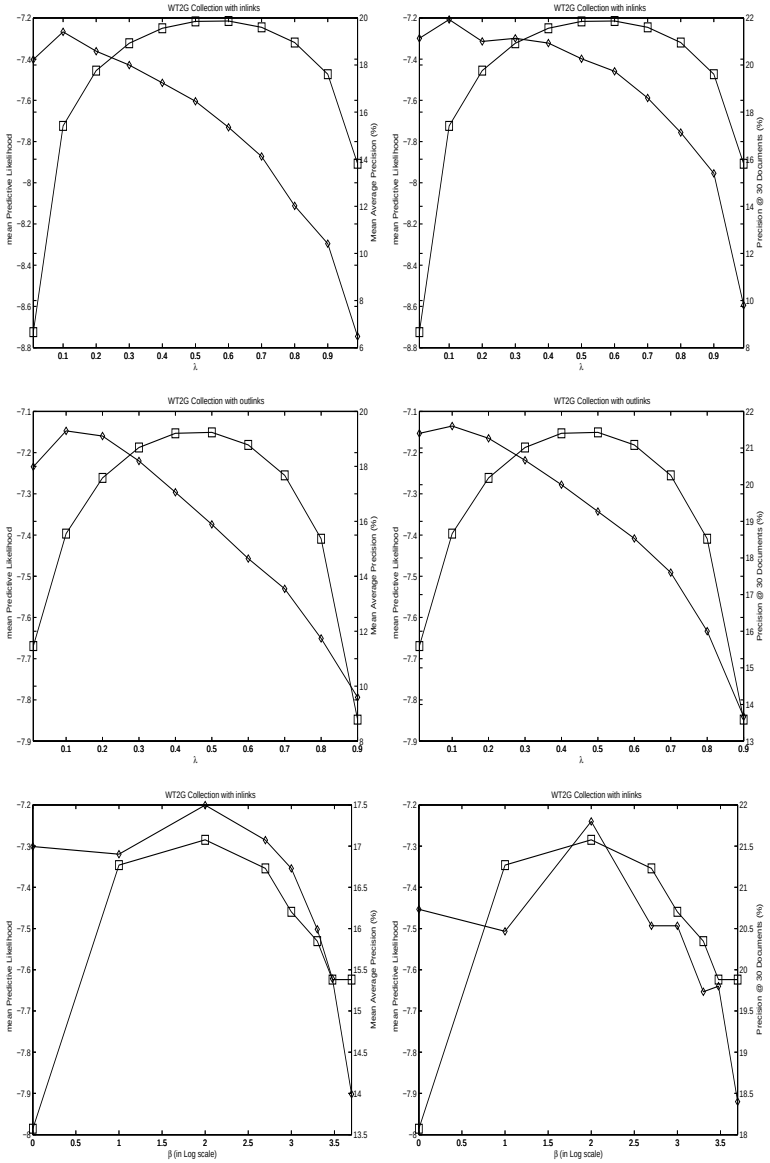

Figure 6.8: WT2g. Top: IN-JM Middle: OUT-JM Bottom: IN-BS Left: mPL vs mAP Right: mPL vs p@30docs. Notice how the best performance for the JM models perform the best when the least amount of context is used (i.e. $\lambda = 0.1$)

empirical term probabilities. Interestingly, the best performance of the IN-JM and OUT-JM was when the λ was set to 0.1. This small λ value means that the scoring was relying mainly on the empirical term probabilities within the document $p(t|d)$ suggesting that the context was very poor for retrieval. We suspect this disappointing performance was due to a few factors:

- The link sparsity meant that the context background models were only built with approximately 4-5 documents. Also, the link structure in the document collection has been criticized for being a poor representation of the web[45].
- The linked documents were probably too coarse adding more noise than information. A more appropiate solution would use the anchor text instead of the entire document. However, instead of trying to embed the anchor text directly into the document model it would be more appropiate to use it as external evidence, such that the document provides evidence of relevance (i.e $p(q|d)$) and the anchor text provides evidence of relevance (i.e. $p(q|a)$, the probability of a query given the anchor text for the document d). Further, the generation of the anchor text a would be conditioned on the document d, because the author of the anchor text will presumably have read document d, and then linked to the document by generating an anchor (i.e. link and text).
- The *ad hoc* task was inappropriate for the contexts used. As we previously mentioned, the out links would provide a good source of contextual evidence when searching for the best entry document (though we were unable to the test whether this was the case, because the WT2g collection did not have such queries.)

Further testing is required to determine whether these are the actual causes or whether there are some other reasons for this failure in performance. Nonetheless, this shows that despite being unable to build a better representation of the document models, that this will not neccessarily translate into better retrieval performance. Perhaps, there is too much bias introduced when we are applying other context based document models, and this needs to be addressed. Alternatively, maybe by focusing solely on generating a better representaiton of the underlying data we have not considered other factors which will affect the performance.

6.4 Chapter Conclusions

This section provides an overview of the results found in general across each of the different studies that we performed on evaluating the context based document models. The overall findings suggest:

- We have shown that different forms of contextual evidence can be used to benefit the retrieval performance, and hence some support towards the Context Hypothesis, while performance increases can be obtained when employing context based document models. However, selecting the parameters that obtain such increases is not always guaranteed by the mPL criterion.
- The context based document models always offered a better representation of the underlying data than the baseline models. With regards to A2, the PLSA models were variable in achieving unification. The topic and cluster based document models displayed unifications between the data model and the retrieval performance. Whilst the LINK based models were unified under (IN/OUT)-BS, but there was a complete mismatch under (IN/OUT)-JM. In both cases, though the context severely degraded retrieval performance. Overall, even though the context document models achieved a better representation this did not necessarily translate into better IR performance.
- Through query analysis, we found there are often particular queries when the type of document model used is more beneficial for retrieval performance than others. Some form of selection would be required to exploit these benefits. This provides an interesting avenue for future work, the prediction of the representation based on the query.

6.4.1 Summary

In this chapter we attempted to create better representations of the document models with respect to the second assumption of Language Modeling. We did this on three different types of collections and using different methods for building the context asso-

ciated with the documents. While it was possible to increase the retrieval performance, this was subject to accurate parameter selection.

Chapter 7

Discussion

In this chapter, we present an overview of the different facets of the research undertaken in this thesis and discuss the implications of this work. We consider the assumptions of the Language Modeling approach and our efforts in attempting to capitalize on these assumptions. In particular we offer our interpretation of the query likelihood to aid in the explanation of the mismatch between the data model and retrieval model. This motivates a different two stage approach. Also, we critically re-examine our use of context on the document modeling side, and consider other ways of incorporating context from the user side. Finally, we propose a context based retrieval model which uses the context of the user to bias the ranking in an integrated Language Modeling Approach.

7.1 The Assumptions of Language Modeling

The Assumptions of Language Modeling were studied in Chapter 5 to deepen our understanding of the retrieval approach. In this section we discuss our general findings as to the validity of these assumptions and their implications.

7.1.1 A1 Correlation

The correlation between the probability of a query given a document and the probability of a document being relevant varied according to the point at which the correlation was measured. We found that as we increased the size of the document cut off, the correlation between the two measures increased. As we previously discussed, this was because of the influx of numbers of non-relevant documents. These non relevant documents had low query likelihoods and low probability of relevance which skewed the results of the correlation. We argued that measuring the correlation at a cut off of thirty documents was the most appropriate point, because that is often the only part of the ranked list examined by a typical user. At this point, the correlation between the two measures was relatively weak and held for about 20% of queries.

Is the query likelihood measure really proportional to the document's relevance? Perhaps, the query likelihood is better correlated to the document's similarity to the query? If so, then when the query is made more like the relevant documents then the correlation will be stronger, because for a document to be relevant it must be similar to the query. This is only the case when *relevance* is considered as *similarity*. This would then be a condition or restriction of the Language Modeling approach. It would be interesting to examine how well a similarity based model (such as the VSM) holds up with respect to A1, and whether there was a greater/stronger correlation between the VSM and the Language Modeling approach. Further experimentation found that there was a dependency between A1 and A3. When queries were formulated according to A3 then the proportion of queries where A1 was upheld increased. We discuss this further in A3 below.

7.1.2 A2 Unification

The unification of the data model and the retrieval model was only ascertained for particular types of document models. The JM, CLU-BS, TOP-BS, CLU-JM and TOP-JM all exhibited the expected behavior, such that the performance of the BRM was not significantly different from the performance of the BDMs. These types of document

models would appear to be good candidate document models to apply as they are consistent with the second assumption (i.e unified). However, the other types of document models were not as well behaved, performing erratically and sensitive to parameter change. Hence, maximizing the mPL for the LP, BS, PLSA-JM and PLSA-BS document model would typically result in sub optimal retrieval performance. This gap in retrieval performance represents the mismatch between the assumption and reality.

When we considered improving the representations of the documents through context based document models, we posited that improved retrieval performance should be obtained, because we will have developed a better data model. Whilst the context based document models that we applied provided a significantly better representation, a corresponding increase in performance was not always obtained.

This calls into question whether obtaining a better statistical representation will actually benefit retrieval performance. Perhaps, the model is taking a much too simplistic view of the retrieval process and not considering other factors which are known to affect retrieval (such as document length normalization). This is because the Language Modeling approach is limited to representing the relevance of a document through the document models. It is only through changes to the document model, that the retrieval function is changed. While this property is appealing, it may be more appropriate to consider the representation separately from the retrieval function.

This mismatch could have been due to factors; such as the query term importance or document length normalization. In Section 5.4.2, we explored the possibility that the query variability contributed to the gap between the BDM and BRM. However, when we applied the second stage of smoothing to compensate for the query terms, the performance of both the BDM and BRM increased. This seemed to indicate that the gap was not a result of the query variation but of some other factor(s). Later in Section 7.1.5 we provide a possible explanation of the phenomena which implicates document length normalization as the cause.

Alternatively, if we consider the magnitude of the change in mPL obtained by the context based document models over the BS and JM models, then the difference is really marginal about 0.1-0.3 in terms of mPL (See Tables 6.1, 6.2 and 6.3). On the

other hand, if we consider the difference between the BS and JM models versus the LP model shown in Table 5.7, then the difference was approximately 1-2, (an order of magnitude higher). The difference in mPL between the BS/JM models over the LP models translates into a substantial difference in mAP of around 5%-9%. If we were to extrapolate this to estimate the performance of the context models performance over the BS/JM models then we would only expect a very small change. Indeed, this is what we witnessed in most cases. We would then have to increase the mPL substantially more to secure a larger increase in mAP. However, there is a limit to how good the representation will be, as there is an upper bound on the mPL. This solution usually does not generalize well because it is the maximum likelihood estimate, which suffers from the ZPP. Hence it is necessary to rely on smoothing to obtain an estimate of the document model. The quality of these representations are limited to the amount of data which is available to estimate the model parameters and improving the quality through context does not provide any substantial improvements.

7.1.3 A3 Discrimination

Whether the query terms used were able to discriminate between relevant and non-relevant documents sufficiently was shown to be affected by type of query issued. We examined three types of queries; (Q0) standard queries consisting of the title of the topic; (Q1) ideal queries consisting of common terms from the relevant documents and (Q2) ideal queries consisting of highly discriminative terms. Whilst the standard queries issued managed to identify a fair proportion of relevant documents at early levels of recall, the ideal queries obtained a much higher proportion of relevant documents for each query. The difference in ideal queries was pronounced; the first was able to identify substantially more relevant documents, while the second identified with high accuracy a smaller subset of relevant documents. In other words the first type of query provided a recall oriented querying strategy and the second type provided a precision oriented strategy. This is actually rather intuitive because we would expect that the more general terms would identify a wider range of documents, whereas more discriminative (and less frequent) terms would identify very specific documents. This

outcome could be useful when considering pseudo relevance feedback. If the query terms are general then pseudo relevance feedback may not be particularly useful. If the query terms were very specific, then performing pseudo relevance feedback would, we anticipate, imporve retrieval performance. The terms used for query expansion can be selected such that a precision or recall oriented search is undertaken. From a usage point of view, a user can submit these kinds of query terms so that their search will be geared towards precision or recall.

The third assumption requires that (1) the user understands how they should query the IRS, and; (2) that they can successfully apply this knowledge when formulating and submitting queries. An open research question (thrown up by this assumption) is, what impact does querying have on the final accuracy of the search results if these conditions are met? For example, will educating the user on how to query an IRS based on the Language Modeling approach affect the quality of their search results? From our results, we would expect that retrieval performance would improve, if users could execute ideal queries. However, these were of course generated with respect to the actual set of relevant documents. So, would the queries of a 'trained' user be significantly different to those of an 'untrained' user? And, this would translate into queries which would be more discriminative of relevant and non relevant documents, contain more information to discern relevant from non relevant, and ultimately deliver superior retrieval performance? It is imperative that any tool be used both efficiently and effectively. An IRS is no different, hence the user must have a reasonable understanding of the basic intuition of the system such that it can be used efficiently and effectively. From A3, there are clear guidelines as to the kind of terms (i.e Q1 or Q2) that a user should pose to the system as a query. If the query is not consistent with A3 then the user only has themselves to blame for the poor search results. If a user/system could recognize this problem, then switching to an alternative approach such as the Translation Language Model[11] (described in Section 3.5.1) could be beneficial if, for instance, the poor query was due to the vocabulary effect. It would be insightful to see whether the Translation Model could address this and whether this would also improve the correlation of A1 and discrimination of A3.

Harper *et al.*[48] consider the 'ideal' queries from a different perspective. Given the

context of the user, as either unfamiliar or familiar with the topic of the search, they hypothesized that:

> Users unfamiliar with a topic will prefer documents in which **highly representative** terms occur, and users familiar with a topic will prefer documents in which **highly discriminating** terms occur.

In their experiments, the terms extracted for the particular user context were used to re-rank the top 1000 documents. Queries consisting of highly representative terms and queries consisting of the highly discriminative terms, correspond to the ideal queries Q1 and Q2 respectively. Their results indicated that the former queries did not yield better results, whilst the latter gave significantly better retrieval (in terms of R-Precision[1]). They suggest that in the latter case, initial queries by familiar users obtained better initial results from which to improve the re-ranking.

However, our study seems to suggest different reasons for this behavior. In the first instance, where Q1 queries were issued this will be of little benefit as they tend to improve recall, and this is not possible when re-ranking the top 1000 (i.e. we can not find any more relevant documents, only change their ranks to increase precision). In the second instance, the discriminative query (Q2) approach dramatically improves the ranking at early levels of precision. This accounts for the improvements in R-Precision witnessed, as opposed to actually receiving better initial queries by familiar users. This raises some interesting issues. How can we measure the quality of a query? With respect to the 'ideal' queries? How does this affect the evaluation? And further, is it possible to create queries that are 'perfect' queries, ideal queries which provide optimal retrieval performance? Whilst these are interesting questions, they are beyond the scope of the thesis and is left for further work.

[1] R-precision is the precision at R where R is the number of relevant documents in the collection for the query. An R-precision of 1.0 is equivalent to perfect relevance ranking and perfect recall. However, a typical value of R-precision which is far below 1.0, does not indicate the actual value of recall (since some of the relevant documents may be present in the hitlist beyond point R).

7.1.4 Assumptions of Retrieval Models

It is worth noting that the assumptions of Language Modeling (in part) are also applicable to other retrieval models. Consider the VSM, we could recast assumptions A1 and A3 as:

VA1 The similarity of a document and a query is correlated with the document being relevant, and

VA3 The user must select terms so that the query would be similar to the relevant documents.

For VA1 to hold, it is reasonable to expect that the query terms used need to be contained within the relevant documents (i.e VA3). However, there is no corresponding assumption for A2 because under the VSM the data model and retrieval model are separate. For instance, we could use the document models from LM with some similarity metric to define a specific VSM.

Each retrieval model makes its own assumptions about the retrieval process (implicitly, or otherwise). Explicitly stating these assumptions may prove useful in identifying other strengths/weaknesses of the model and provide a better understanding of the process to extend and develop the model. For example, under the Language Modeling Framework the user is a direct participant responsible for meeting certain criteria. A further research question this motivates is - how do we go about supporting the user in upholding or meeting such criteria? According to A2.1, the user needs to form an understanding of the distribution of terms within the documents in the collection. Providing this information to the user in some form or another should enable them to formulate queries of better quality.

7.1.5 Interpretation of Smoothing

In Section 3.2.1, we posited that the query likelihood does not distinguish between its contribution to the relevance of a document or its contribution to the non relevance of a document. Under this interpretation of the query likelihood, the probability of a term

being generated from a document model marginalizes relevance in the process, and is expressed in Equation 3.12 (shown below):

$$p(t|\theta_d) = p(t|R,d)p(R|d) + p(t|N,d)p(N|d)$$

If we consider each of the components, we can express the standard Language Modeling approach by defining the components as follows:

1. The $p(t|N,D)$ is approximated by the probability of a term occurring at random in the collection, (i.e. $p(t|\theta_C)$),
2. The $p(t|R,d)$ is approximated by the probability of a query term occurring in the document d, (i.e. the maximum likelihood estimate, $p(t|d)$), and
3. The priors are substituted, where $p(R|d) = 1-\lambda$ and $p(N|d) = \lambda$

Then, we obtain the standard document model, expressed in Equation 3.28, shown below:

$$p(t|\theta_d) = p(t|d)(1-\lambda) + p(t|\theta_C)\lambda$$

The λ parameter represents the amount of smoothing that a document receives and this affects the data model/retrieval model. By optimizing the data model, we can arrive at a value for λ which provides the best fit of the underlying data. Empirically, we found that this was approximately 0.5. However, under the analogy described above, the interpretation of λ and $1-\lambda$ is somewhat different. They are considered to be the prior probability of relevance given the document. Now, without any evidence to suggest otherwise, the probability of relevance given a document is likely to go either way. Similar to flipping an unbiased coin, the probability of each alternative is assumed equal. Therefore, the most sensible estimate for λ would be 0.5. This is an interesting observation, that the estimate of λ according to the mPL was more or less equal to the estimate of λ according to our analogy. Remember that the standard Language Model (i.e JM) was the most consistent in obtaining unification.

If we continue the analogy to consider Bayes Smoothing instead of Jelinek Mercer smoothing, then λ is assigned the value $\frac{\beta}{n(d)+\beta}$. Instead of a fixed prior for each document, the prior using Bayes Smoothing is proportional to the length of the document.

The motivation from a smoothing point of view was that a longer document provides a better sample from which to estimate the document model than short documents, and therefore require less smoothing. On the other hand, short documents are poorer representations and require more smoothing. The analogy provides a different explanation - the probability of relevance is higher for a longer document than a shorter document. That is, we are now accounting for more than just the representation of the data but also by the size of the document. This additional factor may be the cause of the mismatch between the data model and retrieval model under Bayes Smoothing.

7.1.6 Study Limitations and Caveats

The extent to which we can make generalizations from these findings are conditioned on accepting the following; (1) the expression and interpretation of the assumptions of Language Modeling; (2) the statement of the assumptions into empirically based hypotheses, and; (3) the analysis and measurement of each hypothesis. We raise some of the points for each of the assumptions and mention any other related facets.

A1 We measured a document's relevance through the Odds ratio because this is precisely the definition put forth by Lafferty and Zhai[78]. They argued the query likelihood is proportional to the Odds ratio. Nonetheless, document relevance may be quantified in other ways such as those discussed in Section 5.3.3, though we believe our interpretation is sound. However, the method with which we used to measure the Odds ratio, the Relevance Model, could have been different. Instead, we could have employed the BIM, but for consistency we choose the Relevance Model (being a related generative approach). This invites the question, whether the correlation would be as strong under the BIM model or not. Further and more generally, how do other retrieval models correlate with relevance.

A2 One of the main advantages voiced by Ponte[112] and Ponte and Croft[113] is the notion of unification of the retrieval function and data model. We have taken this notion to imply that we can infer a change in behavior of one (the retrieval model), given the behavior of the other (the data model). Such that, if we obtain a better rep-

resentation then we should be able to achieve better retrieval performance. However, maybe this was meant just as an observation of the model, such that the change in one, will directly influence the other, without stating what that change may be. Though, Ponte[112] (p145-145) states that effective retrieval can be improved upon to the extent which the data models are an accurate representation of the data, and that the user both understands the retrieval approach and have some sense of the distribution of terms in documents (i.e. A2 and A3). Hence, we believe the former is implied, implicitly.

A3 Quantifying sufficient discrimination is a contentious point, because what does sufficient discrimination actually mean? We have approached the problem by considering different thresholds which represent the users tolerance to non-relevant documents. However, this still considers the problem at an aggregated level and not at the query level, as for different queries we will have a different tolerance for the irrelevant. Our interpretation was that a user would be happy if half the relevant documents that contained those query terms were returned, then they would have been satisfied. This would represent sufficient discrimination for that user given that request. This implies that if the user submits a very specific query then they will not expect all the possible relevant documents but they expect to see at least half the relevant documents which match that query.

In our study, we simulated the user by generating 'ideal' queries on their behalf and considering the effect. One limitation of this study is not having evidence to confirm or deny whether users can execute the kind of queries that are suggested by A3.

Other We have not considered the external changes such as the effects of stemming and stopping on the model parameter settings. How does the application (or not) of these transformations change the nature of the Language Models? That is will the parameter estimates being different and how will the mAP be affected? Further, how well do other retrieval models satisfy the correlation of A1? For instance does the Binary Independence Model or the Vector Space Model produce stronger correlations with a document's relevance? Presumably, the BIM would have the strongest or highest proportion of correlations, whilst the VSM would probably be on par with the LM

approach. However, without a point of reference or baseline the generality is limited to Language Models.

7.2 Document Model Observations

So far we have been speaking in broad terms about the implications of our findings on the validity of the assumptions. Here we present an overview of our observations for the different document models that we employed through the course of this thesis and comment on any interesting or useful behavior.

7.2.1 Standard Document Models

We summarize our analysis of each of the document models, Laplace, Jelinek Mercer and Bayes Smoothing below.

LP The Laplace smoothed document models produced the poorest representations of the underlying data out of all the document models tested. This is because of the naive assumption made about the prior distribution, i.e that all terms have an equal probability of occurring. The LP document models only really overcomes the ZPP, as opposed to actually generating particularly good representations of the documents. Consequently, the retrieval performance for the LP document models were the worst among those assessed. However, the best retrieval results using LP were found when α was a very small and in the range, $0.01 \leq \alpha \leq 0.0001$.

JM The Jelinek Mercer smoothed document models provided a substantial improvement to the document model quality over the LP document models. This improvement comes from the use of an informed prior (the background collection model). The best representations of the underlying data were obtained when λ was set to approximately 0.55. At this parameter setting, the retrieval performance was optimal or near optimal for the collections we used. That is, we obtained unification under the JM document models. This is very close to the λ values suggested by previous research and also

to our suggested value under our interpretation of the query likelihood (see Section 7.1.5 above) where the value of λ should be set to 0.5. We shall refer to the former estimated Jelinek Mercer solution as JMe and the latter as JM50. Essentially, JMe and JM50 were approximately the same. The difference resides in how they were obtained. Previously, JM50 was originally suggested after empirical studies showed that the retrieval performance was maximized at or around this value. Here, we have approached the query likelihood differently and ascertained JM50 from assigning equal likelihood to the relevance of a document. Then from the assumptions of the LM approach, we have derived JMe by using a technique for estimation that is consistent with the assumptions. This provided the best possible choice of parameter setting according to the data (and data model), without recourse to queries and relevance judgments.

BS Interestingly, the quality of the Bayes Smoothed document models were usually slightly poorer than the Jelinek Mercer models. However, this difference was very small as previously noted. When estimating the BS document models the best representation tended to obtain reasonable retrieval performance. However it was not until the data model was over fitted that the best retrieval performance was actually obtained. We posited earlier, that this was because the prior probability is proportional to the document length (i.e. $p(R|d) \propto n(d)$). Under this interpretation, where we are required to set the prior probability of relevance for a document, then in the absence of any *a priori* knowledge, we would like to set the β parameter such that the *average* probability of relevance given a document is equal to 0.5. To achieve this we would set β equal to the average document length (mean or median) $\hat{n}(d)$. Consequently, documents greater than $\hat{n}(d)$ would be more likely to be relevant *a priori*, and conversely documents shorter than $\hat{n}(d)$ would be less likely. In Table 7.1, we show the average document length, the estimated β parameter from the $B\hat{D}M$, and the β parameter from the BRM. Note that the estimated β is reasonably close to the average document length.

This is an interesting observation, and one that has already been derived from a different perspective. In [33], Fang *et al.* conducted a formal study on the heuristics of retrieval algorithms. They applied constraints based analysis on the properties term weighting function should exhibit. For Bayes Smoothing, they prescribed a lower bound on the β parameter of $\hat{n}(d)$. Our study would seem to indicate that this is the

	$\hat{n}(d)$	$B\hat{D}M$ β	BRM β
MED	83	103.38	300
CACM	91	82.03	1000
CISI	230	228.54	3000
AP	243	279.02	2000
WSJ	247	371.6	2000
WT2g	218	219.55	5000

Table 7.1: The Average Document Length versus the estimated β. Notice the parameter estimated parameter value is reasonably close to the average Document Length. In the case of the WT2g collection, the distribution of document lengths was very skewed, instead we present the median.

case according to our interpretation of the query likelihood and based on our empirical findings.

Essentially, we can re-express the β parameter such that it is composed of two parts, the average document length and a constant ϕ such that:

$$\beta = \hat{n}(d) + \phi \tag{7.1}$$

The $\hat{n}(d)$ represents the portion of smoothing that needs to be applied in order to obtain a reasonably good fit to the data, whilst the ϕ represents the normalization component and could be estimated by drawing upon research in document length normalization.

7.2.2 Context Based Document Models

Our work on the assumptions of Language Modeling suggested two major directions for improving performance of the approach, either by improving the query through A3, or improving the document model through A2. We chose to investigate the latter, with respect to the Context Hypothesis. Hence, we explored the notion of using the semantic associations (which define the context) within the modeling process in order to generate better document models. Thus, we examined whether better representations were obtained, and whether these resulted in better retrieval performance

(A2). We found that we were able to create context based document models that gave a better representation of the underlying generative process; further, that these models could obtain significantly better retrieval performance. However, we could not always estimate the model parameters to obtain such performance. This is a limitation which needs to be addressed in order to use the context in such a manner.

As we previously mentioned, the increase in mPL of the context based models over the standard models (BS and JM) was relatively small. It is an open question as to whether this difference will have a significant impact on the retrieval performance. In terms of mAP the performance of either type of model is not particularly different. A substantially larger increase in the mPL is required based on the differences between BS/JM and LP in mPL/mAP. However, there is a limit to the predictive likelihood and we suspect that it is probably quite close to estimates obtained under the standard and context based document models. Hence, the representations we have obtained are probably as good as they can be[2]. This further suggests that the standard document models are probably very good representations already, so there is probably not much point trying to make them any better. Stated differently, the $p(t|d)$ is already a reasonably good approximation for $p(t|d,R)$.

7.2.3 Model Limits

Similar limitations have been exposed within speech recognition[123], where decreases in perplexity which is proportional to the predictive likelihood, have not translated into better speech recognition performance (i.e. lower word error rate). Is this a problem with the generative probabilistic Modeling approaches in general? The traditional probabilistic model used discrimination to discern whether a document is relevant given a set of features (query terms). Under the Language Modeling approach, the likelihood of a query being generated is used to rank documents, without regard from whence it came (i.e. from a non-relevant or relevant document). The presumption

[2]The maximum predictive likelihood of the model is when the document model tends to the maximum likelihood estimate, reducing the error between the empirical term probabilities and the estimated term probabilities. Unfortunately, we did not compute the maximum predictive likelihood during the course of our experiments, so we are unable to definitively say.

is that the query likelihood will sufficiently discriminate relevant from non relevant, but this does not guarantee optimal ranking under the PRP. The analogy of the generative model is appealing but fundamentally it is limited by how well the predictive likelihood indicates the performance of the model. Lately, there has been a move toward discriminative approaches[103], such as Support Vector Machines[151]. However, such approaches still suffer in that training data (queries and relevance judgements) are required to determine the best model parameters. Nonetheless, despite any possible problems with the Language Modeling approach it still represents an elegant model of retrieval, which offers a principled solution to parameter estimation. Whilst, this is of limited utility, we must remember that the LM approach is only a model of the retrieval process and every model has its limitations. Some of those limitations for the LM approach for *ad hoc* retrieval have been exposed and discussed during the course of this thesis.

In our case there becomes a point beyond which the quality of the data model is no longer indicative of the performance of the retrieval model. As we have already mentioned this is due to other factors influencing the retrieval performance.

7.2.4 The Two Stage Model: Reconsidered

The two stage model was originally developed from the empirical results motivating that the document and the query need to be modelled separately[163]. Recall, that the first stage was to obtain a better representation of the documents, where they suggested Bayes Smoothing. The second stage was to account for the query, where Jelinek Mercer smoothing was suggested.

However, from our findings on the behavior of the different document model and our interpretation of the query likelihood motivates a different approach for two stage smoothing. Under the decomposition of the query likelihood in Equation 3.12 the $p(t|R,d)$ can be approximated by the $p(t|\theta_d)$, where we recommend JM50/JMe because it will provide the best representation.

$$p(t|R,d) \approx (1-\lambda)p(t|d) + \lambda p(t|\theta_C) \tag{7.2}$$

The second stage of smoothing is introduced when we have to account for the probability of a term given a document and non relevance. This is approximated by the non-relevance model (i.e the probability of the term coming from a non relevant document) and is set to equal the background collection model $p(t|\theta_C)$. Now, we need to assign the prior probability of relevance given a document, which we make proportional to the document length (i.e. Bayes Smoothing).

The final estimation is shown in Equation 7.3, where the second stage is accounting for document length normalization. Presumably, if query variance needed to be accounted for, then it could add a further layer of smoothing.

$$p(t|\theta_d) = \frac{n(d)}{n(d)+\beta}\left((1-\lambda)p(t|d)+\lambda p(t|\theta_C)\right)+\frac{\beta}{n(d)+\beta}p(t|\theta_C) \tag{7.3}$$

Under this 'reconsidered' two stage approach it may be possible to obtain better retrieval performance from the context based models. This is because in Chapter 6, we only considered making a better representation of document (or $p(t|R,d)$), whilst ignoring the influences of the $p(t|N,d)$ and the document length.

7.2.5 Other Retrieval Models

We have exclusively restricted our attention to the Language Modeling approach to *ad hoc* retrieval and we have not considered other retrieval models. We did so for a couple of reasons; (1) we wanted to deepen our understanding of the Language Modeling approach, so that we could motivate future developments (such as the context based document models). And (2) we wanted to see if the assumptions of Language Modeling could be used to obtain better retrieval. Hence, the most appropriate baseline is the standard Language Modeling approaches. Language models have been shown to achieve comparable performance to the other models, and hence represent a sufficient baseline for initial comparisons (i.e. can the baseline of standard document models be surpassed by context based document models?).

Nonetheless, further research is motivated by our work to re-assess the Language modeling approach with respect to the other retrieval models on the following points: (1)

how do language models perform against other current competing models (such as OKAPIs BM25 and VSMs TF.IDF) when the language models are either estimated (JMe) or set to a standard parameter value setting (JM50, where $\lambda = 0.5$)?, and (2) do other models correlate to relevance as well or better than LMs under the A1 assumption?

7.3 Context Hypothesis

In Chapter 6, we examined three different forms of context within the context based document models (PLSA, TOPIC and LINK). The context based document models were our attempt at providing evidence for the Context Hypothesis. We have shown that on occasions the contexts (PLSA and TOPIC) can benefit retrieval and so providing some support for the Context Hypothesis. However, on the whole there was no significant difference between the standard and the context document models, when the parameters were estimated. As mentioned above, we can only extract so much (if any) in terms of retrieval performance by improving the document representations. So, under the context based document models we can not provide enough evidence to definitely support the Context Hypothesis.

However, it may be the way in which we have implemented the Context Hypothesis through the document modeling side or the context we employed. Using the semantic associations between documents as the context and the way we evaluated their performance requires that these associations

- provide a better representation of the documents (i.e. A2)
- capture the users understanding of the documents in the collection (i.e. A2.1)
- are generalizable to all users, and
- are applicable regardless of the query/need (i.e query independent)

This raises several issues with context based document models. The quality of the context will determine how good our representations are, which will in turn affect the

document's score with respect to a query (i.e. A2). How well the context will match the query terms depends on how reflective the contexts are of representing the users context. Here we have assumed that the contexts are generalizable to all users of the system. However, in general this did not appear to be the case. For specific queries we noted a substantial increase in retrieval performance. Possibly this was due to the context being appropriate to that request. If this is the case, then model selection is required to select which representations are going to provide the best retrieval performance. Whether this is possible is as yet undetermined, but is being considered elsewhere[111].

Our interpretation of the Context Hypothesis has been steered by the assumptions of Language Modeling which prompted the development of context based document models. The alternative to modeling on the document side, looking to improve the query. For instance, we could employ context to expand the query or re-weight the importance of query terms. Another option is that, we could consider context as a bias that the user has for particular documents in collection given their information need. This view considers the user's immediate search context and defines their pre-existing beliefs about what is relevant and what is not. During the course of searching, the user refines their search, and their context narrows as they focus on particular documents (i.e. the relevant set of documents). In that sense, we can consider relevance as a specific context. Restated, the semantic associations between documents define the context, and this context is the set of documents relevant to a query. Under the Context Hypothesis, a request to the IRS would comprise of two parts; the query and the context. This context is usually assumed to be undefined and represents when no bias is introduced by the user's prior beliefs on relevance. Here we still consider context as defined by a probability distribution over the vocabulary, however, other efforts have examined context as a set of non textual features[122].

In the following subsection, we present the Integrated Language Modeling approach, where we first derive the model and then show how we can incorporate the user's context as an *a priori* belief about the relevance of documents.

7.3.1 The Integrated Language Model

We begin our decomposition of the Integrated Language Modeling approach by starting with the premise that we would like to rank documents according to the log Odds of relevance given the document and a query (i.e the log Odds Ratio). This can be expressed as:

$$\log O(r|d,q) = \log \frac{p(q|d,R)}{p(q|d,N)} + \log \frac{p(R|d)}{p(N|d)} \tag{7.4}$$

Recall from Section 3.2.1, that the Odds Ratio and the right hand side expression are mathematically equivalent, and for the expression to be reduced to the query likelihood approach the two sub assumptions, A1.1 and A1.2, were required. We make no such assumptions. Instead, we apply Bayes Theorem to the prior on relevance given a document such that we obtain Equation 7.5.

$$\log O(r|d,q) = \log \frac{p(q|d,R)}{p(q|d,N)} + \log \frac{p(d|R)p(R)}{p(d|N)p(N)} \tag{7.5}$$

Note that, the decomposition in Equation 7.5 is in direct contrast with the previous probabilistic models because of the dependencies assumed. In Figure 7.1, the dependencies between the variables q, d and R are shown for the classical probabilistic model[119], the language model[113] according to [78], the generative relevance model[83] and the integrated language model.

From this decomposition, we estimate the Odds ratio through an Integrated Language Modeling approach. That is, we evaluate each conditional probability using generative language models. We describe how to implement the model and show how under different conditions the model can reduce to either the standard Language Modeling approach or the Relevance Modeling approach (described in Section 3.5.5). Then, we show how the user's context can be encoded within such a model.

First, we assume a generative view of relevance can be taken as in [82], where relevance and non-relevance are described as multinomial term distributions, respectively. Thus, relevance R is defined by the relevance model θ_R, and similarly, non-relevance N is defined by the non-relevance model θ_N. Further, we ignore the prior probability

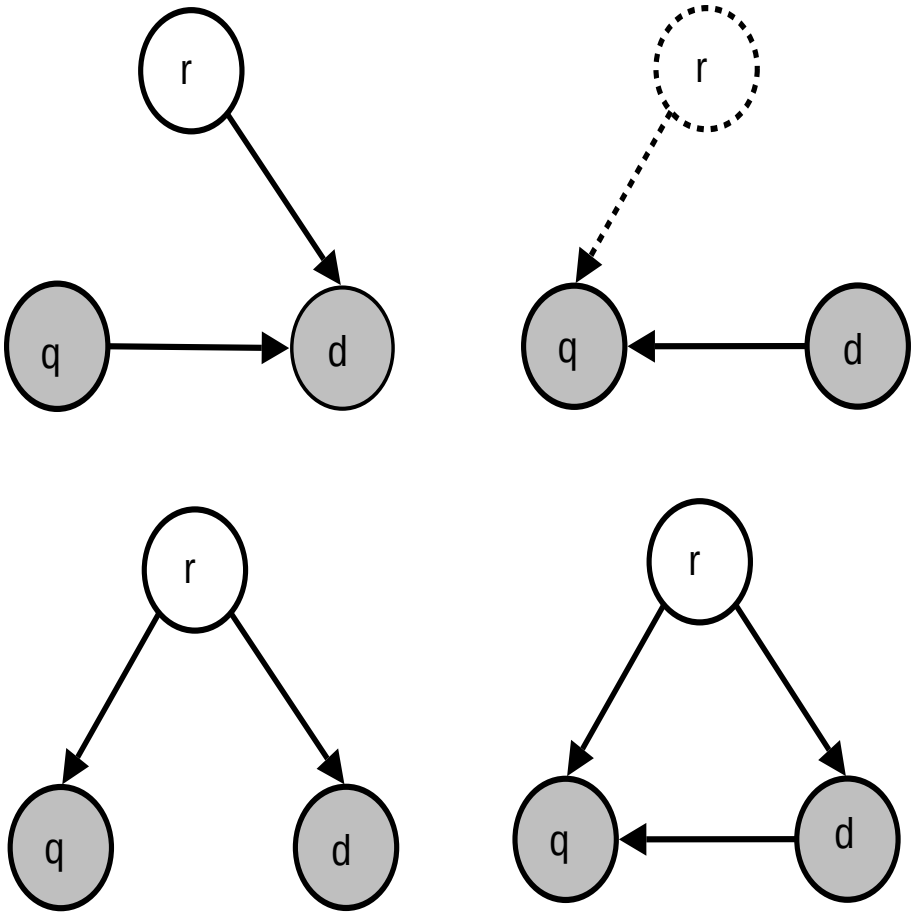

Figure 7.1: Graphical diagrams showing the dependencies between the query q, the document d and relevance r variables in different probabilistic IR models. (Shaded circles represent observable variables)

of relevance versus non relevance $\frac{p(R)}{p(N)}$, since it is a constant and will not affect the ranking.

We also assume that $p(q|d,R)$ can be approximated by $p(q|\theta_d,\theta_R)$ and the $p(q|d,N)$ can be approximated by the $p(q|\theta_N)$ as done earlier. The odds ratio is now proportional to Equation 7.6.

$$\log \frac{p(R|q,d)}{p(N|q,d)} \propto \log \frac{p(q|\theta_d,\theta_R)}{p(q|\theta_N)} + \log \frac{p(d|\theta_R)}{p(d|\theta_N)} \tag{7.6}$$

The query likelihoods are computed according to Equation 3.17 where it is assumed that the query terms are drawn independently and identically from the document model and non relevance model, such that:

$$p(q|\theta_d,\theta_R) = \prod_{t\in q} p(t|\theta_d,\theta_R)^{n(t,q)} \tag{7.7}$$

$$p(q|\theta_N) = \prod_{t\in q} p(t|\theta_N)^{n(t,q)} \tag{7.8}$$

where $p(t|\theta_d,\theta_R) = (1-\lambda)p(t|\theta_d) + \lambda p(t|\theta_R)$. The document likelihoods are computing according to Equation 3.50, where again the document terms are sampled independently and identical drawn from the (non)relevance models.

$$p(d|\theta_R) = \prod_{t\in d} p(t|\theta_R)^{n(t,d)} \tag{7.9}$$

$$p(d|\theta_N) = \prod_{t\in d} p(t|\theta_N)^{n(t,d)} \tag{7.10}$$

The full ranking function, after substituting in the above expression can be expressed as in shown below:

$$\begin{aligned}
\log \frac{p(R|q,d)}{p(N|q,d)} &\propto \log \frac{\prod_{t\in q} p(t|\theta_d,\theta_R)^{n(t,q)}}{\prod_{t\in q} p(t|\theta_N)^{n(t,q)}} + \log \frac{\prod_{t\in d} p(t|\theta_R)^{n(t,d)}}{\prod_{t\in d} p(t|\theta_N)^{n(t,d)}} \\
&= \sum_{t\in q} n(t,q) \log \frac{\big(\lambda p(t|\theta_d) + (1-\lambda)p(t|\theta_R)\big)}{p(t|\theta_N)} + \sum_{t\in d} n(t,d) \log \frac{p(t|\theta_R)}{p(t|\theta_N)}
\end{aligned}$$

So far we have presented an integrated model without respect to how one would actually instantiate the document model θ_d, relevance model θ_R and non relevance model θ_N. The following sections detail how this could be performed when various states of knowledge exist.

7.3.1.1 Without any prior knowledge

It is a typical scenario in information retrieval that the system is submitted a impoverished description of the users information need in the form of a query. Thus, we have no *a priori* knowledge of the relevance model θ_R. In this situation the relevance model can be assumed to be equivalent to the collection model θ_C since we do not know any better. Whilst we have no knowledge about non-relevance distribution either, we can use the collection model as an estimate of the non-relevance model θ_C. This was originally proposed in the context of the classical probabilistic model[26] and the idea has been successfully applied since for (non) Relevance Models[83]. This is quite a reasonable estimate for non-relevance, and has been show to perform empirically well in both instances. Therefore, $\theta_R = \theta_N = \theta_C$ and it represents the case when we have no *a priori* knowledge. Consequently, the log posterior probability of the document given relevance over non-relevance will be equal to zero. Further, if we set λ equal to one, then $p(t|\theta_d, \theta_R) = p(t|\theta_d)$, making the ranking function proportional to the standard Language Modeling approach.

$$\begin{aligned} \log \frac{p(R|q,d)}{p(N|q,d)} &\propto \log \frac{p(q|\theta_d)}{p(q|\theta_C)} + \log \frac{p(d|\theta_C)}{p(d|\theta_C)} \\ &\propto \log p(q|\theta_d) \end{aligned}$$

Obviously, this is similar to decomposition offered by Lafferty and Zhai[78]. However, we arrive at the LM approach as a result of not knowing any other information, as opposed to making a convenient assumption. When relevance information, either, *a priori* or *a posteriori*, does become available we can incorporate this directly into the model. This is without recourse to any theoretical problems of how to deal with the relevance data, because there is a natural mechanism to utilize such information.

7.3.1.2 With relevance information

When relevance information is available (either implicitly, explicitly or through pseudo relevance feedback) then it is possible to obtain an estimate of the relevance model $p(t|\hat{\theta}_R)$ *a posteriori*. Here, the relevance feedback is the context that is defined by the set of documents which are relevant to the information need. A relevance model $p(t|\hat{\theta}_R)$ is created with this set of feedback documents and then ranking can be performed according to Equation 7.11, where the results will be query biased, so to speak.

Note, that this is going to result in a slightly different ranking to the originally proposed relevance model[83], because the influence of the query is still a factor in the equation. Depending on the difference between the two distributions, θ_R and θ_N, the bias introduced by the query will vary. If the difference is very small, the influence of the query will dominate the overall score, whereas if the difference is large, then the document prior will dominate the ranking. This is a known problem with using the document prior within the scoring function. This issue that will need to be addressed for the effective utilization of contextual evidence for both this approach and the standard LM approach.

If we perform query contraction, such that $|t \in q| = 0$, then the ranking documents will be based solely on the document likelihood and the model will be equivalent to the generative Relevance Model[83].

7.3.1.3 With Context

Any form of context, defined as semantic associations between documents, can be used to bias the retrieval of the documents so that the documents are 'in context' with the user and their information need. As we previously mentioned in Chapter 4, the context could be anything from topics and semantic clustering to user profiling and collaborative filtering.

Given a set of predefined contexts $x \in X$, then the user could select the specific context x that is the most appropriate for their need, or the IRS could attempt to select the context

x on the user's behalf. The context defined by $p(t|x)$ would then be used to estimate an *a priori* relevance model to bias the ranking according to the context x. The contexts used in this thesis could be re-used such that they are chosen as the context of the user, instead of using them for the context based document models. It would be interesting to see whether selecting the context could improve the retrieval performance under the integrated model, though this is left to future work.

The advantage of the Integrated Language modeling approach is that *a priori* knowledge (i.e. context) can be encoded directly into the model, if available, whist still ranking with respect to the query. When *a posteriori* relevance data becomes available, this too can be encoded directly into the model as relevance feedback. Further, query expansion (or even contraction) may be performed, independently or in conjunction with the relevance model. The integrated model provides several possibilities for ingraining contextual evidence within a principled framework. Furthermore, the Integrated Language Modeling approach provides a novel combination of the query likelihood and document likelihood approaches within one framework.

7.4 Summary

In this chapter we discussed the main points relating to our research, including the validity of the assumptions of Language Modeling, why the assumptions broke down, and areas for further research. We acknowledged that there were other ways to incorporate context and concluded the chapter by proposing an alternative implementation of the Context Hypothesis using the Integrated Language Modeling approach. This approach can incorporate the users direct and immediate context within the model and represents an integration between the different paradigms of language modeling for *ad hoc* text retrieval.

Chapter 8

Conclusions

This chapter concludes the thesis with an overview of the work performed herein, followed by a summary of the contributions to knowledge. Finally, we conclude the thesis by detailing directions for future research which stem directly from this work.

8.1 Summary of Work

The premise of this thesis was that documents that are semantically associated tend to be more relevant than documents that are not, with respect to the user and their query. We adopted the Language Modeling approach as an intuitive framework for naturally embedding the user's context and understanding of the language used within documents. By considering the underlying assumptions of the model, we identified the possibility of obtaining better retrieval performance by building better representations of the documents using the context associated with that document. Hence, we proposed context based document models that attempted to capture the user's understanding. The user's understanding was quantified through the semantic associations between documents and reflected how the users perceived the documents in the document and their context. This provided an instantiation of the Context Hypothesis, for which we provided some evidence towards.

Our testing and analysis commenced with an examination of three main assumptions of the Language Modeling approach for *ad hoc* information retrieval. We found that the underlying assumptions held to a certain extent, but there were times when the assumptions were violated. We have ascertained that the query likelihood was correlated to the document's relevance. Further, this correlation improved when the user submitted queries that sufficiently discriminated relevant documents from non relevant documents. However, to make a stronger claim that the query likelihood is proportional to the document's relevance can not be justified by our research. The unification of the data and retrieval model only occurred when certain smoothing techniques were applied and the divergence of the data and retrieval model appeared to be from the other factors known to affect the retrieval performance. When we built the context based document models we were consistently able to build better document representations, however this did not neccessarily translate into better retrieval performance. This shows that there would appear to be a limit as to how good the representation needs to be, in order to achieve effective retrieval performance. On the other hand, the analysis of the third assumption suggested that much improvement to retrieval performance could be obtained if better queries (ones which are consistent with the assumptions) were submitted.

Under our interpretation of the smoothing in Section 7.1.5, when we estimated a document language model, one part was attributed to the query's contribution to the relevance of a document and the other accounted for the query's contribution to the non relevance of a document. Consequently, by smoothing the document models with the background collection model, is it accounting for the query terms' contribution to non-relevance? (i.e. the noise detracting from our correlation between $p(q|d)$ and $p(q|d,R)$). By doing so, the model's retrieval performance should improve and the correlation in A1 should also improve. When we employed the context based document models our focus was on attempting to improve the representational quality of the documents. That is, we attempted to improve the document such that the distribution of terms appearing within the document would be more indicative of the document's relevance. This led us to derive a different formulation of the two stage model, where the first stage attempts to make the best representation of the underlying data as pos-

sible, whilst the second stage of smoothing accounts for the non-relevance. However, after re-considering how we can effectively use the context of document associations to improve the retrieval performance, we proposed the Integrated Language Modeling where the information need is composed of both query terms and query context. Under this approach, context was considered as a loosely defined notion of relevance and when the context becomes more focused (explicated and model), it would define a specific context, that of relevance with respect to the query.

The Language Modeling approach has provided a renaissance of the application of probability theory to *ad hoc* Information Retrieval, which has led to many interesting avenues of research. This thesis has taken an in depth examination into the theory and application of the LM approach and exposed some of the limitations. Essentially, a Language Model is only as good as the parameters that can be estimated. The extent to which we can estimate these parameters such that it maximizes the retrieval performance is dependent on the quality and amount of data that is available and, of course, the validity of the model assumptions.

8.2 Contributions to Knowledge

Within this thesis there are several notable contributions. These are outlined below:

- The formalization of the underlying assumptions of the Language Modeling Approach for *ad hoc* Information Retrieval (See Section 3.2 and Appendix A).
- Development of a principled framework for context based document modeling (See Chapter 4).
- Analysis of the underlying assumptions of the Language Modeling approach to *ad hoc* retrieval [3](See Chapter 5).
- An evaluation of Probabilistic Latent Semantic Analysis within a language modeling framework[4] (See Section 6.1).
- An empirical analysis of context based document language models on different

collections. Contexts were represented by the association between documents, either through unsupervised learning techniques, user interaction, or through explicit user reference (such as hyper links and citations)[4, 5]. (see Chapter 6).

- A different interpretation of smoothing was set forth which acknowledges relevance in the model. (See Section 7.1.5).
- Recapturing dependencies to derive a two component language model - that addresses the criticisms of the standard and relevance modeling approaches by providing a mechanism for both query expansion and relevance feedback. (See Section 7.3.1).

8.3 Further Work

We have identified several avenues of future work which are motivated by the work contained herein.

- The analysis of the assumptions made by other retrieval models; specifically whether other retrieval models correlate as well or better than the Language Modeling approach to the ranking according to the relevance of a document.
- A re-assessment of the retrieval performance obtainable by the Language Modeling approach; when compared with other competing retrieval models how does the Language Modeling approach fare when its parameters are estimated or set to some nomimal value obtained through empirical evaluation such as JM50 and JMe.
- An assessment of our two stage model to determine whether accounting for the non-relevance contribution of query terms will improve the retrieval performance of the context based document models.
- An exploration into document model selection, where the document model that best represents the user's understanding with respect to the query is selected with the goal of maximising the retrieval performance.

- A user study to validate whether the users can submit queries which are consistent with the assumptions. Further, if users are trained can they submit queries which are more effective with respect to retrieval performance.
- The development of user interfaces that support the user in formulating queries which are consistent with the underlying assumptions of the Language Modeling approach. Providing this information to the user in some form or another should enable them to formulate queries of better quality.
- An investigation into whether we can determine when automatic query expansion will improve the retrieval performance, and what type of query expansion is most applicable given the information need (for instance using Q1 or Q2).
- The development and empirical testing of the Integrated Language Model to see whether the context can be used as a surrogate for relevance, and if this can improve retrieval performance. Further, to determine whether we can automatically identify the appropiate context of the user to use in the Integrated Language Model.

Appendix A

Assumptions of Language Modeling

The underlying assumptions of Langauge Modeling are stated as follows:

A1 Correlation The probability of a query given a document is *correlated* with the probability of a document being relevant[113, 57] . Stated, more firmly, the probability of a query given a document is *proportional* to the probability of the document being relevant[78]. Where for the latter, the following assumptions are required:

A1.1 The probability of a document and a query given the event of non-relevance, the document and query are independent. i.e. $p(d,q|N) = p(d|N)p(q|N)$.

A1.2 The probability of a document and relevance (or non relevance) is independent, i.e. $p(d,R) = p(d)p(R)$ and $p(d,N) = p(d)p(N)$

A2 Unification The data model and the retrieval function are one and the same as relevance is subsumed by the document modeling process[113, 78].

A2.1 The user has some understanding of the distribution of terms with in documents.

A3 Discrimination The terms that a user submits as a query will be sufficient in discriminating relevant from non relevant documents

A3.1 The user will issue query terms that are highly discriminative[113], i.e.

will identify relevant from non-relevant, or

A3.2 The user will issue query terms that are highly likely in relevant documents[96].

Bibliography

[1] G. Amati. *Divergence from Randomness*. PhD thesis, Department of Computer Science, University of Glasgow, 2003.

[2] A. Arampatzis, J. Beney, C. Koster, , and T. van der Weide. Kun on the trec-9 filtering track: Incrementality, decay, and threshold optimization for adaptive filtering systesms. In *Proceedings of the 9th Text REtrieval Conference*, 2001.

[3] L. Azzopardi, M. Girolami, and C. J. van Rijsbergen. Investigating the relationship between language model perplexity and ir precision-recall measures. In *Proceedings of the 26th Annual International ACM SIGIR conference on Research and development in information retrieval*, pages 369–370, Toronto, Canada, 2003.

[4] L. Azzopardi, M. Girolami, and C. J. van Rijsbergen. Topic based language models for ad hoc information retrieval. In *Proceedings of the International Joint Conference in Neural Networks, IJCNN*, Budapest, Hungary, 2004.

[5] L. Azzopardi, M. Girolami, and C. J. van Rijsbergen. User biased document language modelling. In *Proceedings of the 27th Annual International ACM SIGIR conference on Research and development in information retrieval*, pages 542–543, Sheffield, UK, 2004.

[6] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.

[7] R. K. Belew. *Finding Out About: A Cognitive Perspective on Search Engine Technology and the WWW*. Cambridge University Press, 2000.

[9] N. J. Belkin. Anomalous states of knowledge as a basis for information retrieval. *The Canadian Journal of Information Science*, 5:133–143, 1980.

[8] N. J. Belkin and W. B. Croft. Information filtering and information retrieval: Two sides of the same coin? *Communications of the ACM*, 35(12):29–38, 1992.

[10] J. M. Benardo and Smith A. F. M. *Bayesian Theory*. Wiley, New York, 1994.

[11] A. Berger and J. Lafferty. Information retrieval as statistical translation. In *SIGIR-99 Research and Development in Information Retrieval*, pages 222–229, Berkeley, CA., 1999.

[12] D. M Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[13] A. Bookstein and D. Swanson. Probabilistic models for automatic indexing. *Journal of the American Society for Information Science*, 25:312–318, 1974.

[14] J. D. Burger, D. Palmer, and L. Hirschman. Named entity scoring for speech input. In *Thirty-Sixth Annual Meeting of the Association for Computation Linguistics and Seventeenth Information Conference on Computational Linguistics*, pages 201–205, San Franciso, California, 1998.

[15] F. Chen, S. and J. Goodman. An empirical study of smoothing techniques for language modeling. In *Proceedings of the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics*, pages 310–318, San Francisco, 1996. Morgan Kaufmann Publishers.

[16] Y. Chiaramella and J. P. Chevallet. About retrieval models and logic. *The Computer Journal*, 35:233–242, 1992.

[17] M. Claypool, P. Le, M. Waseda, and D. Brown. Implicit interest indicators. Technical Report WPI-CS-TR-00-18, Computer Science Department, Worcester Polytechnic Institute, Worcester, MA, 2000.

[18] C. W. Cleverdon, J. Mills, and Keen. M. *Readings in Information Retrieval*, chapter Factors Determining the Performance of Indexing Systems, Volume I - Desing, Volume II - Test Results, ASLIB Cranfield Project, pages 1–24. Morgan Kaufman, 1997.

[19] W. Cleverdon, C., J. Mills, and M. Keen. Aslib cranfield research project: factors determining the performance of indexing systesms. Technical report, Cranfield Institute of Technology, Cranfield, England, 1966.

[20] D. Cohn and H. Chang. Learning to probabilistically identify authoritative documents. In *Proceedings of the 17th International Conference on Machine Learning*, pages 167–174. Morgan Kaufmann, 2000.

[21] W. S. Cooper. Some inconsistencies and misnomers in probabilistic information retrieval. *ACM Transactions on Information Systems*, 13:1, 100-111.

[22] W. S. Cooper. A definition of relevance for information retrieval. *Information Storage and Retrieval*, 7:19–37, 1971.

[23] N. Craswell, D. Hawking, and S. Robertson. Effective site finding using link anchor information. In *Proceedings of the 24th annual international ACM SI-*

GIR conference on Research and development in information retrieval, pages 250–257. ACM Press, 2001.

[24] F. Crestani. *A Study of the Kinematics of Probabilities in Information Retrieval*. PhD thesis, Department of Computing Science, University of Glasgow, Glasgow, Scotland, 1998.

[25] F. Crestani, M. Lalmas, C. J. van Rijsbergen, and I. Campbell. "is this document relevant? ..probably". a survey of probabilistic models in information retrieval. *ACM Computing Surveys*, 30(4):528–552, 1998.

[26] W. B. Croft and D. J. Harper. Using probabilistic models of document retrieval without relevance information. *Journal of Documentation*, 35:285–295, 1979.

[27] D. Cutting, J. Kupieg, J. Pedersen, and P. Sibun. A pratical part-of-speech tagger. In *Third Conference on Applied Natural Language Processing*, 1992.

[28] S. Deerwester, S. T. Dumais, G. W. Furnas, and T. K. Landauer. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407, 1990.

[29] M. DeGroot. *Optimal Statistical Decisions*. McGraw-Hill, New York, 1970.

[30] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of Royal Statistics Society*, B(39):1–38, 1977.

[31] F. Diaz and R. Jones. Using temporal profiles of queries for precision prediction. In *Proceedings of the 27th annual international conference on Research and development in information retrieval*, pages 18–24. ACM Press, 2004.

[32] W. Fan, M. Luo, L. Wang, M. Xi, and A. Fox, E. Tuning before feedback: combining ranking discovery and blind feedback for robust retrieval. In *Proceedings of the 27th annual international conference on Research and development in information retrieval*, pages 138–145. ACM Press, 2004.

[33] H. Fang, T. Tao, and C. Zhai. A formal study of information retrieval heuristics. In *Proceedings of the 27th annual international conference on Research and development in information retrieval*, pages 49–56. ACM Press, 2004.

[34] C. Fox. *Information Retrieval: Data Strutures and Algorithms*, chapter Lexical Analysis and Stoplists, pages 102–130. Prentice Hall, 1992.

[35] T. J. Froehilch. Relevance reconsidered - towards an agenda for the 21st century: Introduction to special topic issue on relevance research. *Journal of the American Society of Information Science*, 45:124–134, April 1994.

[36] N. Fuhr. Probabilistic models in information retrieval. *The Computer Journal*, 35(3), 1992.

[37] N. Fuhr and C. Buckley. Models for retrieval with probabilistic indexing. *Information Processing and Management*, 25:55–72, 1989.

[38] N. Fuhr and C. Buckley. A probabilistic learning approach for document indexing. *ACM Transactions on Information Systems*, 9:223–248, 1991.

[39] G. Furnas, T. K. Landauer, L. M. Gomez, and S. T. Dumais. The vocabulary problem in human-system communications. *Communications of the ACM*, 30:964–971, 1987.

[40] J. Gao, J. Y. Nie, G. Wu, and G. Cao. Dependence language model for information retrieval. In *In the proceedings of the 27th Annual ACM Conference on Research and Development in Information Retrieval, SIGIR*, pages 170–177, Sheffield, UK, 2004.

[41] M. Girolami and A. Kaban. On an equivalence between plsi and lda. In *Proceedings of the 26th Annual ACM Conference on Research and Development in Information Retrieval, SIGIR*, pages 433–434, Toronto, Canada, 2003.

[42] I. J. Good. The population frequencies of species and the estimation of population parameters. *Biometrika*, 40(324):237–264, 1953.

[43] W. Greiff, A. Morgan, R. Fish, M. Richards, and A. Kundu. Fine-grained hidden markov modeling for broadcast news story segmentation. In *Human Language Technology Conference*, San Diego, California, 2001.

[44] W. R. Greiff and W. T. Morgan. Contributions of language modeling to the theory and practice of ir. In W. B. Croft and J. Lafferty, editors, *Language Modeling for Information Retrieval*, pages 73–94. Kluwer Academic Publishers, 2003.

[45] C. Gurrin and A. F. Smeaton. Improving the evaluation of web search systems. In *Proceedings of the 25th European Conference on IR Research (ECIR)*, pages 25–40. Springer, 2003.

[46] D. Harman. How effective is suffixing? *Journal of the American Society for Information Science*, 42:7–15, 1991.

[47] D. Harman. Overview of the first text retrieval conference (trec-1). In D. K. Harman, editor, *The First Text REtreival Conference (TREC 1)*, pages 1–20. NIST Special Publication 500-207, February 1993.

[48] D. Harper, G. Muresan, B. Liu, I. Koychev, D. Wettschereck, and N. Wiratunga. The robert gordon university's hard track experiments at trec 2004. In *In Proceedings of the 13th Text REtrieval Conference (TREC2004)*, 2004.

[49] D. J. Harper and C. J. van Rijsbergen. An evaluation of feedback in document retrieval using co-occurrence data. *Journal of Documentation*, 34:189–216, 1978.

[50] S. P. Harter. A probabilistic approach to automatic keyword indexing, part i: On the ditribution of speciality words in technical literature. *Journal of the American Society for Information Science*, 26:197–206, 1975.

[51] S. P. Harter. A probabilistic approach to automatic keyword indexing, part ii: An algorithm for probabilsitic indexing. *Journal of the American Society for Information Science*, 26:280–289, 1975.

[52] C. Hauff and L. Azzopardi. Age dependent document priors in link structure analysis. In *The 27th European Conference in Information Retreival*, pages 552–554. Springer, 2005.

[53] H. Hawking, T. Upstill, and N. Craswell. Toward better weighting of anchors. In *Proceedings of the 27th annual international conference on Research and development in information retrieval*, pages 512–513. ACM Press, 2004.

[54] B. He and I. Ounis. A query-based pre-retrieval model selection approach to information retrieval. In *In Proceedings of RIAO 2004 (Recherche d'Information Assistee par Ordinateur - Computer assisted information retrieval*, 2004.

[55] D. Hiemstra. A linguistically motivated probablistic model of information retrieval. In C. Nicolaou and C. Stephanidis, editors, *Lecture Notes in Computer Science: Research and Advanced Technology for Digital Libraries*, volume 513, pages 569–584. Springer-Verlag, 1998.

[56] D. Hiemstra. A probabilistic justification for using tf . idf term weighting in information retrieval. *International Journal for Digital Libraries*, 3:131–139, 2000.

[57] D. Hiemstra. *Using Language Models for Information Retrieval*. PhD thesis, University of Twente, Enschede, 2001.

[58] D. Hiemstra. Term-specific smoothing for the language modeling approach to information retrieval: The importance of a query term. In *2002 ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 35–41, Tampere, Finland, 2002.

[59] D. Hiemstra, S. Robertson, and H. Zaragoza. Parsimonious language models for information retrieval. In *Proceedings of the 27th Annual ACM Conference on Research and Development in Information Retrieval, SIGIR*, pages 178–185, Sheffield, UK, 2004.

[60] T. Hofmann. Probabilistc latent semantic indexing. In *Proceedings of the*

22nd International Conference on Research and Development in Information Retrieval. ACM Press, 1999.

[61] T. Hofmann. Probabilistic latent semantic analysis. In *Proceedings of the 15th Conference of Uncertainty in Artificial Intelligence, UAI'99*, pages 289–296, Stockholm, Sweden, 1999.

[62] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42:177–196, 2001.

[63] T. Hofmann and J. Puzicha. Unsupervised learning from dyadic data. Technical Report TR-98-042, International Computer Science Institute, December 1998 1998.

[64] D. A. Hull. Using statistical testing in the evaluation of retrieval experiments. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 329–338. ACM Press, 1993.

[65] D. A. Hull. Stemming algorithms: A case study for detailed evaluation. *Journal of the American Society of Information Science*, 47:70–84, 1996.

[66] P. Ingwersen. *Information Retrieval Interaction*. Talyor Graham, London, 1992.

[67] N. Jardine and C. J. van Rijsbergen. The use of hierarchic clustering in information retrieval. *Information Storage and Retrieval*, 7:217–240, 1971.

[68] F. Jelinek and R. L. Mercer. Interpolated estimation of markov source parameters from sparse data. In *Workshop on Pattern Recognition in Practice*, Amsterdam, The Netherlands, 1980.

[69] R. Jin, A. G. Hauptmann, and C. Zhai. Title language model for information retrieval. In *2002 ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 42–48, Tampere, Finland, 2002.

[70] T. Kalt. A new probabilistic model of text classification and retrieval. Technical Report CIIR TR98-18, University of Massachusetts, January 25, 1996 1996.

[71] S. M. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-35(3):400–401, 1997.

[72] W. Kraaij and M. Spitters. Language models for topic tracking. In W. B. Croft and J. Lafferty, editors, *Language Modeling for Information Retrieval*, pages 95–124. Kluwer Academic Publishers, 2003.

[73] W. Kraaij, T. Westerveld, and D. Hiemstra. The importance of prior probabilities for entry page search. In *2002 ACM SIGIR Conference on Research and*

Development in Information Retrieval (SIGIR), pages 27–34, Tampere, Finland, 2002.

[74] R. Krovetz. Viewing morphology as an inference process. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 191–202. ACM Press, 1993.

[75] S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–86, 1951.

[76] O. Kurland and L. Lee. Corpus structure, language models and ad hoc information retrieval. In *Proceedings of the 27th Annual ACM Conference on Research and Development in Information Retrieval, SIGIR*, pages 194–201, Sheffield, UK, 2004.

[77] J. Lafferty and C. Zhai. Document language models, query models, and risk minimization for information retrieval. In *2001 ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, New Orleans, LO, 2001.

[78] J. Lafferty and C. Zhai. Probablistic relevance models based on document and query generation. In W. B. Croft and J. Lafferty, editors, *Language Modeling for Information Retrieval*, pages 1–10. Kluwer Academic Publishers, 2003.

[79] M. Lalmas. *Theories of Information and Uncertainty for modelling of Information Retrieval: an application of Situation Theory and Dempster-Shafer's Theory of Evidence*. PhD thesis, University of Glasgow, UK, 1996.

[80] M. Lalmas. *The flow of information in information retrieval: Towards a general framework for the modeling of information retrieval*. 1998.

[81] R. Y. K. Lau, P. D. Bruza, and D. Song. Belief revision for adapative information retrieval. In *In the proceedings of the 27th Annual ACM Conference on Research and Development in Information Retrieval, SIGIR*, pages 130–137, Sheffield, UK, 2004.

[82] V. Lavrenko and W. B. Croft. Relevance-based language models. In *Proceedings of the 24th annual international ACM SIGIR conference*, pages 120–127, New Orleans, LA, 2001. ACM Press.

[83] V. Lavrenko and W. B. Croft. Relevance models in information retrieval. In W. B. Croft and J. Lafferty, editors, *Language Modeling for Information Retrieval*, pages 11–56. Kluwer Academic Publishers, 2003.

[84] S. Lawrence and C. L. Giles. Searching the web: General and scientific information access. *IEEE Communications*, 37:116–122, 1999.

[85] P. M. Lee. *Bayesian Statistics: An Introduction*. Arnold, second edition edition, 1997.

[86] D. D. Lewis. Evaluating text categorization. In *Proceedings of Speech and Natural Language Workshop*, pages 312–318. Morgan Kaufmann, 1991.

[87] X. Li and W. B. Croft. Time-based language models. In *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*, pages 469–475. ACM Press, 2003.

[88] G. J. Lidstone. Note on the general case of the bayes-laplace formula for inductive of a posteriori probabilities. *Transactions of the Faculty of Actuaries*, 8:182–192, 1920.

[89] X. Liu and W. B. Croft. Cluster-based retrieval using language models. In *Proceedings of the 27th Annual ACM Conference on Research and Development in Information Retrieval, SIGIR*, pages 186–193, Sheffield, UK, 2004.

[90] J. Lu and J. Callan. Federated search of text-based digital libraries in hierarchical peer-to-peer networks. In *The 27th European Conference in Information Retreival*, pages 52–66. Springer, 2005.

[91] H. P. Luhn. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2:159–165, 1958.

[92] D. J. C. MacKay and L. C. Peto. A hierarchical dirichlet language model. *Natural Language Engineering*, 1:1–19, 1995.

[93] C. D. Manning and H. Schutze. *Foundations of Statistical Language Processing*. MIT Press, Cambirdge, Massachusetts, 2000.

[94] A. A. Markov. An example of statistical investigation in the text of 'eugene onyegin' illustrating coupling of 'tests' in chains. In *Proceedings of the Academy of SCiences*, volume Volume 7 of VI, pages 153–162, St. Petersburg, 1913.

[95] M. E. Maron and J. L. Kuhns. On relevance, probabilsitic indexing and information retrieval. *Journal of the Association for Computing Machinery*, 7:216–244, 1960. initial attempt at appliying probability theory to information retrieval.

[96] D. R. H. Miller, T. Leek, and R. M. Schwartz. A hidden markov model information retrieval. In *22nd Annual International ACM SIGIR conference on Research and development in information retrieval*, pages 214–221, California, US, 1999. ACM Press.

[97] W. L. Miller. A probabilistic search strategy for medlars. *Journal of Documentation*, 27:254–266, 1971.

[98] M. Mitra, A. Singhal, and C. Buckley. Improving automatic query expansion. In *21st International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 206–214, 1998.

[99] V. O. Mittal and M. J. Witbrock. Language modeling experiments in non-extractive summarization. In W. B. Croft and J. Lafferty, editors, *Language Modeling for Information Retrieval*. Kluwer Academic Publishers, 2003.

[100] E. Mittendorf and P. Schuble. Document and passage retrieval based on hidden markov models. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 318–327. Springer-Verlag New York, Inc., 1994.

[101] S. Mizzaro. Relevance: The whole history. *Journal of the American Society of Information Science*, 48(9):810–832, 1997.

[102] S. Mizzaro. How many relevances in information retrieval? *Interacting with Computers*, 10(3):303–320, 1998.

[103] R. Nallapati. Discriminative models for information retrieval. In *Proceedings of the 27th annual international conference on Research and development in information retrieval*, pages 64–71. ACM Press, 2004.

[104] R. Nallapati and J. Allan. Capturing term dependencies using a sentence tree based language model. In *Proceedings of Conference of Information and Knowledge Management*, 2002.

[105] K. Ng. A maximum likelihood ratio information retrieval model. In *Proceedings of the Eighth Text REtrieval Conference (TREC-8)*, 1999.

[106] J. Y. Nie. An information retrieval model based on modal logic. *Information Processing and Managament*, 25:477–491, 1989.

[107] Y. Ogawa, T. Morita, and K. Kobayashi. A fuzzy document retrieval system using the keyword connection matrix and a learning method. *Fuzzy Sets and Systems*, 39:163–179, 1991.

[108] P. Ogilvie and J. Callan. Language models and structured document retrieval. In *Proceedings of the first INEX workshop*, 2003.

[109] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

[110] T. K. Park. Toward a theory of user-based relevance: A call for a new paradigm of inquiry. *Journal of the American Society of Information Science*, 45:135–141, 1994.

[111] V. Plachouras and I. Ounis. Usefulness of hyperlink structure for query-biased topic distillation. In *Proceedings of the 27th annual International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 448–455. ACM Press, 2004.

[112] J. M. Ponte. *A Language Modeling Approach to Information Retrieval*. PhD thesis, University of Massachusetts Amherst, 1998.

[113] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proceedings of the Twenty First ACM-SIGIR*, pages 275–281, Melbourne, Australia, 1998. ACM Press.

[114] M. F. Porter. An algorithm for suffix stripping. *Program*, 14:130–137, 1980.

[115] S. J. Press. *Bayesian Statistics*. Wiley, 1989.

[116] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of IEEE*, volume 77, pages 257–286, 1989.

[117] L. R. Rabiner, S. E. Levinson, and M. M. Sondhi. On the application of vector quantization and hidden markov models to speaker-independent, isolated word recogintion. *The Bell System Technical Journal*, 62:1075–1106, April 1983.

[118] B. L. Raktoe and J. J. Hubert. *Basic Applied Statistics*. Marcel Dekker Inc., New York, 1979.

[119] S. E. Robertson and K. Sparck-Jones. Relevance weighting of search terms. *Journal of the American Sociery for Information SCience*, 27:129–146, 1976.

[120] S. E. Robertson and K. Sparck-Jones. The probability ranking principle in ir. *Journal of Documentation*, 33(4):294–304, 1977.

[121] J. Rocchio. *The SMART Retrieval System: Experiments in Automatic Document Processing*, chapter Relevance feedback in IR, pages 313–323. Prentice Hall, 1971.

[122] H. Rode and D. Hiemstra. Conceptual language models for context-aware text retrieval. In *In Proceedings of the 13th Text REtrieval Conference (TREC2004)*, 2004.

[123] R. Rosenfeld. Two decades of statistical language modeling: Where do we go from here? *Proceedings of the IEEE*, 88(8), 2000.

[124] I. Ruthven and M. Lalmas. A survey on the use of relevance feedback for information access systems. *Knowledge Engineering Review*, 18(2):95–145, 2003.

[125] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

[126] G. Salton, E. A. Fox, and H. Wu. Extended boolean information retieval. *Communications of the ACM*, 26(11):1022–1036, 1983.

[127] G. Salton and M. E. Lesk. Computer evaluation of indexing and text processing. *Journal of the ACM*, 15(1):8–36, 1968.

[128] G. Salton and M. McGill. *Introduction to Information Retrieval*. McGraw-Hill Inc., New York, NY, 1983.

[129] L. Saul and F. Pereira. Aggregate and mixed-order markov models for statistical language processing. In *Proceedings of the Second International Conference on Empirical Methods in Natural Language Processing*, pages 81–89, Somerset, New Jersey, 1997. Association for Computational Linguistics.

[130] J. Savoy. Bayesian inference networks and spreading activation in hypertext systems. *Information Processing and Managament*, 23:389–406, 1992.

[131] T. Seracevic. The concept of "relevance" in information science: a historical review. In T. Seracevic, editor, *Introduction to Information Science*, page Chapter 14. R. R. Bower Company, New York, USA, 1970.

[132] C. E. Shannon. *The Mathematical Theory of Communication*. University of Illinois Press, Urbana, Illinois, 1948.

[133] T. B. Sheridan and W. R. Ferrel. *Man-Machine Systesm: Information, Control and Decision Models of Human Performance*. MIT Press, Cambridge, Mass., 1974.

[134] S. Siegel. *Non Parametric Statistics for the Behavioral Sciences*. McGraw-Hill and Kogakusha, international student edition edition, 1956.

[135] M. Simons, H. Ney, and S. C. Martin. Distance bigram language modelling using maximum entropy. In *IEEE Internation Conference on Acoustics, Speech and Signal Processing*, volume 2, pages 787–790, Munich, Germany, April 1997.

[136] A. Smeaton. *Natural Language Information Retrieval*, chapter Using NLP or NLP Resources for Information Retrieval Tasks, pages 99–112. Kluwer Academic Publishers, 1997.

[137] F. Song and W. B. Croft. A general language model for information retrieval. In *SIGIR ACM Research and Development in Information Retrieval*, pages 279–280, Berkeley, CA., 1999.

[138] K. Sparck-Jones. *Natural Language Information Retrieval*, chapter What is the Role of NLP in Text Retrieval?, pages 1–24. Kluwer Academic Publishers, 1997.

[139] K. Sparck-Jones, S. E. Robertson, D. Hiemstra, and H. Zaragoza. Language modeling and relevance. In W. B. Croft and J. Lafferty, editors, *Language Modeling for Information Retrieval*, pages 57–71. Kluwer Academic Publishers, 2003.

[140] K. Sparck-Jones, S. Walker, and S. E. Robertson. A probabilistic model of information retrieval: development and comparative experiments. *Information Processing and Management*, 36:779–808, 2000.

[141] K. Sparck-Jones and P. Willet, editors. *Readings in Information Retrieval*. Morgan Kaufman, 1997.

[142] M. Srikanth and R. Srihari. Incorporating query term dependencies in language models for document retrieval. In *In the proceedings of the 26th Annual ACM Conference on Research and Development in Information Retrieval, SIGIR*, pages 405–406, Toronto, Canada, 2003.

[143] J. A. Swets. *Effectiveness of Information Retrieval Methods*. Bolt, Beranek and Newman, 1967.

[144] J. M. Tague-Sutcliffe. Some perspectives on the evaluation of information retrieval systems. *Journal of the American Society for Information Science*, 47:1–3, 1996.

[145] R. S. Taylor. *Value-added processes in information systems*. Ablex Publishing, Norwood, NJ, 1968.

[146] H. Turtle and W. B. Croft. Inference network for document retrieval. In *Proceedings of the 13th Annual International ACM SIGIR Conference on Human Factors in Computing Systems*, pages 1–24, Brussels, Belgium, 1990.

[147] C. J. van Rijsbergen. A theoretical basis for the user of co-occurence data in information retrieval. *Journal of Documentation*, 33:106–119, 1977.

[148] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, London, second edition edition, 1979.

[149] C. J. van Rijsbergen. A non-classical logic for information retrieval. *The Computer Journal*, 29:481–485, 1986.

[150] C. J. van Rijsbergen. Towards a new information logic. In *1989 ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 77–86, Cambridge, USA, 1989.

[151] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, 1995.

[152] R. W. White, I. Ruthven, and J. M. Jose. Web document summarisation: a task-oriented evaluation. In *Proceedings of First International Workshop on Digital Libraries DLib 2001*, Munich, Germany, 3-7 September 2001.

[153] R. W. White, I. Ruthven, and J. M. Jose. The use of implicit evidence for relevance feedback in web retrieval. In *Proceedings of 24th BCS-IRSG European Colloquium on IR Research ECIR2002*, Glasgow, Scotland, 25-27 March 2002.

[154] S. K. M. Wong, W. Ziarko, and P. C. N. Wong. Generalized vector space model in information retrieval. In *Proceedings of the Eighth Annual International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 18–25. ACM Press, 1985.

[155] J. Xu and J. Callan. Effective retrieval with distributed collections. In *21st International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 112–120, 1998.

[156] J. Xu and R. Weischedel. A probabilistic approach to term translation for cross-lingual retrieval. In W. B. Croft and J. Lafferty, editors, *Language Modeling for Information Retrieval*, pages 125–140. Kluwer Academic Publishers, 2003.

[157] W. B. Xu, J.; Croft. Query expansion using local and global document analysis. In *Proceedings of the Nineteenth Annual International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 4–11, Zurich, Switzerland, 1996. ACM Press.

[158] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of the 14th International Conference on Machine Learning*, pages 412–420. Morgan Kaufmann Publishers, 1997.

[159] J. P. Yarmon, I. Carp, L. Gillick, S. Lowe, and P. van Mulbregt. A hidden markov model approach to text segmentation and event tracking. In *International Conference on Acoustics, Speech and Signal Processing*, Seatle, Washington, May 1998.

[160] H. Zaragoza, D. Hiemstra, M. Tipping, and S. Robertson. Bayesian extension to the language model for ad hoc information retrieval. In *Twenty-Sixth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 4–9, Toronto, Canada, July 2003.

[161] C. Zhai and J. Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *Proceedings of CIKM*, pages 403–410. ACM Press, 2001.

[162] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *ACM SIGIR Conference on Research*

and Development in Information Retrieval (SIGIR), pages 49–56, Tampere, Finland, 2001. ACM Press.

[163] C. Zhai and J. Lafferty. Two-stage language models for information retrieval. In *2002 ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 49–56, Tampere, Finland, 2002.

[164] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Transactoins on Information Systems*, 22:179–214, 2004.

[165] Y. Zhang and J. Callan. Maximum likelihood estimation for filtering thresholds. In *In the proceedings of the 24th Annual ACM Conference on Research and Development in Information Retrieval, SIGIR*, New Orleans, USA, 2001.

[166] G. K. Zipf. *Human Behavior and the Principle of Least Effort*. Addison-Wesley, Cambridge, MA, 1949.